

LINS – In.Co. – Facultad de Ingeniería - UdelaR



*Observatorio De Tecnologías de la Información  
Geográfica*

**Integrantes:**

Verónica Cardozo  
Sabrina Sosa  
Cristina Vilaboa

**Tutores:**

Msc. Ing. Raquel Sosa  
Ing. Bruno Rienzi

*Informe de Proyecto de Grado presentado al Tribunal Evaluador como  
requisito de graduación de la carrera Ingeniería en Computación*

Julio 2018

## Resumen

En los últimos años se ha detectado un crecimiento significativo en la cantidad de aplicaciones que hacen uso de información geográfica. Este hecho da lugar a que las Tecnologías de Información Geográfica (TIG) estén en constante expansión. Estas tecnologías se utilizan para construir sistemas de software de mediano y gran porte, lo que genera un gran interés, tanto en los desarrolladores como en las personas que trabajan en el área, en tener centralizada, categorizada y actualizada la información referente a las TIG.

Este proyecto toma como insumo un trabajo realizado en Facultad de Ingeniería en 2016, en el cual se relevó información sobre los estándares del Open Geospatial Consortium (OGC) y los productos que los implementan. También se toma la categorización de los productos que utiliza este trabajo, la cual utiliza los diferentes componentes presentes en la arquitectura de referencia planteada por el OGC de los Sistemas de Información Geográfica (SIG): Map Server, GeoDBMS, Componentes GUI (Desktop y Web) y librerías. A partir de dichos trabajos y complementando la información con otras fuentes, se busca diseñar y construir un observatorio de TIG. El cometido principal que se plantea es el de centralizar los datos sobre los diferentes productos existentes, categorizarlos respetando la arquitectura planteada y que dichos datos se mantengan actualizados. Por otro lado, se plantea la necesidad de ofrecer al usuario una interfaz amigable para que pueda buscar y comparar diferentes productos que implementan estándares geoespaciales de una manera sencilla.

A través de dicho observatorio la comunidad de usuarios TIG podrá acceder a la información de los diferentes productos, así como a los estándares que implementan. El observatorio es una plataforma colaborativa para que sus usuarios puedan interactuar a través de un foro, realizar calificaciones y comentarios sobre los productos. Gracias a la participación activa por parte de los usuarios, se logra una mejor comunicación de la comunidad SIG, lo cual permite que los usuarios se puedan basar en la experiencia de otros. También los usuarios son parte fundamental de la actualización de los datos debido a que cuentan con la funcionalidad de sugerir nuevos productos para que los mismos sean agregados al sistema luego de los controles pertinentes.

En cuanto a las tecnologías utilizadas, para el desarrollo del front-end se utilizó Angular 5. Para el servidor de back-end se utilizó la plataforma Java Empresarial. Finalmente como servidor de aplicaciones se decidió utilizar Wildfly y PostgreSQL como motor de base de datos.

Al finalizar este proyecto se logra crear un prototipo que cumple con los objetivos planteados, el cual posee una gran cantidad de información recolectada, incluyendo la disponible en la web del OGC de productos certificados. Para obtener la información y luego detectar posibles cambios se utiliza la técnica de Web Scraping. El prototipo estará disponible en un servidor de Firebase con el objetivo de que pueda ser utilizado por diferentes usuarios y de esta forma continuar con el enriquecimiento de los datos.

**Palabras claves:** Tecnologías de la información Geográfica, Web Scraping, Comunidad, Observatorio.

## Contenido

Resumen.....	2
1. Introducción .....	6
1.1 Motivación .....	6
1.2 Objetivos .....	7
1.3 Aportes del proyecto.....	8
1.4 Organización del documento .....	8
2. Marco Teórico .....	10
2.1 Conceptos relevantes.....	10
2.1.1 Datos e Información Geográfica (IG).....	10
2.1.2 Modelos y estructuras de datos.....	10
2.1.3 Sistema de Información Geográfica (SIG) .....	12
2.1.4 Tecnologías de Información Geográfica (TIG) .....	14
2.1.5 Open Source Geospatial Foundation (OSGeo).....	15
2.1.6 Organizaciones reguladoras de estándares y programas .....	16
2.1.7 Observatorio Tecnológico .....	22
2.2 Arquitectura de referencia para el proyecto .....	22
2.2.1 Categorización de los productos .....	22
2.3 Trabajos relacionados .....	23
2.3.1 Observatorios tecnológicos.....	23
2.3.2 Foros de comunidades .....	26
2.3.3 Herramientas de comparación de productos .....	28
2.3.4 Herramientas de calificación de productos .....	29
2.4 Tecnologías investigadas.....	30
2.4.1 Web Scraping .....	30
2.4.2 Java edición empresarial .....	33
3. Análisis.....	38
3.1 Usuarios del sistema .....	38
3.2 Relevamiento de requerimientos .....	39

3.2.1	Gestión de usuarios.....	39
3.2.2	Gestión de productos.....	41
3.2.3	Gestión de estándares.....	42
3.2.4	Actualización de datos .....	43
3.2.5	Foro .....	43
3.3	Casos de uso críticos .....	44
3.4	Entidades principales .....	45
4.	Solución planteada.....	47
4.1	Arquitectura de la solución .....	47
4.2	Diagrama de solución.....	49
4.2.1	Back-end.....	49
4.2.2	Base de datos del sistema .....	56
4.2.3	Componente de actualización de datos.....	56
4.2.4	Front-end.....	62
4.3	Implementación .....	63
4.3.1	Autenticación y Autorización .....	63
4.3.2	Comunidad de usuarios.....	67
4.3.3	Gestión de productos.....	71
4.3.4	Reportes .....	74
4.3.5	Notificaciones.....	75
4.3.6	Actualización de datos .....	76
4.4	Decisiones tomadas .....	78
4.4.1	Alcance de la solución .....	78
4.4.2	Tecnologías utilizadas.....	79
5.	Pruebas del sistema y gestión del proyecto.....	84
5.1	Pruebas del sistema .....	84
5.1.1	Pruebas del front-end .....	84
5.1.2	Pruebas del back-end.....	84
5.1.3	Carga y actualización de datos.....	84
5.2	Gestión del proyecto.....	85

5.2.1	Planificación inicial .....	85
5.2.2	Etapas desarrolladas .....	86
5.2.3	Desviación del proyecto .....	88
5.2.4	Herramientas utilizadas.....	89
6.	Conclusiones y trabajo a futuro .....	91
6.1	Conclusiones.....	91
6.2	Trabajo a futuro .....	92
7.	Referencias.....	93
	Apéndice 1: Documento de requisitos.....	97
	Apéndice 2: Casos de prueba del sistema.....	101

## 1. Introducción

Las Tecnologías de Información Geográfica (TIG) son un conjunto de Tecnologías de la Información y Comunicación (TIC) especializadas en la gestión y análisis de datos geográficos. Su capacidad para trabajar con la información espacial es un elemento importante para la comunicación, difusión e intercambio de conocimientos.

En la actualidad las TIG son herramientas básicas en la investigación, la planificación y la gestión del territorio. Éstas han demostrado su capacidad para resolver multitud de problemas con componente espacial, lo que hace que sean cada vez más utilizadas en la investigación, en el mundo profesional y en aplicaciones comerciales. [1]

### 1.1 Motivación

Dentro de las tecnologías de la información, las TIG constituyen un campo en creciente expansión, debido a su versatilidad, con aplicación en ámbitos tan distintos como el medio ambiente y los recursos naturales, la ordenación del territorio, el urbanismo, la planificación del transporte, la gestión y planificación de los servicios públicos, entre otros.

Debido a esa versatilidad, en los últimos años, ha aumentado considerablemente la variedad de TIG que se encuentran disponibles. Dichas tecnologías van teniendo cada vez mayor diversidad en cuanto a sus funcionalidades y el lugar que ocupan en el desarrollo de Sistemas de Información. Este auge se ve reflejado en el hecho de que cada vez se desarrollan más sistemas que tienen un componente geográfico. Por ejemplo, las aplicaciones móviles modernas suelen incluir un componente de localización para mejorar la experiencia de los usuarios, como es el caso de, la aplicación Uber, que permite la visualización de mapas para que los usuarios puedan conocer la ruta que sigue el conductor hasta su destino; este es solo uno de los muchos ejemplos disponibles de aplicaciones con componente geográfico.

Otro factor, que influye en el crecimiento de las TIG, es el hecho de que en la actualidad se maneja un mayor volumen de información geográfica, lo que lleva a que los Sistemas de Información deban convertirse en sistemas que permitan el manejo de dicha información.

En este contexto es que el Open Geospatial Consortium (OGC)<sup>1</sup> ha definido una arquitectura de referencia que sirve para enmarcar los estándares geoespaciales. Además el OGC tiene procesos de certificación de productos que cumplen con los estándares definidos. Este sistema produce una constante actualización en cuanto a las tecnologías que siguen un estándar y a su vez, se siguen sumando cada vez más estándares o actualizando las versiones existentes.

---

<sup>1</sup> Consorcio de los estándares geoespaciales abiertos <http://www.opengeospatial.org>

En la actualidad existen más de 900 TIG con más de 6000 estándares implementados [2]. Esto es producto del cambio en el modelo de los SIG, que son los sistemas que se construyen utilizando estas TIG. Estos sistemas evolucionaron de aplicaciones “Standalone” específicas a sistemas de información completos con múltiples componentes [3]. De esta forma las TIG pasaron a integrarse con todo tipo de sistemas, lo que llevó a nuevas categorizaciones de las mismas.

El gran número de TIG disponibles conlleva una gran cantidad de usuarios de las mismas. Estos usuarios o bien hacen uso de los propios SIG o en muchos casos son desarrolladores de software que construyen estos sistemas. Cualquiera de estos usuarios se enfrenta en muchas ocasiones al desafío de encontrar la TIG que mejor se ajuste a sus necesidades.

Cómo trasfondo de este notable crecimiento, descrito anteriormente, en cuanto a la variedad de TIG y al constante cambio de la información (por ejemplo, qué tecnologías implementan determinado estándar) surge una problemática que involucra directamente a los desarrolladores de sistemas. A estos desarrolladores se les plantea el desafío de realizar la integración de un componente geográfico en el sistema que se encuentran trabajando y ahí comienza la búsqueda de qué tecnología se puede utilizar, cuál es la que mejor se adecua a sus necesidades entre la gran variedad de tecnologías disponibles, qué tecnología implementa determinado estándar para integrar fácilmente al sistema existente o para comenzar un nuevo desarrollo.

## 1.2 Objetivos

En este contexto, nos proponemos realizar el estudio de Observatorios Tecnológicos, es decir, de plataformas que obtienen información actualizada de otras fuentes y que la utilizan para sacar conclusiones sobre la misma. Además se debe realizar el diseño e implementación de una propuesta para un observatorio de TIG. El Observatorio deberá contar con facilidades para la búsqueda y comparación entre las diferentes tecnologías que implementan estándares geoespaciales.

Este proyecto tiene como propósito general la realización de un Observatorio de TIG. Para cumplir con el mismo, se definen los siguientes objetivos específicos:

- Realizar un estudio de Observatorios Tecnológicos en general.
- Realizar un estudio de TIG: ver cuáles existen, lograr una categorización de las mismas y caracterizarlas.
- Definir los requerimientos para una plataforma que soporte un observatorio.
- Diseñar e implementar un prototipo de esta plataforma.
- Realizar una carga inicial de información y fuentes de información en la plataforma.

La aplicación tiene dos desafíos importantes: la amigabilidad de la interfaz de usuario y la recolección de los datos para la misma. Se espera realizar una carga inicial en base al relevamiento de observatorios y sitios de TIG. También se estudiarán mecanismos que faciliten la actualización automática de la información.

### 1.3 Aportes del proyecto

Los principales aportes que se lograron con este proyecto son:

- Conocimientos adquiridos gracias a la investigación realizada, este aporte se ve reflejado en el capítulo 2 de este informe, en donde se detallan los puntos que se estudiaron al comenzar con el proyecto y los que nutrieron a las integrantes del grupo con las herramientas necesarias para llevar a cabo el resto del mismo.
- Se centraliza gran parte de la información sobre los diferentes estándares y productos utilizados en el desarrollo SIG, facilitando la toma de decisiones a la hora de la implementación del software. Además se brinda información sobre el cumplimiento de los estándares según los productos y permite la comparación entre ellos, logrando un buen nivel de usabilidad y aportando información central para la toma de decisiones.
- Se crea un foro específico para la comunidad SIG, mejorando la integración y facilitando un canal para evacuar dudas sobre esta rama de los sistemas de información. De esta forma se ayuda al acercamiento de nuevos participantes fomentando mayor y mejor desarrollo de SIG.
- Además esta plataforma permite nuevas integraciones y facilita cambios en el sistema de carga de datos, de esta forma se evita la desactualización de la misma. Nuevos desarrolladores pueden integrar nuevas fuentes de información dinámicamente así como cambiar algunas configuraciones del sistema para mejorar la calidad de los datos. Por otro lado la plataforma es colaborativa, por lo que usuarios pueden ingresar nuevos datos desde la propia interfaz que luego serán aprobados por un administrador, de esta forma también se logra una mejor actualización de la información.
- Por último el observatorio cuenta con información sobre todos los estándares disponibles en el OGC (a la fecha del 30 de Julio de 2018) y muchos productos que implementan los mismos. Además el observatorio presenta reportes con información general de todos los datos del sistema, que permiten sacar conclusiones sobre las implementaciones de los estándares y la popularidad de las diferentes TIG.

### 1.4 Organización del documento

El documento está dividido en varios capítulos, cuyos contenidos se describen a continuación.

En el capítulo 2 se encuentra la información recabada durante la investigación realizada para el desarrollo del proyecto. La misma incluye tanto conceptos necesarios para la comprensión de este informe, así como también los diferentes antecedentes en los que se basó el desarrollo.

En el capítulo 3 se detallan todos los requerimientos que fueron considerados para el desarrollo de la plataforma. Así como también el alcance determinado y el modelo de datos utilizado.

Durante el capítulo 4 se describe la solución final a la que se llega. Incluye las tecnologías utilizadas y todas las decisiones tomadas en la implementación. Además detalla el funcionamiento y la implementación de los componentes centrales.

Luego el capítulo 5 muestra las pruebas realizadas en el sistema, y explica la carga inicial de los datos en el mismo. También describe algunos puntos claves en la gestión del proyecto por parte del grupo.

Por último el capítulo 6 expone las conclusiones y lista algunas funcionalidades importantes que se podrían agregar en el futuro.

## 2. Marco Teórico

Esta sección tiene como finalidad explicar los conocimientos previos necesarios para comprender el concepto central del proyecto, observatorio de TIG. Además se detalla la arquitectura de referencia de la que parte el mismo y se mostrarán brevemente algunos ejemplos de observatorios tecnológicos. Por último en esta sección se explicarán algunas metodologías y herramientas utilizadas para la implementación de la solución.

### 2.1 Conceptos relevantes

#### 2.1.1 Datos e Información Geográfica (IG)

Al hablar de IG puede surgir una confusión con el término “dato”. Estos parecen similares pero tiene significados distintos, se refiere a un “dato” como información sin procesar que se obtiene de la observación de la realidad. Mientras que la IG, a partir de estos datos, brinda un contenido más elaborado.

*“Los datos constituyen lo que se llama IG, cuando se almacenan sistemáticamente, se consultan, explotan, o cuando se combinan produciendo nueva información. La IG se crea, en última instancia, por el usuario” [4].*

Las características más significativas de los datos geográficos son las siguientes: posición (responde a la pregunta ¿Dónde?), atributos temáticos (responden a la pregunta ¿Qué?), tiempo (responde a la pregunta ¿Cuándo?), relaciones espaciales (proximidad a cierto fenómeno o existencia de características comunes) [4].

#### 2.1.2 Modelos y estructuras de datos

Una vez obtenidos los datos de la realidad, resulta fundamental la elección del modelo y la estructura de los mismos a la hora de construir un SIG. Estos datos hay que convertirlos en IG con un formato digital comprensible por el software elegido. Para este pasaje se crean reglas que permiten transformar datos de la realidad espacial compleja en información sintética digital. Son estas reglas el modelo de datos.

En cuanto a la estructura de los datos, existen dos grandes formas de estructurar la información real en un sistema de información: el modelo vectorial y el modelo raster.[5]

##### 2.1.2.1 Modelo vectorial

Este modelo representa los fenómenos espaciales por las coordenadas de los límites o perímetro que los definen. Esto logra que cada primitiva geométrica utilizada en la diagramación (ilustración 1) de estos fenómenos esté representada por uno o varios pares de coordenadas, de la siguiente manera:

- Elementos puntuales (punto): su posición quedaría definida por un único par de coordenadas.
- Elementos lineales (línea): se representan con al menos dos vértices diferentes, además el primero y el último deben ser siempre distintos. Su posición se define con los pares de coordenadas de cada uno de los puntos que definen los tramos que componen la línea.

- Elementos superficiales (polígono): se representan con los pares de coordenadas de todos los vértices que componen la poligonal del recinto cerrado. Se requiere al menos tres pares de coordenadas diferentes y no alineadas y en este caso sí es necesario que el primero y el último coincidan, cerrando de esa forma el polígono.



*Ilustración 1: Primitivas geométricas vectoriales: punto, línea y área [4]*

A partir de estas primitivas geométricas, cualquier fenómeno geográfico espacial es representable a través de vectores. Por ejemplo un fenómeno espacial del tipo construcción puede o bien ser puntual o superficial, y una ruta sería lineal o superficial.

Los SIG también pueden contar con geometrías complejas que estén compuestas por dos o más simples, como lo pueden ser un área con una parte hueca adentro [4].

#### *2.1.2.2 Modelo raster*

Este modelo consiste en convertir una imagen fotografía o modelo digital en un formato conseguido mediante superposición de una rejilla de unidades poligonales de igual forma y tamaño (pixel) sobre la imagen original. Este proceso es llamado rasterización y a cada pixel dentro de la imagen se le asocia un valor específico al área que representa en la misma, como puede ser un color o una altura.

La forma poligonal más utilizada para los pixeles es la cuadrada, y cuanto más pequeña sea la misma, más precisa será la representación digital de la imagen. La escala de la imagen raster en estos casos está dada por la longitud de los lados del pixel.

Para ubicar los pixeles en la rejilla, se les asigna un par de coordenadas que tienen origen (0,0) en la esquina superior izquierda de la imagen, ya que muchos equipos como escáneres, sensores espaciales e impresoras así lo definen.

Una misma imagen analógica puede tener asociadas diferentes imágenes digitales, como es el caso, de una fotografía aérea representada digitalmente por su componente RGB, su formato raster se descompone en tres capas en las que cada pixel representa el valor de la componente del color que corresponda según su capa: capa roja (R), capa verde (G) y capa azul (B).

Las imágenes raster se pueden clasificar según la información que se le asocia a los píxeles de la misma:

- Imagen digital térmica: cuando esa información es el valor de la temperatura
- Imagen en color natural: cuando esa información es el valor de la componente RGB
- Modelo Digital del Terreno (MDT): cuando esa información es el valor de la cota del terreno
- Modelo Digital de Elevaciones (MDE): cuando esa información es el valor de la cota calculada por el sensor al incidir la señal emitida sobre el primer obstáculo que encuentra. Para obtener esta información se necesitan complejos cálculos y procesamientos puramente manuales como la identificación y toma de puntos de referencia en el terreno.

Los tres primeros tipos de imagen raster, obtienen su información de forma automática a través de sensores digitales terrestres, aerotransportados pasivos o activos. En la mayoría de los casos, las imágenes obtenidas requieren ciertos tratamientos digitales, muchos de ellos automatizados hasta componer la imagen final raster, como se muestra en la ilustración 2 [4].



*Ilustración 2: Ejemplo de Imagen raster [4]*

### 2.1.3 Sistema de Información Geográfica (SIG)

A continuación se definirá lo que es un sistema de información, para luego explicar el concepto de SIG.

Un sistema de información (SI) es un conjunto de componentes integrados para recolectar, guardar y procesar datos para poder dar información y conocimiento al usuario. [6]

Existen diferentes definiciones de lo que es un SIG:

- *“... es un sistema informático utilizado para la colección, uso, análisis, presentación, manejo y distribución de datos espaciales.” [7]*

- “... sistema que integra hardware, software y datos para capturar, manejar, analizar y presentar todo las formas de información georreferenciada.”[3]
- “... es un tipo especial de sistema de información usado para ingresar, almacenar, procesar, analizar y visualizar datos geoespaciales e información para ayudar a la toma de decisiones.”[8]

En estas definiciones se puede ver que un SIG es un caso particular de un sistema de información en el que cambia el tipo de datos que maneja. En los primeros años que surgió este concepto (1960) los SIG eran vistos como una simple herramienta para tomar decisiones, pero eso fue cambiando drásticamente en los últimos años, evolucionando a una ciencia de información geográfica. [8]

En la actualidad, muchos sistemas de información convencionales, descubren tener mayor utilidad y mejor usabilidad, integrando la dimensión espacial de sus datos a los sistemas, generando una mayor cantidad de SIG que en sus orígenes.

En la actualidad se utilizan SIG en diferentes rubros, con muchas aplicaciones diferentes. A continuación se listarán algunos de sus usos: [9]

1. Compañías lo utilizan para determinar dónde ubicar una nueva sucursal
2. Científicos lo utilizan tanto para comparar estadísticas de la población, como lo puede ser el consumo de agua, como para seguir los patrones de migración de ciertos animales.
3. Empresas estatales lo utilizan para planear respuesta en caso de un desastre natural, un SIG podría ayudar a las autoridades a ver qué barrios tienen más riesgo, dónde ubicar refugios e incluso las rutas a seguir para las evacuaciones.
4. Ingenieros usan SIG para diseñar, implementar y manejar la comunicación de las redes móviles y de internet. Otros pueden usarlo para diseñar redes de tráfico vial.[4]

Estos son solo algunos ejemplos, pero demuestran el potencial que se le puede dar a los SIG y la importancia de los mismos en la actualidad.

### 2.1.3.1 Componentes principales

En las últimas décadas los SIG dejaron de ser un software “Standalone” para pasar a tener una típica arquitectura en capas de las aplicaciones empresariales. Estas capas serán de utilidad para clasificar luego las diferentes tecnologías.

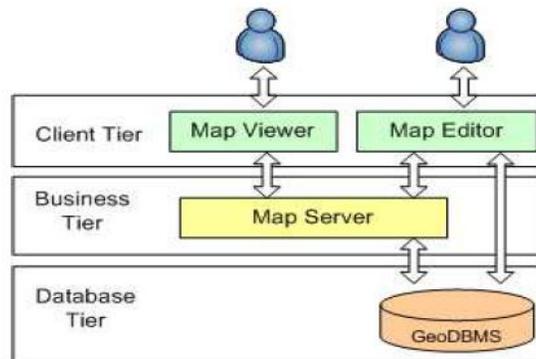


Ilustración 3: Arquitectura en capas de un SIG típico [3]

A continuación se describen brevemente las diferentes capas presentadas en la ilustración 3.

- Un GeoDBMS da soporte al almacenamiento y consulta de objetos geográficos.
- Un Map Server es un componente de software que generalmente se ejecuta sobre un servidor web o servidor de aplicaciones. Su objetivo principal es proveer a clientes de Internet mapas generados dinámicamente.
- Un Map Viewer es una aplicación cliente que despliega un mapa en una interfaz de usuario.
- Un Map Editor además de ser un Map Viewer permite crear mapas, modificar objetos y otro tipo de datos.

#### 2.1.4 Tecnologías de Información Geográfica (TIG)

TIG es un término que comprende un conjunto de herramientas utilizadas para la recolección, almacenamiento, edición, consulta, manejo, análisis y presentación de información geográficamente referenciada para relacionar fenómenos y comprender las relaciones espaciales entre los mismos. [10]

En el desarrollo de los SIG se involucran diferentes tipos de tecnologías. Estas se podrían dividir de la siguiente forma: hardware, software y servicios. Las últimas dos forman son las que consideramos TIG. En la ilustración 4 se muestra la evolución en la inversión por parte del gobierno de estados unidos en cada una de las categorías descritas anteriormente. Este aumento en la inversión lleva a una evolución de los SIG y las TIG.

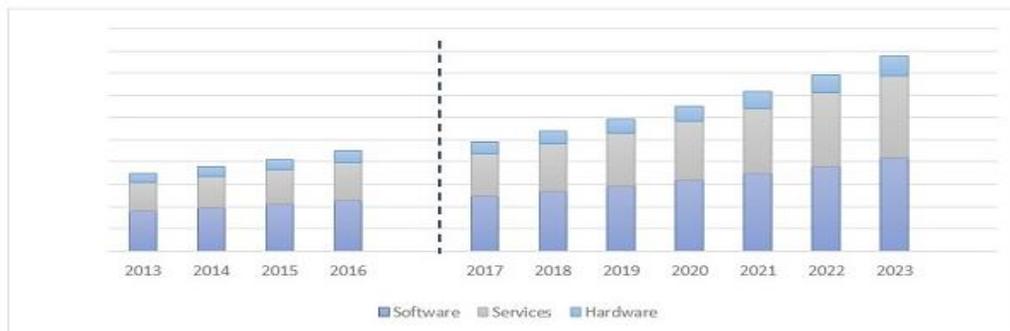


Ilustración 4: Crecimiento relativo de los componentes hardware, software y servicios de la industria SIG global [11]

Además del crecimiento que hubo hasta ahora de esta inversión se proyecta que el mismo seguirá creciendo varios años más, como lo muestra la ilustración 4 en los años posteriores. Esta inversión fomenta un aumento en la cantidad de componentes y va evolucionando los SIG, pero al mismo tiempo implica un desafío para los desarrolladores. Estos deben estar constantemente actualizados para poder sacar el mejor provecho de este auge de la tecnología.

Las TIG como otros softwares pueden calificarse entre código abierto o propietario. Además en el caso particular de las TIG las mismas suelen especializarse en alguna de las capas presentes en la arquitectura clásica descrita anteriormente. También parece relevante la información de las fechas de creación de las tecnologías y fechas de sus actualizaciones, ya que en este rubro hay cambios muy vertiginosos y fácilmente pueden quedar desactualizadas.

### 2.1.5 Open Source Geospatial Foundation (OSGeo)

La OSGeo es una fundación sin fines de lucro que tiene como misión fomentar la adopción de tecnologías geoespaciales abiertas. Esta fundación de software es inclusiva y se basa en una filosofía abierta y en un desarrollo del software basado en una comunidad participativa.

La OSGeo provee apoyo financiero, organizacional y legal a la comunidad geoespacial de código abierto. Además es una entidad independiente a la cual los miembros de la comunidad pueden brindarle tanto código como fondos y se asegura que sus colaboraciones serán usadas para el beneficio público. La fundación también provee un foro común y una infraestructura compartida para mejorar la colaboración entre distintos proyectos.

Entre sus objetivos se destacan: [12]

- Proveer recursos para los proyectos de la fundación
- Fomentar la interoperabilidad con estándares abiertos y comunitarios
- Asegurar interoperabilidad entre los proyectos de la fundación
- Asegurar calidad e innovación en sus proyectos

Con el fin de llevar a cabo su misión la OSGeo promueve el acceso libre a los geodatos así como el uso del software de código abierto en la industria. A su vez también fomenta la implementación de estándares abiertos y la interoperabilidad basada en estándares en los proyectos de la fundación. Por otro lado promueve la comunicación y cooperación entre las comunidades de OSGeo en diferentes lenguajes de programación y sistemas operativos y apoya el uso del software de OSGeo en educación.

Otro de los servicios que brinda la OSGeo es el de facilitar el acceso a software SIG, publicando los sitios para descargar y probar los distintos softwares. Entre los productos que se encuentran disponibles para su descarga se destaca la máquina virtual OSGeoLive, que permite probar una gran variedad de software geoespacial de código abierto sin instalar nada. Está compuesto completamente de software libre, lo que permite que se duplique y distribuya libremente. Ésta es una excelente recopilación anual de las últimas tendencias en las aplicaciones más usadas entre los servidores y clientes de IG libres y gratuitos. Posee guías de acceso rápido que resultan de gran ayuda (presentadas en español y otros idiomas) que orientan al usuario sobre cómo usar los softwares que están contenidos en la distribución.

## 2.1.6 Organizaciones reguladoras de estándares y programas

### 2.1.6.1 *World Wide Web Consortium (W3C)*

Es una organización, fundada en 1994 la cual se encarga de promover estándares relativos a la Web, entre los que se encuentran: HTML, XML, XHTML, CSS, XML Schema, SOAP, WSDL, SVG, GeoRSS entre otros. [13]

### 2.1.6.2 *Comité ISO*

La Organización Internacional de Normalización (International Organization for Standardization, conocida por las siglas ISO) es una organización independiente y no-gubernamental formada para la creación de estándares internacionales compuesta por diversas organizaciones nacionales de estandarización.

El Comité ISO es el grupo responsable de desarrollar, emitir y difundir los documentos ISO sobre Gestión de la Calidad. Las Normas buscan ser una herramienta para la mejora de las organizaciones.

Las Normas Internacionales ofrecen especificaciones de clase mundial para productos, servicios y sistemas, para garantizar la calidad, la seguridad y la eficiencia. Son fundamentales para facilitar el comercio internacional.

ISO ha publicado 21740 Normas Internacionales y documentos relacionados, que abarcan casi todas las industrias, desde la tecnología hasta la seguridad alimentaria, la agricultura y la sanidad. [14]

### 2.1.6.3 *Open Geospatial Consortium (OGC)*

El OGC es un consorcio internacional sin fines de lucro, que tiene como objetivo el desarrollo de estándares abiertos, interoperables y de calidad para el intercambio de información geográfica. Actualmente el OGC cuenta con más de 40 estándares, 500 miembros, y cientos de aplicaciones y productos, a su vez tiene una amplia comunidad de usuarios, socios de alianzas y actividades colaborativas con ISO y organizaciones de desarrollo de estándares. Entre los principales miembros se encuentran universidades, gobiernos y empresas comerciales, organizaciones sin fines de lucro y de investigación. El OGC proporciona un proceso de consenso utilizado por las comunidades para resolver problemas relacionados con la creación, la comunicación y el uso de la información espacial. Los estándares de OGC son utilizados en una amplia variedad de dominios incluyendo medio ambiente, defensa, salud, agricultura, meteorología y desarrollo sostenible entre otros. [15]

Las actividades del OGC se articulan alrededor de tres programas: a) Programa de especificación, b) Programa de interoperabilidad y c) Programa de alcance y adopción.

El consorcio dispone de una infraestructura tecnológica tal (sistema de gestión de contenidos en la web, foros, sistemas para videoconferencias, newsletters, etc.) que le permite con facilidad el trabajo en equipo, aun cuando sus miembros se encuentren dispersos por el mundo. Esta infraestructura, así como el consorcio en general, se financia de diversas formas. Entre ellas se encuentran las cuotas de sus miembros, licencias de certificación, ingresos generados por la gestión de los diversos proyectos en los que participa, entre otros. [4]

#### 2.1.6.4 *El programa de especificación*

Define el proceso mediante el cual un estándar OGC se formaliza. Abordan el problema de la interoperabilidad desde un punto de vista teórico, y los resultados deben ser aprobados mediante un consenso por parte de los miembros. Quién coordina esto es el comité técnico (Technical Committee, TC), el cual decide cuáles son aquellos trabajos que van a realizarse y sobre la validez de los resultados de los mismos.

El hecho que dispara la elaboración de un estándar OGC puede darse, por ejemplo, cuando un miembro detecta una necesidad determinada a partir de los resultados del programa, y en base a esto elabora una propuesta para cumplir con dicha necesidad. El TC forma un grupo de trabajo de estándares (SWG) ad hoc para cada propuesta. Éste último es el encargado de concretar la propuesta, utilizando como guía para ello un conjunto de documentos que conforman un modelo para el desarrollo de especificaciones, establecido por OGC. El resultado final de este proceso es un conjunto de documentos técnicos y de diferentes propuestas de acciones.

La creación de grupos de trabajos temáticos (DWG), en donde se discuten cuestiones relacionadas con interoperabilidad de información geoespacial en sectores profesionales, también es una facultad del TC. Los resultados de estos trabajos también deben ser formalizados, y sus propuestas deben ser enviadas al TC.

El comité de planificación es quién tiene la capacidad de aprobar y adoptar las recomendaciones del TC y hacerlas públicas como estándar OGC. [4]

#### 2.1.6.5 *El Programa de interoperabilidad*

Permite definir, documentar, mejorar y poner a prueba las especificaciones actuales de OGC. Consigue de manera rápida y práctica el desarrollo de estudios y prototipos de infraestructuras interoperables, basados en estándares OGC y especificaciones, presentados por los participantes de los proyectos. También proporciona un programa de servicio de pruebas de conformidad (CITE) que permite evaluar y certificar la compatibilidad de una aplicación con un estándar determinado. [4]

#### 2.1.6.6 *Los estándares del OGC*

La materialización de un estándar en una especificación es el principal producto de los programas OGC. Cada uno de ellos tiene como objetivo solucionar aspectos de interoperabilidad, para que su posterior implementación en algún producto y/o servicio no dependa del productor ni de la implementación. En una arquitectura cliente-servidor, por ejemplo, esto debe permitir que no sea necesario el adecuamiento de los códigos que implementan a cada actor, para que los mismos puedan intercambiar datos de forma exitosa.

También se trabaja con estándares de tipo abierto, lo cual implica que los mismos deben estar disponibles de forma pública y libre, sin costo alguno, y el mismo debe estar aprobado. [4]

#### 2.1.6.7 *Clasificación de los estándares OGC*

El OGC produce y revisa estándares desde hace más de dos décadas. De manera muy general, los estándares OGC pueden agruparse en cuatro grandes categorías: estándares de codificación y datos, estándares de servicios web, estándares para API y estándares para clientes web (ilustración 5).

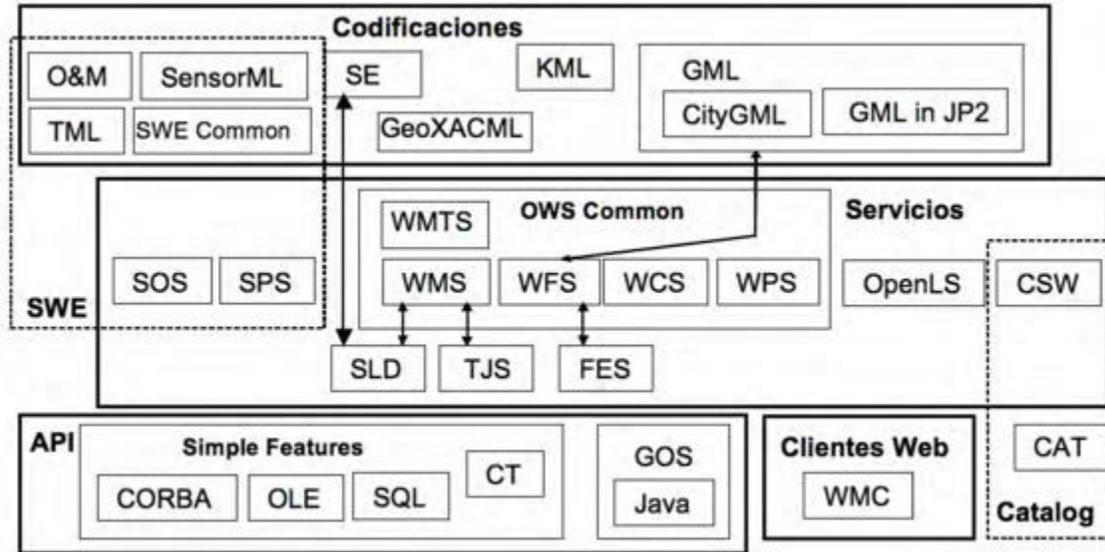


Ilustración 5: Clasificación de los estándares OGC [4]

### Estándares OGC para visualizar datos

Originalmente aplicados en la creación de portales que visualizaban mapas incrustados en páginas web, hoy sus aplicaciones se han extendido a diversos tipos de productos, como pueden ser los SIG de escritorio o aplicaciones para móviles, por ejemplo.

A continuación se describen algunos de ellos y sus operaciones principales [4]:

- **WMS:**
  - GetMap: es obligatoria y obtiene un mapa de una zona definida por su ámbito y dimensiones en píxeles a partir de los datos de una o varias capas.
  - GetFeatureInfo: es opcional y devuelve información más detallada sobre un punto específico del mapa.
- **SLD y SE:** permiten al usuario definir nuevos estilos de simbolización. SLD extiende a WMS, que describe ampliaciones de la petición GetMap para solicitar estilos definidos por el usuario a un servidor WMS. SE es un lenguaje transversal de codificación de estilos, codificado en XML y aplicable tanto al modelo raster como al vectorial.
- **TJS:** permite al usuario enriquecer los mapas disponibles en un servicio WMS. Define un formato XML para tablas de datos denominado GDAS
  - DescribeFrameworks, DescribeDatasets, DescribeData y GetData: permiten describir y obtener las tablas de datos disponibles por el servicio.
  - DescribeJoinAbilities, DescribeKey y JoinData: permiten enlazar una tabla GDAS con un conjunto de entidades espaciales y crear una nueva capa en un servidor WMS asociado.

- **WMTS:** a diferencia de WMS, discretiza el espacio en un conjunto de niveles de zoom predefinidos, donde para cada uno de ellos define una matriz regular de teselas.
  - GetTile: obtiene las teselas.
  - GetFeatureInfo: peticiones incorporadas por este estándar.

### Estándares OGC para el acceso a datos

Se distingue entre los estándares que utilizan el modelo de datos vectorial y el modelo de datos raster. Por lo general, sólo permiten la descarga de datos, salvo por aquellos que son llamados transaccionales, quienes permiten la actualización de los mismos también.

A continuación se describen algunos de ellos y sus operaciones principales [4]:

- **WFS:** permite el acceso a datos vectoriales en formato GML.
  - DescribeFeature: permite obtener el esquema de aplicación de una o varias FeatureTypes.
  - GetFeature, GetFeatureWithLock y GetGMLObject: permite descargar un documento GML, con los datos vectoriales.
  - Concepto Transaction: el estándar tiene la capacidad transaccional de crear, editar y borrar entidades vectoriales del servidor, pasando a estar disponibles inmediatamente para el resto de los usuarios.
- **WCS:** permite acceder a datos raster. Considera los raster multidimensionales.
  - DescribeCoverage: permite obtener una descripción de cada una de las capas y todas sus dimensiones.
  - GetCoverage: permite obtener un subconjunto de los datos de las capas escogiendo una envolvente espacial y un rango dimensional.
- **SOS:** define una interface para obtener información acerca de sensores y sus observaciones. Ayuda a sistemas de observación a establecer un web service con la información acerca de las plataformas y sus sensores, las observaciones en tiempo real (además de la histórica), y datos brutos o procesados.
  - DescribeSensor
  - GetObservation

### Estándares OGC para codificación de datos

Los estándares OGC, en su mayoría, utilizan el estándar de W3C XML como base de su codificación. El mismo define un formato de texto controlado por un documento de esquema XSD.

A continuación se describen algunos de ellos y sus características principales:

- **KML:** es la codificación XML nativa de Google Earth, inherentemente 3D y pensado para la presentación final de datos (incluyendo incluso la posición del observador) y no para su almacenado y análisis.
- **GML:** es una codificación XML inicialmente concebida para el intercambio de datos vectoriales. El GML proporciona herramientas para la definición de entidades vectoriales 2D y 3D, simples o complejas. Un conjunto de datos vectoriales codificado en GML, requiere la definición previa de los tipos de entidades a partir de propiedades espaciales definidas por el GML y de atributos temáticos definidos con tipos básicos XML o por tipos complejos personalizables. Los datos GML deben cumplir y validarse con el esquema de aplicación definido. En la práctica, esto significa que la estructura de un documento GML genérico resulta impredecible por las aplicaciones que deben leerlo, lo que se soluciona a partir de perfiles de aplicación GML.
  - **GML-SF:** perfil GML para entidades simples proporciona un conjunto de restricciones a la flexibilidad del GML y facilitan su uso práctico. *“Establece tres niveles de restricciones (SF-0, SF-1 y SF-2). Por ejemplo, el nivel más restrictivo (SF-0) limita las entidades gráficas a una lista de primitivas que permiten definir puntos, líneas o polígonos y los tipos de atributos a tipos XML básicos y la cardinalidad a 1.”*[4]
  - **Perfiles de Aplicación Específicos:** para aplicaciones concretas.

### Estándares para sensores [4]:

- **SWE Common:** define el modelo conceptual y la codificación XML para la descripción de los datos relacionados con sensores.
- **O&M:** define un modelo conceptual, que determina, que una medición es la acción de estimar el resultado de una propiedad de una entidad de interés de acuerdo a un proceso.
- **SensorML:** permite representar la descripción de elementos (sensores, plataformas u otros procesos). Como por ejemplo cualquier parámetro asociado a un procedimiento de medida.

### Estándares OGC para catálogos y registros

*“Los servicios de catálogo ofrecen la capacidad de publicar y localizar colecciones de metadatos, tanto sobre datos como sobre servicios, así como sobre otras fuentes de información. Los metadatos recogidos y ofrecidos por estos catálogos contienen información que describen los recursos disponibles”*[4] (ilustración 6).

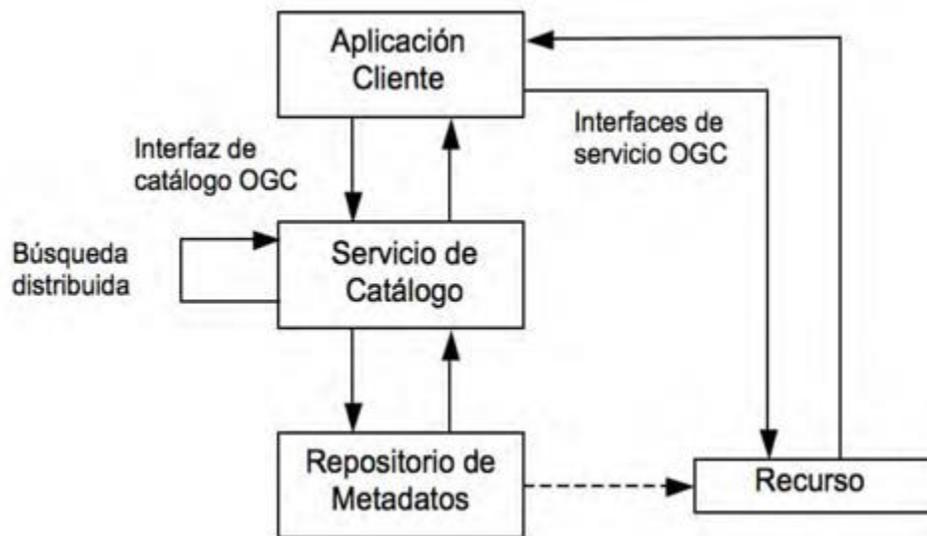


Ilustración 6: Arquitectura de referencia de OGC para servicios de catálogo [4]

Esta información debe ser suficiente como para que se puedan intercambiar datos entre diferentes sistemas y personas. Haciéndolos necesarios para la localización de recursos geoespaciales dentro de una comunidad. Su papel en las infraestructuras de datos espaciales (IDE) es esencial, gracias a la facilidad que brindan para automatizar y mejorar la eficiencia de la búsqueda de recursos geográficos. [4]

#### Estándares OGC para el procesamiento de datos

Los estándares OGC hacen posible que la información espacial pueda distribuirse en Internet volviendo sumamente importante las capacidades de los servicios web de visualización y consulta de esta información.

A continuación se describen algunos de ellos y sus características principales [4]:

- **WPS:** permite publicar, localizar y usar procesos geoespaciales, posibilita describir datos de entrada y salida así como los modos de proceso y acceso a esa información. Permite también, que procesos específicos sobre datos geoespaciales, se puedan publicar como servicios web para ser usados por clientes distribuidos en la web.
- **WCPS:** se basa en WPS y formaliza las operaciones de procesamiento raster.

### 2.1.7 Observatorio Tecnológico

Un observatorio tecnológico es un espacio que permite gestionar en tiempo real las fuentes de información (públicas y privadas) en un área específica, transformando los datos en información útil expresada en resultados que respondan a los diferentes niveles de toma de decisiones estratégicas, tácticas u operacionales de la empresa. Puede incluir estudios para:

- Alertas automatizadas por tecnologías, empresas, instituciones, expertos, repercusión técnica según sea el interés
- Boletines de Vigilancia Tecnológica por proyectos o tecnologías
- Informes de Valorización Tecnológica
- Informes de Identificación de Terceros

En lo que respecta a los observatorios tecnológicos existen diferentes tipos de definiciones. Lo que todos tienen en común es la importancia de tener la información actualizada, ya que hoy en día los cambios en la tecnología son vertiginosos y es importante mantenerse actualizado. También, resulta relevante exponer la información para que personas no especializadas puedan entenderla. En muchos casos también, se estudian en base a las nuevas tecnologías los impactos que tienen en determinados sectores. [16]

## 2.2 Arquitectura de referencia para el proyecto

El proyecto surge como una continuación a un trabajo realizado en Facultad de Ingeniería en 2016 [17], en el que se relevaron datos de los estándares del OGC y las tecnologías que cumplían los mismos. Este trabajo brinda la arquitectura de referencia a utilizar en la solución así como la carga de datos inicial del sistema.

En este trabajo se relevaron únicamente los estándares del OGC y luego las tecnologías son categorizadas según el nivel de cumplimiento de los mismos y se agregan los certificados de cumplimiento únicamente de las tecnologías que obtuvieron alguno de estos del año 2010 en adelante [17].

### 2.2.1 Categorización de los productos

La categorización viene dada por las capas definidas en la sección 2.1.3.1 con algunas diferencias:

- Map Server
- GeoDBMS
- Componentes GUI (Desktop y Web): estos equivalen a los pertenecientes a la client tier de la ilustración 3, es decir, los de tipo Map Viewer y Map Editor.
- Librerías: esta categoría es transversal a todas las capas.

Además los productos están clasificados según el nivel de cumplimiento de los estándares de OGC con los siguientes niveles [17]:

- Con certificados: además de implementar los estándares han pasado las pruebas de cumplimiento de OGC y también obtuvieron la Licencia Trademark OGC.
- Implementador referente: su implementación debe ser gratuita, completamente funcional, haber pasado las pruebas de una clase de conformidad asociada a un estándar de implementación y estar disponible públicamente para que los desarrolladores puedan realizar pruebas. La implementación necesita pasar la prueba cada año para mantener su buen estatus en condición de implementación de referencia.
- Implementa estándar pero sin certificado: el producto sostiene implementar algún estándar pero el mismo no tiene certificaciones.

## 2.3 Trabajos relacionados

En esta sección, por un lado, se describen algunos observatorios tecnológicos existentes en la web, y por otro, se describen sitios con funcionalidades similares a las del observatorio, como son: foros de comunidad y sitios de comparación y calificación.

### 2.3.1 Observatorios tecnológicos

#### **AGESIC - Agencia de Gobierno Electrónico y Sociedad de la Información y el Conocimiento**

El AGESIC cuenta con un observatorio tecnológico con el cual se busca generar información y propuestas para innovar y mejorar los servicios al ciudadano. El mismo se nutre de varias fuentes, entre ellas se encuentra el análisis de publicaciones generadas por referentes internacionales (Gartner [18], Forrester [19], Google Research [20]) y también por referentes específicos en su área que es el gobierno electrónico (Corea del Sur, Holanda, Reino Unido). Este observatorio tiene un proceso claro para la validación y las pruebas que se realizan en las diferentes herramientas observadas para luego publicar resultados de las mismas. [21] Este proceso se describe en la ilustración 7.



Ilustración 7: Proceso de Observación y generación. \*POC: Pruebas de Concepto [21]

### OVTT - Observatorio Virtual de Transferencia de Tecnología

Es un proyecto institucional desarrollado por la Universidad de Alicante (UA), en colaboración con el Banco Santander (Santander Universidades), para impulsar la transferencia de conocimiento y tecnología, la innovación y el emprendimiento en Iberoamérica. Su misión es ser un instrumento de mediación tecnológica en red que, a través de esta plataforma virtual, facilita el acceso a información científica y tecnológica disponible en Internet a empresarios, investigadores y emprendedores, para que la apliquen eficientemente en su actividad profesional y genere oportunidades de colaboración e innovación en Iberoamérica. Promoviendo, con ello, la importancia de implementar prácticas de vigilancia tecnológica y el uso avanzado de Internet en proyectos de investigación y desarrollo emprendedores. [22]



Ilustración 8: Interfaz gráfica del observatorio virtual de transferencia de tecnología [22]

### EITO - Observatorio Europeo de Tecnologías de la Información

Desde 1993, el Observatorio Europeo de Tecnologías de la Información ofrece información de alta calidad y actualizada sobre los mercados europeos y mundiales de la tecnología de la información, las telecomunicaciones y la electrónica de consumo. El EITO es administrado por Bitkom Research GmbH, una filial al 100% de BITKOM, la Asociación Federal de Tecnología de la Información, Telecomunicaciones y Nuevos Medios en Alemania. [23]

**eito** | european IT observatory

Basket  
Your basket is empty.

+49 30 276 76 - 560

Market Reports About EITO Press Partners & Sponsors Research Services Contact

Home » Research Services » Services

### Services

As partner for ICT market research, Bitkom Research offers customized market research services and supports customers in the following areas:

- Conception of consumer and business surveys
- Preparation of market analyses and statistics
- Execution of complex market research studies
- Consulting, presentations and workshops

Strengths that you can profit from:

- Access to a unique expert network via close relationship to BITKOM e.V.
- High acceptance and media coverage of the published study results
- Best value for money through our trim, efficient organization
- Experience in multi-client studies

The company **presentation** provides you with an overview of the Bitkom Research services.

Ilustración 9: Interfaz gráfica del observatorio europeo de tecnologías de la información [23]

## Observatorio digital de las áreas protegidas

El observatorio digital de las áreas protegidas es un conjunto de servicios web y aplicaciones que pueden usarse principalmente para acceder, monitorear, reportar y posiblemente predecir el estado general y la presión a la que están sometidas las áreas protegidas. Este observatorio ofrece sus datos a los usuarios para que puedan usarlos en sus sistemas, además ofrece reportes sobre los mismos. Los datos los obtiene a partir de instituciones que contribuyen con el observatorio. Además cuenta con herramientas online diseñadas para mejorar la toma de decisiones en las áreas protegidas. Por último también provee un framework abierto para ayudar a otras instituciones a desarrollar sistemas para preservar las áreas protegidas. [24]

Privacy statement Legal notice Cookies Search

European Commission

JOINT RESEARCH CENTRE  
DIGITAL OBSERVATORY FOR PROTECTED AREAS (DOPA)

European Commission > EU Science Hub > DOPA

HOME POLÍTICA APLICACIONES SERVICIOS FACTSHEETS DOCUMENTOS MAPAS Y DATOS PROYECTOS NOTICIAS Y EVENTOS CONTACTOS

### Bienvenidos al Observatorio Digital para las Áreas Protegidas

"Estamos ahogándonos en información, pero sedientos de conocimiento. A partir de ahora, el mundo estará a cargo de los sintetizadores, personas capaces de reunir la información adecuada en el momento adecuado, tener un pensamiento crítico y tomar decisiones acertadas sobre las cuestiones importantes."  
E. Wilson, 1998, Consilience

hábitats clima especies conservación presiones conectividad

Ilustración 10: Interfaz gráfica del observatorio digital para las áreas protegidas [24]

## Observatorio tecnológico de la Unión Internacional de Telecomunicaciones

La Unión Internacional de Telecomunicaciones mantiene un observatorio tecnológico para realizar un seguimiento de las tecnologías de la información y la comunicación emergentes. Su cometido principal es el de realizar informes sobre las últimas tecnologías y además evaluar el impacto de las mismas en los países desarrollados o en desarrollo. [25]

Todos estos observatorios comparten algunas características en común, como lo son colaborar con diferentes comunidades, ayudar a la toma de decisiones brindando datos actualizados e incluso ayudar a la visualización de conclusiones a través de reportes de interés sobre estos datos.



Ilustración 11: Interfaz gráfica del observatorio tecnológico de la Unión Internacional de Telecomunicaciones [25]

### 2.3.2 Foros de comunidades

#### Slashdot [26]

Es un sitio web de noticias sociales sobre ciencia, tecnología y política que son enviadas y evaluadas por los usuarios y editores del sitio. Cada noticia tiene una sección de comentarios adjunta a la que los usuarios pueden agregar comentarios en línea y los editores son los encargados de aceptar o rechazar cada contribución.

Los propios usuarios de Slashdot envían resúmenes de noticias con enlaces a las fuentes y cada noticia se convierte en el tema de una conversación entre usuarios. Otros miembros actúan como moderadores, moderando las discusiones, califican elementos en una escala (generalmente) de 1-5 y asignan una evaluación.

**Slashdot** Stories Firehose > All Popular Polls / Deals Submit Search

Topics: Devices Build Entertainment Technology Open Source Science YRO

Follow Slashdot blog updates by subscribing to our blog RSS feed

**chip stop** Free Shipping! Electronic Components and Semiconductors

Have you META-MODERATED today? Sign up for the Slashdot Daily Newsletter! DEAL: For \$25 - Add A Second Phone Number To Your Smartphone for life! Use promo code SLASHDOT25.

**Lawyers Are Sending Mobile Ads To Patients Sitting In Emergency Rooms**

Posted by BeauHD on Tuesday May 29, 2018 @07:40PM from the digital-ambulance-chasers dept.

An anonymous reader quotes a report from NPR:

Patients sitting in emergency rooms, at chiropractors' offices and at pain clinics in the Philadelphia area may start noticing on their phones the kind of messages typically seen along highway billboards and public transit: [personal injury law firms looking for business by casting mobile online ads at patients](#). The potentially creepy part? They're only getting fed the ad because somebody knows they are in an emergency room. The technology behind the ads, known as geofencing, or placing a digital perimeter around a specific location, has been deployed by retailers for years to offer coupons and special offers to customers as they shop. Bringing it into health care spaces, however, is raising alarm among privacy experts.

[Law firms](#) and [marketing companies](#) from Tennessee to California are also testing out the technology in hospital settings. "Is everybody in an emergency room going to need an attorney? Absolutely not," Kakis says. "But people that are going to need a personal injury attorney are more than likely at some point going to end up in an emergency room." The advertisers identify someone's location by grabbing what is known as "phone ID" from Wi-Fi, cell data or an app using GPS. Once someone crosses the digital fence, Kakis says, the ads can show up for more than a month -- and on multiple devices.

f t in g+ reddit ads court mobile

Ilustración 12: Interfaz gráfica de slashdot [26]

## Stackoverflow [27]

Es un sitio web de preguntas y respuestas dedicadas especialmente para programadores y profesionales de la informática con el fin de que los desarrolladores aprendan y compartan sus conocimientos.

Todos los usuarios pueden hacer y responder preguntas así como votar por las preguntas y por sus respuestas, cuando se vota por una pregunta, el usuario puede calificarlas como más relevante o menos relevante; por otra parte, cuando se vota por las respuestas, éstas pueden ser más acertadas o menos acertadas.

**stackoverflow** Questions Developer Jobs Tags Users Search...

All Questions newest 983 featured frequent votes active unanswered

0 votes  
0 answers  
2 views  
**How to limit process memory in Erlang**  
I would like to know how I can limit the maximum memory (heap) a process can use to allocate objects. I looked into spawn\_opt with max\_heap\_size as mentioned here but I keep getting a badarg error. Is ...  
memory-management erlang asked 1 min ago yoda 11 • 1

0 votes  
0 answers  
2 views  
**Weird matTooltip behavior with toolbar-button and nested button**  
I am experiencing an odd behavior - though perhaps it works as intended (which is the reason why I am not filing a bug right away) or I am using mat-menu-item-directive in a place I should not? When ...  
angular-material angular6 asked 1 min ago Igor 490 • 2 • 7 • 29

0 votes  
0 answers  
2 views  
**Understanding a Little More About Test Styles**  
Could you tell me the difference or between the following test styles? Data - driven, Workflow Scenario Exploratory I have been searching for something about this for a while, but I couldn't find ...  
testing qa asked 1 min ago user6868934 1

Ilustración 13: Interfaz gráfica de stackoverflow [27]

### 2.3.3 Herramientas de comparación de productos

#### CPU Boss[28]

Es una plataforma en línea que permite a los usuarios encontrar y comparar las especificaciones de los procesadores de computadoras en la cual se resumen las principales características de cada CPU en una sola página.

Dicha plataforma cumple el rol de facilitar la búsqueda de una CPU, surgió para resolver el problema de “¿Que computadora debería tener?”

La investigación de las CPU puede ser un proceso que consuma mucho tiempo para los usuarios. Estos navegan la web y mantienen varias búsquedas en diferentes páginas del navegador para comparar las especificaciones disponibles de las CPU. CPU Boss es útil porque captura una cantidad enorme de especificaciones, benchmarks, y comentarios de la web que luego centraliza y ordena en un solo sitio web.

El objetivo principal de dicha plataforma es ahorrar tiempo a los usuarios en la investigación a través de la web las distintas CPU existentes. CPU boss no solo permite comparar diferentes CPU sino que también permite realizar búsquedas por diferentes criterios como puede ser el precio, clock speed, entre otros.

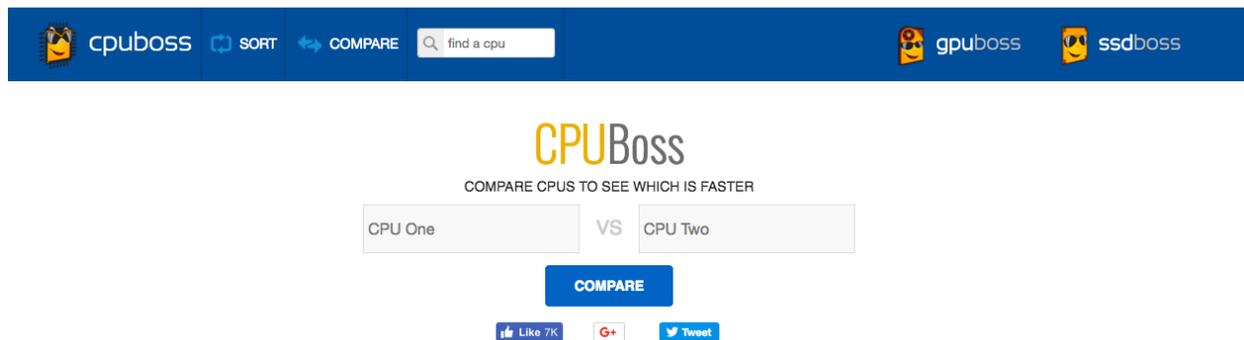


Ilustración 14: Interfaz gráfica de CPU Boss [28]

#### VERSUS.COM [29]

Es una herramienta de comparación en línea para teléfonos móviles, tabletas, cámaras, videocámaras y ciudades.

Es el primer y único proveedor del mundo de algoritmos multi-idioma y big-driven que genera contenido de texto natural totalmente automatizado. Con un gran enfoque en la interfaz de usuario, la plataforma de comparación convierte los datos sin procesar en lenguaje natural y simplifica la complejidad.

Al usar Versus, los consumidores pueden ver los resultados en un formato simple y claro, sin 'sobrecarga de información' y, en última instancia, permitiendo que los usuarios obtengan datos de productos complejos de manera rápida y eficiente.

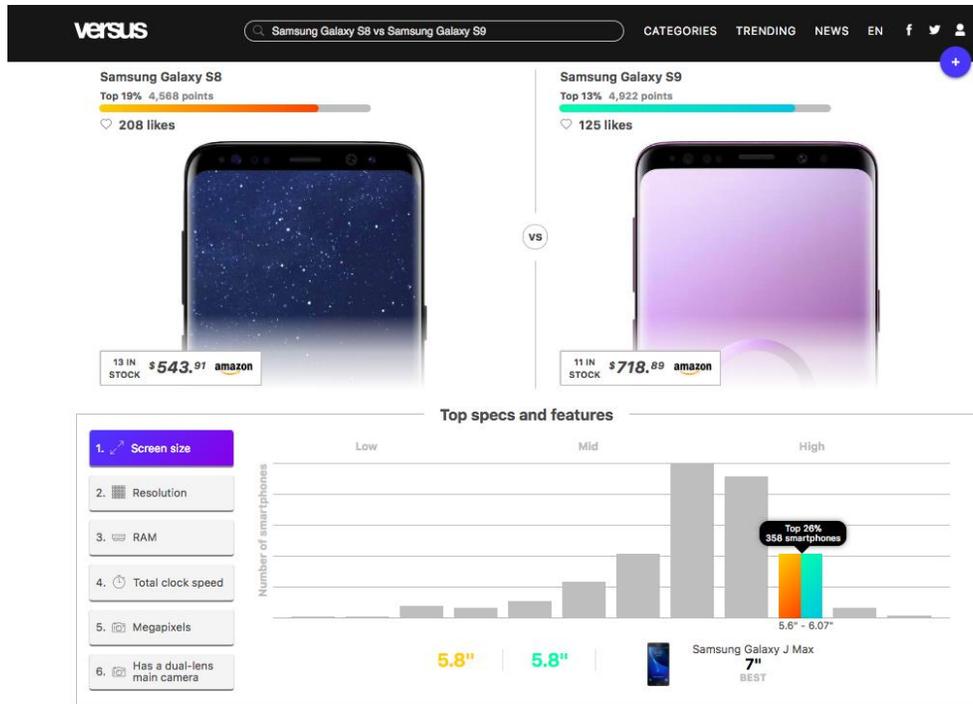


Ilustración 15: Interfaz gráfica de Versus [29]

### 2.3.4 Herramientas de calificación de productos

#### PedidosYa [30]

Es una compañía Uruguaya dedicada a pedidos de comida online en América Latina la cual brinda una plataforma sin costo adicional que permite a los usuarios elegir diferentes platos dentro de distintas opciones de restaurants. Cada usuario tiene la posibilidad de calificar y comentar cada pedido realizado los cuales sirven como referencia a los otros usuarios.

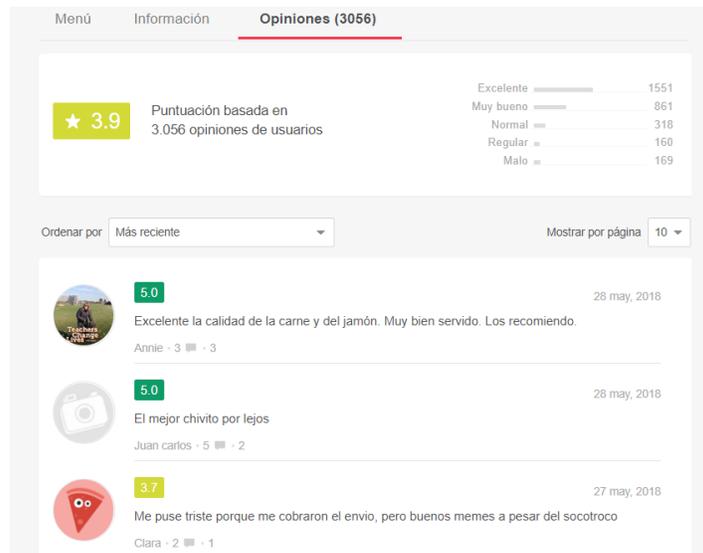


Ilustración 16: Interfaz gráfica de PedidosYa [30]

## Uber [31]

Uber es una empresa de tecnología que ofrece un servicio de transporte mediante una aplicación móvil. Existen dos tipos de perfiles de usuarios dentro de la aplicación: conductores y pasajeros. Al finalizar cada viaje el pasajero tiene la posibilidad de calificar al conductor y realizar un comentario acerca del viaje con el fin de evaluar a los conductores.

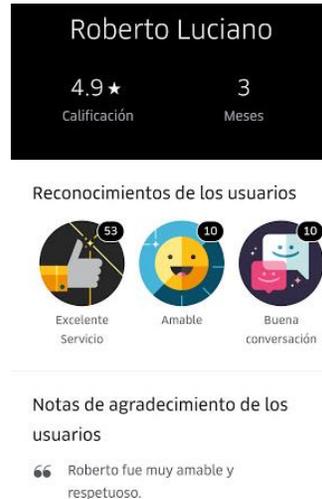


Ilustración 17: Interfaz gráfica de Uber [31]

## 2.4 Tecnologías investigadas

### 2.4.1 Web Scraping

El Web Scraping es una técnica utilizada para extraer información de páginas web de forma automatizada. Se utiliza mediante programas de software los cuales generalmente simulan la navegación de un humano en la web con el fin de recolectar información de diferentes sitios web utilizando el protocolo HTTP.

El Web Scraping se enfoca en la transformación de datos sin estructura en la web (como el formato HTML) en datos estructurados que pueden ser almacenados en alguna fuente de almacenamiento como puede ser una base de datos.

La comparación de precios en tiendas, la monitorización de datos relacionados con el clima, la detección de cambios en sitios webs son algunos de los ejemplos de la utilización de esta técnica.

Web Scraping es un campo en continuo desarrollo, compartiendo un objetivo común con la visión de la web semántica. Esta visión es bastante ambiciosa y todavía requiere de varios avances en el procesamiento de texto, el entendimiento semántico, inteligencia artificial y la interacción humano computadora. Actualmente las soluciones de Web Scraping varían entre procesos que requieren esfuerzo humano hasta sistemas totalmente automatizados que pueden convertir páginas web enteras en información estructurada. [32]

Otra técnica similar, que es a veces confundida con esta, es la técnica de Web Crawling. La diferencia con esta técnica, es que el Web Scraping extrae información específica de determinadas páginas de

interés, mientras que el Web Crawling parte de una lista de URLs y las recorre para seguir obteniendo links a otros sitios web. Los buscadores como Google utilizan esta técnica para indexar las páginas web. En la ilustración 18 se muestra un típico uso de cada técnica para apreciar mejor las diferencias.

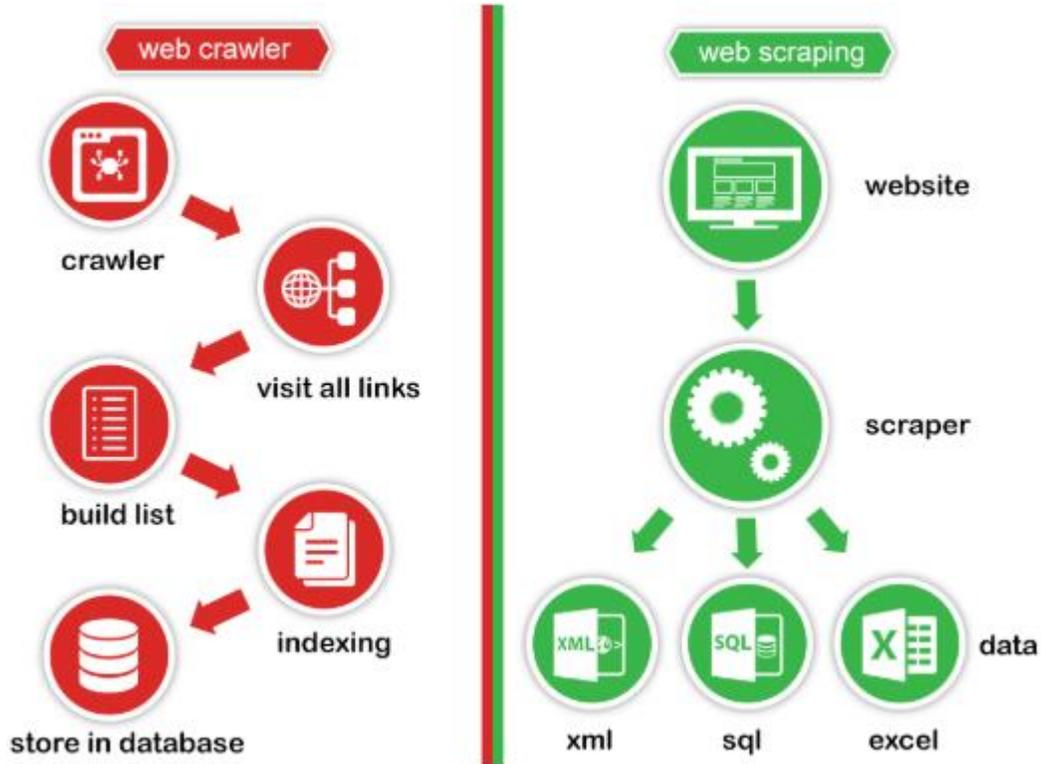


Ilustración 18: Diferencias entre Web Crawling y Web Scraping[33]

### 2.4.1.1 Buenas prácticas

El web scraping, al simular el comportamiento humano frente a sitios web, debe ser llevada a cabo cuidadosamente. A continuación se listan algunas buenas prácticas a tener en cuenta al utilizarla:

- Respetar los deseos del dueño de la página a scrapear: los sitios web tienen varias maneras de indicar limitaciones en cuanto al uso de sus datos y el scraping de los mismos:
  - robots.txt[34]: este archivo se encuentra en la raíz de los sitios web y es utilizado para indicar qué recursos pueden ser accedidos e incluso pueden indicar qué robots son los autorizados a hacerlo. Este es un estándar de exclusión de robots que nace en 1994 debido al mal comportamiento de algunos scrapers o crawlers que terminaban generando una denegación de servicio para los sitios web. Este archivo tiene un formato estandarizado que indica el agente y los recursos permitidos o denegados para ese agente. Es importante recalcar que este archivo es público y puede no ser respetado por robots maliciosos, por lo que no debe utilizarse para ocultar información. A continuación se muestra como ejemplo un fragmento del robots.txt de google.

```
User-agent: *
Disallow: /search
Allow: /search/about
Allow: /search/static
Allow: /search/howsearchworks
Disallow: /sdch
Disallow: /groups
Disallow: /index.html?
```

*Ilustración 19: Extracto del archivo robots.txt de Google*

Al utilizar en el campo agente el carácter especial \*, se indica que las reglas se aplican a todos los agentes. Al momento de scrapear un sitio, se debería verificar previamente este archivo para ver qué recursos pueden ser accedidos por el agente a utilizar.

- términos y condiciones [35]: antes de proceder a obtener datos de forma automática de un sitio web, hay que leer los términos y condiciones que imponen (en caso que existan). Verificar que el uso que se le va a dar a los datos está permitido y que los mismos no son ofrecidos mediante APIs o estén a la venta.
- Evitar sobrecargar el servidor con peticiones [36]: para evitar ataques de denegación de servicio o simplemente la degradación del mismo, es importante programar el scraper para que no realice grandes cantidades de peticiones concurrentes ni muy cercanas en el tiempo. Otra forma de evitar estos problemas es programar para que la extracción de los datos se realice en horas menos populares.

#### 2.4.1.2 Web Scraping en la web semántica

El concepto de web semántica es una extensión del World Wide Web mediante estándares definidos por World Wide Web Consortium (W3C). Estos estándares promueven formatos de datos comunes y protocolos de intercambio en la web, fundamentalmente el framework de descripción de recursos (RDF). [13]

Este concepto se utiliza en una técnica específica para realizar Web Scraping, llamada Web Scraping semántica. Esto se basa en el caso en el que las páginas que se recorren para extraer información tengan metadatos, o anotaciones específicas, como las que aparecen en RDF. En ese caso se puede expandir la técnica que realiza un parseo de los elementos HTML, sustituyendo estos últimos por las anotaciones específicas de RDF. Estas anotaciones son independientes de los estilos y hacen referencia a los datos en sí mismos por lo que ayudan mucho en caso de las ediciones de las páginas. En otros casos se utilizan las anotaciones organizadas en capas semánticas que son guardadas y manejadas en forma separada de las páginas web, de esta forma se obtienen instrucciones y la estructura de los datos antes de comenzar a extraer información de la página en sí misma. [37]

#### 2.4.2 Java edición empresarial

Cuando se hace referencia a la tecnología Java, se incluyen dos conceptos: lenguaje de programación y plataforma. Esta tecnología fue desarrollada inicialmente por James Gosling en Sun Microsystems (actualmente Oracle Corporation). [38]

El lenguaje de programación Java es un lenguaje de alto nivel orientado a objetos, cuenta con una sintaxis y un estilo particular. Por otro lado Java cuenta con varias plataformas diferentes, siendo una plataforma un ambiente particular en el que corren las aplicaciones Java.

Existen cuatro plataformas para el lenguaje de programación Java, estas son:

- Java edición estándar (Java SE)
- Java edición empresarial (Java EE)
- Java edición micro (Java ME)
- JavaFX

Todas estas plataformas consisten de una máquina virtual Java y una interfaz de programación de aplicaciones (API por su sigla en inglés). La máquina virtual es un programa que corre las aplicaciones Java, mientras que la API es un conjunto de componentes de software que se utiliza para la creación de otros componentes o aplicaciones.

La plataforma más comúnmente usada y en la que la mayoría de los usuarios piensa cuando se habla de Java es la estándar. La API de esta plataforma provee el núcleo de todas las funcionalidades del lenguaje de programación Java.

La plataforma Java empresarial está construida arriba de la estándar, y se creó para desarrollar y correr aplicaciones de gran escala, con muchas capas, de forma escalable y confiable. Este tipo de aplicaciones son las llamadas “aplicaciones empresariales”, ya que son diseñadas para resolver problemas encontrados en grandes empresas. Esto no quiere decir que este tipo de aplicaciones sea útil solo para las grandes corporaciones, sino que en este mundo cada vez más conectado, este tipo de aplicaciones provee beneficios que son útiles o incluso esenciales para todo tipo de desarrolladores. Al buscar construir aplicaciones confiables y escalables se encuentran varias complejidades, es por este motivo que nace Java EE, para brindar ciertas facilidades mediante un modelo de desarrollo, una API y un ambiente que permite a los desarrolladores concentrarse más en las funcionalidades mismas de su aplicación.

La arquitectura para la que está diseñado Java EE es una arquitectura en capas. Esto permite que las funcionalidades de la aplicación estén separadas en áreas independientes. Típicamente las aplicaciones con esta arquitectura cuentan con una capa cliente, una capa de datos y una capa en el medio de ambas, llamada capa de negocio, Java EE se centra en esta última. A continuación se describen las mismas [39]:

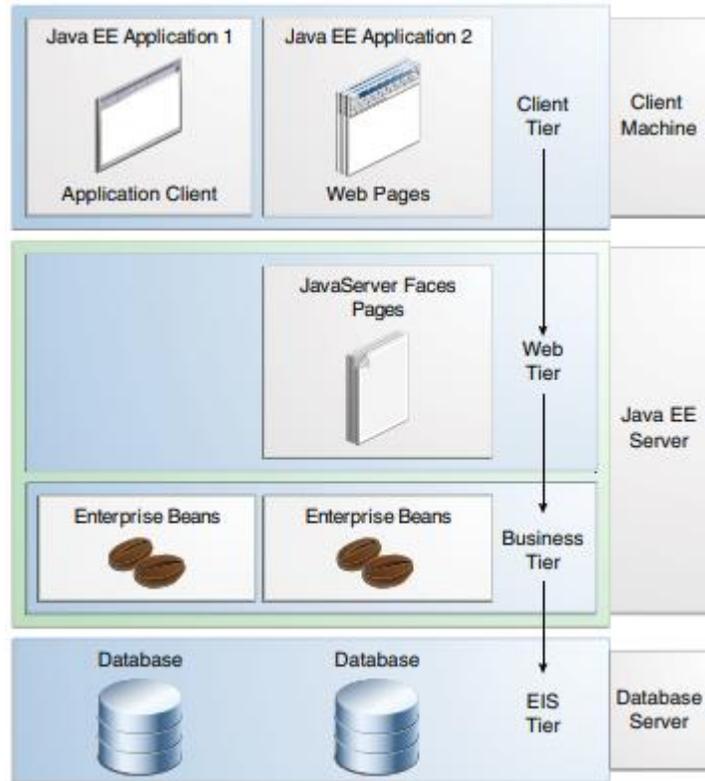


Ilustración 20: Arquitectura típica de una aplicación empresarial en la plataforma Java EE [40]

- Capa cliente: se encarga de consumir o acceder al servidor Java EE. Típicamente se encuentra en una máquina diferente a la del servidor. La capa cliente realiza consultas que luego son procesadas por la capa del medio y contestadas nuevamente a la capa cliente. Diferentes tipos de aplicaciones pueden ser clientes de Java EE, ni siquiera tienen por qué ser aplicaciones Java.
- Capa de datos: se compone de servidores de base de datos u otro tipo de aplicaciones que nutren a la aplicación Java EE con datos. Típicamente también se encuentra separada del servidor. Para facilitar la comunicación con esta capa, Java EE cuenta con tecnologías específicas, algunas de estas son:
  - La API de conectividad de la base de datos (JDBC): se utiliza para acceder y recolectar datos de la capa de datos. Un uso común de la misma es la realización de consultas SQL a una base de datos particular.
  - La API de persistencia (JPA): es de más alto nivel que JDBC ya que además de poder acceder y recolectar datos también mapea los mismos a objetos de Java brindando muchas facilidades para poder consultarlos o guardarlos, ocultando la complejidad de JDBC.
- Capa de negocio: esta capa es la encargada de procesar toda la información y es la implementada con Java EE. Dentro de esta plataforma se definen también otras sub capas para lograr una solución más modular:
  - La capa web: es la encargada de la comunicación con la capa cliente. También cuenta con tecnologías específicas de Java EE, por ejemplo: Servlets (clases Java que dinámicamente procesan las solicitudes del cliente y construyen respuestas) y JavaServer Faces (componente de interfaz de usuario que permite incluir, validar y guardar componentes específicos de interfaz gráfica).
  - La capa de negocio: es la encargada de la lógica central de la aplicación. En una aplicación empresarial bien diseñada la funcionalidad central se encuentra en componentes de esta capa. Algunas de las tecnologías que brinda Java EE para esta capa son las siguientes:
    - JavaBeans empresariales (EJB): son componentes manejables que encapsulan las funcionalidades.
    - JAX-RS RESTful web services: una API para crear servicios web que responden a métodos HTTP.

Los servidores Java EE son aplicaciones de servidor que implementan la plataforma Java EE. A veces son llamados servidores de aplicación porque permiten servir datos de aplicaciones a los clientes, como lo servidores web ofrecen páginas web a los navegadores. Estos servidores ofrecen servicios en forma de contenedor para cada tipo diferente de componente presente en la aplicación empresarial.

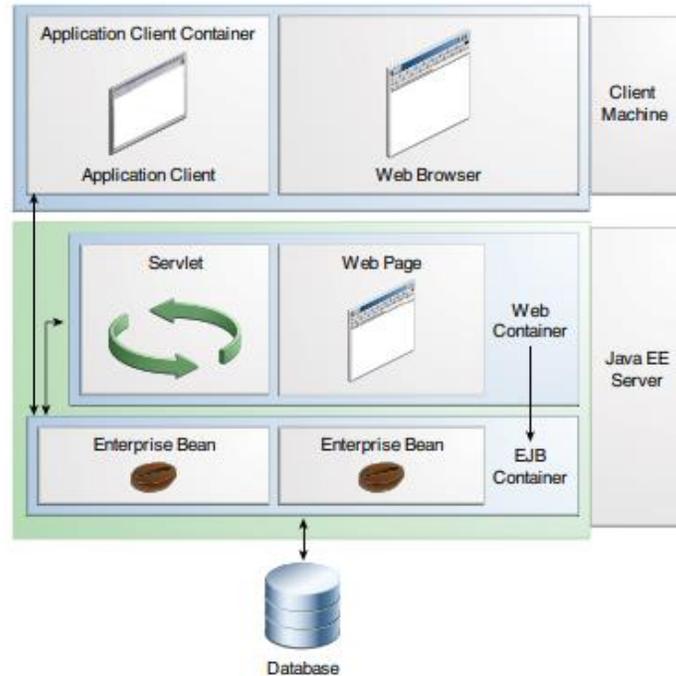


Ilustración 21: Servidor Java EE y contenedores [40]

Algunos de los servicios ofrecidos por los contenedores son [40]:

- El modelo de transacción: permite especificar relaciones entre métodos que conforman una única transacción para que todos esos métodos sean tratados como una única unidad.
- Interfaz de directorio y nombrado (JNDI): ofrece una API de búsqueda, y provee una interfaz unificada para múltiples servicios para que los componentes puedan acceder a estos.

Además dentro del servidor Java EE se puede encontrar dos tipos de contenedores diferentes:

- Contenedor EJB: encargado de ejecutar los EJB para las aplicaciones Java EE.
- Contenedor Web: maneja la ejecución de páginas web, servlets y algunos componentes EJB para aplicaciones Java EE.



### 3. Análisis

En este capítulo en primer lugar se detallan las características de los usuarios del sistema y luego se describe el análisis de los requerimientos. Este proceso es realizado con el fin de comprender las funcionalidades que debe proveer el sistema, de forma que permita cumplir con los objetivos planteados. Es fundamental que en esta etapa, se comprenda completamente la naturaleza del producto a construir mediante requisitos funcionales y no funcionales. En este capítulo, no se explicará en detalle cada requerimiento (la información completa se encuentra en el documento de requisitos en el apéndice 1 de este informe), sino que se agruparán en categorías y se describirán las características principales de cada una.

Al finalizar este análisis se presenta un diagrama de casos de uso críticos en donde se resume quién puede utilizar el sistema y qué puede hacer cada actor; no se muestran los casos de uso en detalle, solamente se resumen algunas de las relaciones entre los casos de uso, los actores y el sistema. Luego, siguiendo el análisis se presenta una parte del modelo de dominio del sistema y se explican las entidades más relevantes e interesantes del mismo. Para finalizar este capítulo se presentan algunas características de la arquitectura del sistema, su organización y principales componentes, una primera aproximación a lo que es la solución concreta que se describe en el capítulo 4.

#### 3.1 Usuarios del sistema

Este sistema está destinado a la comunidad de desarrolladores y usuarios TIG en general. Las razones que llevan a una persona a utilizar el observatorio pueden ser variadas, desde usuarios no técnicos que buscan una TIG completa, hasta desarrolladores que buscan una TIG sencilla en forma de librería para incluir en su proyecto. Pero todos los usuarios tienen en común el hecho de que buscan un lugar que les muestre las opciones que tienen disponibles y cuáles son las que más se adecúan a lo que están buscando.

Los potenciales usuarios de este sistema son:

- Desarrolladores de SIG: estos usuarios invierten mucho tiempo de su trabajo en la investigación de los productos que se encuentran disponibles para desarrollar un nuevo SIG. Con el observatorio se pretende disminuir el tiempo dedicado a esta investigación, centralizando la información que se considera relevante en el momento de la elección de un producto. Por ejemplo, para los desarrolladores es de gran interés saber si una tecnología es paga o gratuita, si implementan determinados estándares o no. Por esta razón estos datos estarán disponibles en el sistema para su consulta. Otra facilidad que se le brinda a estos usuarios es un foro, mediante el cual pueden evacuar dudas específicas de alguna de los productos o compartir su conocimiento con el resto de los usuarios.

- Creadores de TIG: a estos usuarios les interesa comunicar acerca de las actualizaciones y cambios en las funcionalidades de sus productos, así como, informar de sus nuevos productos. Les interesa que su producto aparezca en el observatorio para que a la hora de tomar una decisión, los usuarios lo tengan en cuenta; y además que la información que se incluya sobre el mismo sea actualizada y completa. Su participación es importante a su vez en el foro, ya que podría resultar de ayuda para la comunidad compartiendo información o ayudando a resolver las dudas que puedan surgir sobre su/s producto/s.
- Empresas con interés en la construcción de SIG: estas empresas se interesan en respetar estándares para facilitar la comunicación de los sistemas, por lo tanto les es útil saber qué producto implementa un determinado estándar, esta información y otras cuestiones de su interés pueden encontrarla en el observatorio.
- Investigadores en el área geoespacial: puede resultarles de interés ver gráficas del uso de determinados productos, así como qué estándares son los más implementados, entre otras conclusiones que deben poder sacarse del sistema mediante reportes.

### 3.2 Relevamiento de requerimientos

Los requerimientos del sistema se pueden dividir en diferentes áreas:

- Gestión de Usuarios
- Gestión de Productos
- Gestión de Estándares
- Actualización de Datos
- Foro

A continuación se detallan cada una de ellas.

#### 3.2.1 Gestión de usuarios

Existen diferentes tipos de usuarios los cuales van a tener distintas funcionalidades disponibles en el sistema:

- Usuarios anónimos: son usuarios que no están registrados en el sistema, los cuales tienen acceso a la visualización de los productos (con la posibilidad de aplicar diferentes filtros) y de los estándares. También pueden realizar comparaciones entre los diferentes productos y visualizar el foro (no participar del mismo). Podrán ver los reportes que genera el sistema pero no se tendrán en cuenta para la generación de los mismos, por ejemplo, en el reporte que muestra los productos más visualizados sus visitas no se contabilizan, ya que resulta interesante registrar la actividad de usuarios registrados de los que se tiene cierta información y no de usuarios de los que no se conoce nada.

- Usuarios colaboradores: usuarios registrados en el sistema, el ingreso lo pueden hacer mediante las credenciales provistas al momento de registrarse o mediante su cuenta de Google. Los colaboradores pueden hacer todo lo que hacen los usuarios anónimos pero además tienen otras funcionalidades disponibles, como ser, sugerir nuevos productos, comentar y calificar los productos del sistema. Estos usuarios tienen un perfil dentro del sistema, dicho perfil cuenta con información de los mismos, como por ejemplo, su nombre, una imagen y su país de origen; se le permite al usuario editar los datos de su perfil cuando lo desee. Además dentro de este perfil se cuenta con el concepto de “conocimiento” sobre productos presentes en el sistema que indica cierto nivel de manejo de los mismos. Los usuarios podrán agregar o eliminar estos conocimientos de su perfil, y a su vez podrán validar los conocimientos de otros usuarios. Los colaboradores podrán participar del foro creando nuevos temas de discusión o contestando hilos ya existentes. Estos usuarios se tienen en cuenta al momento de generar los reportes.
- Usuarios administradores: al igual que los colaboradores son usuarios que están registrados en el sistema, un administrador puede dar de alta nuevos administradores. Estos usuarios son los encargados de dar de alta nuevos productos (de cero o aceptando sugerencias de los colaboradores), además pueden modificar productos ya existentes o eliminarlos. En cuanto a los estándares pueden dar de alta nuevos y editar o eliminar los que ya se encuentran en el sistema. Cumplen el rol de mediadores en el foro pudiendo borrar comentarios o hilos que les parecen inapropiados o que no corresponden. Los administradores recibirán notificaciones vía mail cuando los colaboradores ingresen una nueva sugerencia de un producto nuevo.

El siguiente diagrama ilustra los casos de uso que puede realizar cada tipo de usuario:

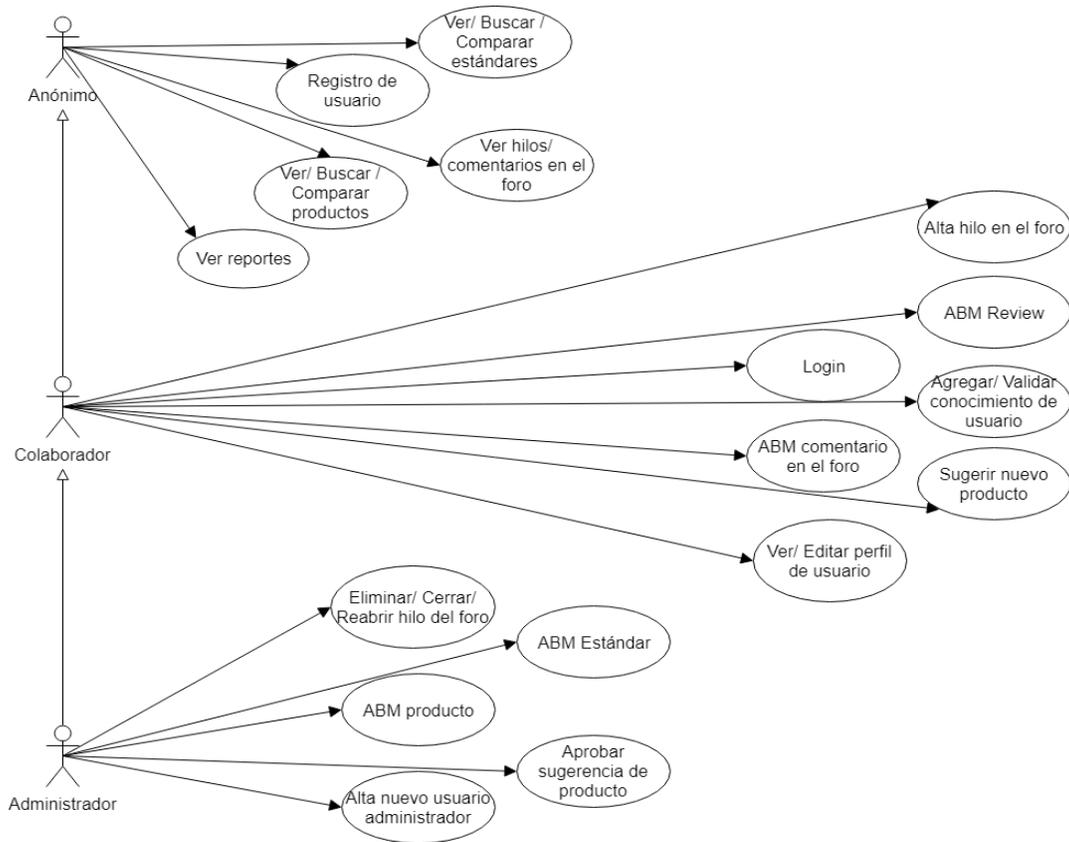


Ilustración 24: Diagrama de casos de uso<sup>2</sup>

### 3.2.2 Gestión de productos

El sistema cuenta con productos que se corresponden a las TIG, en el observatorio se le debe mostrar a los usuarios una lista con todos los productos cargados, asegurando la completitud y fiabilidad de la información que se brinda sobre los mismos. Es por esta razón que los datos deben pasar por el control de un usuario administrador para poder quedar disponibles en el sistema. El flujo para ingresar un nuevo producto al observatorio es el siguiente: una vez que se ingresa una nueva sugerencia de un producto, se notifica a los administradores, los cuales evalúan los datos ingresados y tras corroborar su correctitud, realizan el alta del producto efectiva en el sistema.

<sup>2</sup> Las funcionalidades disponibles para el administrador incluyen las del colaborador y el anónimo y las funcionalidades disponibles para el colaborador incluyen las del anónimo.

Para que la búsqueda de productos sea más sencilla para los usuarios, en el listado de productos se debe poder aplicar distintos filtros de modo que la lista se reduzca a los productos que cumplan con lo pedido. Los filtros que se deben permitir son:

- Nombre de producto, se debe permitir introducir parte o todo el nombre del producto.
- Tipo de producto, es decir, a qué capa, de la arquitectura de referencia explicada en la sección 2.2, pertenece. Se deben listar las opciones para que el usuario seleccione una.
- Costo, se debe permitir seleccionar si se busca productos gratuitos o que tengan un costo.
- Soporte, el usuario debe poder seleccionar si desea que los productos que se muestran tengan soporte o no.
- Estándares implementados por los productos, se debe mostrar una lista de los estándares disponibles y se debe permitir al usuario seleccionar uno o varios estándares.
- Open source o no.

Se le debe permitir al usuario acceder a una ficha del producto con toda la información disponible y se le dará la opción de comentar y calificar el producto según su experiencia. A su vez, podrá ver las calificaciones y comentarios de otros usuarios lo que sirve como guía al momento de considerar o no el uso de determinado producto.

Otra funcionalidad que debe proveer el sistema, en cuanto a los productos, es la comparación de los mismos, se deben poder elegir dos productos que se desean comparar y los resultados de la comparación deben mostrar de forma clara para el usuario las diferencias y similitudes que tienen. Además se deberá mostrar la calificación (dentro del ranking del sistema) de cada uno de los productos que se están comparando.

Otra herramienta que debe brindar el sistema es un espacio en donde los usuarios puedan contar sus inquietudes, dar aportes o compartir sus vivencias en cuanto a los productos y estándares del sistema. Esta es una parte fundamental, ya que permite la interacción entre la comunidad SIG y la colaboración entre los usuarios.

### 3.2.3 Gestión de estándares

Los estándares con los que debe contar el sistema son estándares geoespaciales y al igual que con los productos, los encargados de realizar la gestión de los mismos son los usuarios administradores. En el sistema se deben poder visualizar todos los estándares con sus respectivos detalles para que los usuarios tengan la información más relevante de forma sencilla y amigable. En cuanto al ingreso de los estándares al sistema se puede realizar mediante el componente de Web Scraping el cual debe contemplar el caso de nuevas versiones de un estándar. En el caso de que el componente de Web Scraping no encuentre toda la información necesaria para dar de alta un nuevo estándar, lo ingresará como un “estándar pendiente” y luego, un usuario administrador podrá completar la información y aceptar el estándar definitivamente en el sistema. Además, los administradores podrán agregar nuevos estándares al sistema de forma manual, utilizando un formulario provisto por la interfaz web.

#### 3.2.4 Actualización de datos

La recolección de datos es una de las principales complejidades del sistema dado que la información no está centralizada.

Los datos van a tener una carga inicial por parte de las desarrolladoras y luego contará con varias alternativas para su actualización:

- Sistema externo encargado de verificar si hay nuevas liberaciones de productos o estándares en la web mediante Web Scraping.
- Sistema externo que busca nuevas versiones de productos mediante sus cuentas oficiales de Twitter.
- Plataforma colaborativa que le permite a los colaboradores del sistema sugerir cambios, ya sea en estándares o en productos, que luego son aprobados por los administradores.

#### 3.2.5 Foro

El foro es el componente del sistema que facilita el intercambio de información y la evacuación de las dudas en la comunidad SIG. Para poder participar en el mismo se necesita estar logueado en el sistema, y los administradores deben ser quienes tengan la función de moderadores del mismo, pudiendo eliminar comentarios inapropiados o cerrar hilos que hayan sido resueltos o no correspondan con la temática. Por lo tanto las funcionalidades básicas permitidas en el foro son:

- Crear nuevos hilos de conversación: todo usuario logueado podrá crear un nuevo hilo de discusión.
- Agregar comentario a un hilo: todo usuario logueado podrá agregar comentarios a cualquier hilo que se encuentre abierto.
- Editar / Eliminar comentario: el usuario que realizó el comentario podrá editar o eliminar el comentario que realizó en un hilo. Además el administrador podría borrar comentarios que se consideren inapropiados.
- Abrir / Cerrar / Eliminar hilo: los administradores del sistema podrán cerrar los hilos de discusión inhabilitando el agregado de comentarios al mismo pero sin eliminarlo. Además podrá reabrirlo si lo desea, o incluso borrarlo y que el resto de los usuarios ya no puedan tener acceso al mismo.

### 3.3 Casos de uso críticos

A partir del relevamiento de requerimientos surgieron diversos casos de uso. Los casos de uso que se consideran más importantes o que son críticos se encuentran en el siguiente diagrama.

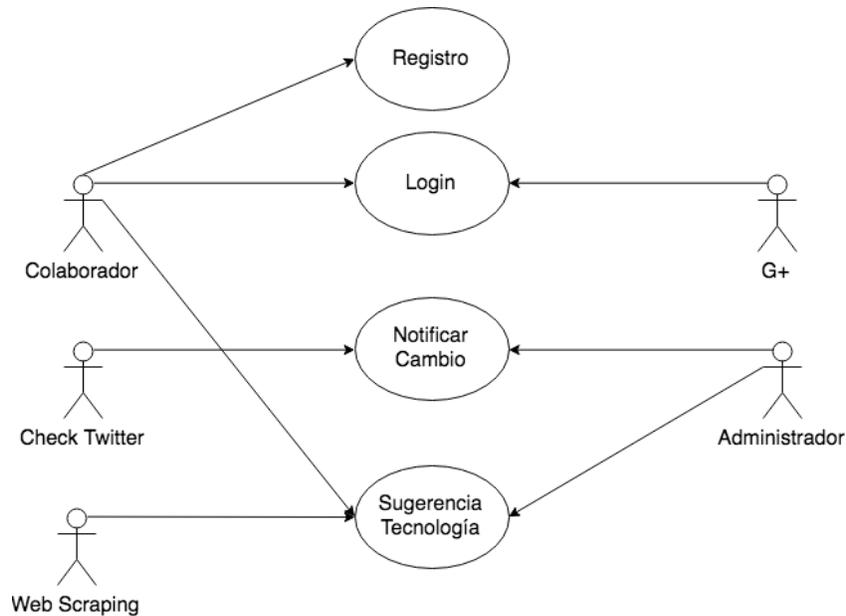


Ilustración 25: diagrama de casos de uso críticos

En este diagrama se pueden observar los diferentes actores del sistema, estos son, los usuarios administradores y colaboradores, explicados en la sección 3.1, otros actores son los componentes Check Twitter y Web Scraping, encargados de procesos automáticos para consultar distintas fuentes de dónde relevar datos, y por último se muestra el actor G+ (Google+) que permite que los usuarios accedan al sistema utilizando su cuenta google que permite que se conviertan en usuarios colaboradores una vez que ingresan por este medio.

Los casos de usos presentados en este diagrama se consideran críticos, ya que implican la mayor parte de los riesgos técnicos que puedan surgir durante la etapa de desarrollo.

### 3.4 Entidades principales

La ilustración 26 muestra una sección del modelo de dominio, en donde se visualizan las entidades que se consideran principales para el sistema.

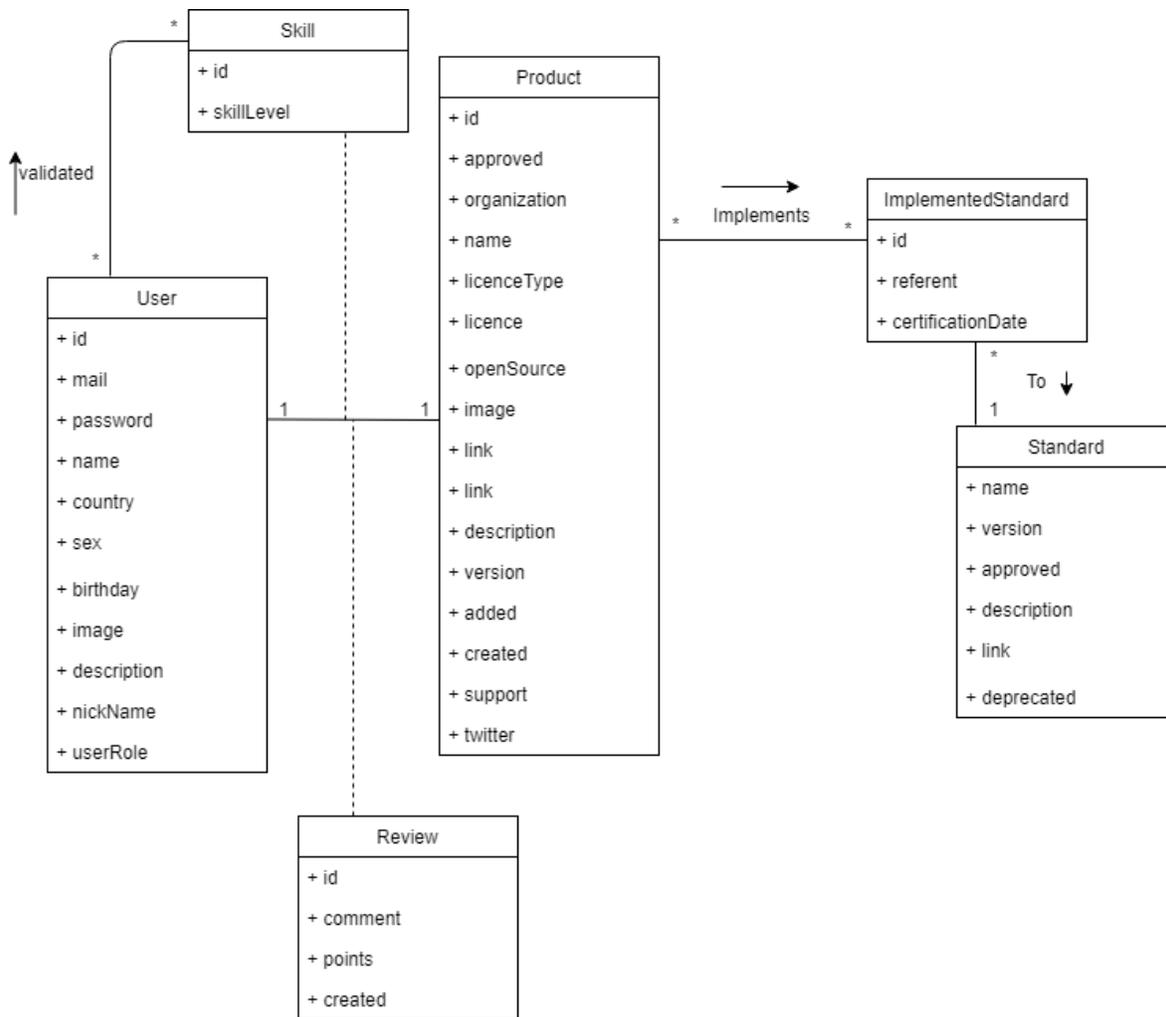


Ilustración 26: extracto de las entidades principales del modelo de dominio de la solución

La entidad “User” es la encargada de mantener los datos de los usuarios del sistema, entre los que se destaca el “userRol” que permite diferenciar entre usuarios colaboradores o administradores. Se consideró pertinente incluir la información del país de origen de los usuarios, ya que, en la sección de reportes, se generará un reporte que muestra la cantidad de usuarios que utilizan el sistema en cada país. Por otro lado, con el fin de preservar la identidad de los usuarios, cabe mencionar que la contraseña que se va a persistir tiene un mecanismo de hashing que se explicará más adelante en este informe pero cuyo objetivo es no dejar vulnerable la información sensible (la clave de ingreso al sistema, en este caso).

Otra entidad interesante es la de “Product” donde se maneja toda la información de los productos del sistema, entre los datos que se manejan se resaltan: la versión del producto, tipo de licencia (comercial, gratuita o sin información, si no se tiene información sobre la licencia del producto), si tiene soporte o no y para diferenciar aquellos productos que son sugerencias de los que ya son efectivos en el sistema se utilizará la propiedad “approved”.

Luego se destaca la entidad “Standard” la cual se corresponde a los estándares del sistema, esta entidad se encuentra identificada por su nombre y versión, ya que, pueden existir distintas versiones de un mismo estándar en el sistema. Al igual que para los productos, se utiliza el atributo “approved” para diferenciar los estándares que todavía no se aprobaron (los que se ingresaron mediante alguno de los mecanismos de actualización de datos, como se explicó anteriormente en este capítulo) de los que ya se aprobaron.

Una relación interesante a explicar es la que se genera entre “Product” y “Standard”. Como se plantea en la arquitectura de referencia utilizada para este proyecto (sección 2.2), los productos en el área de los SIG se pueden categorizar por el nivel de cumplimiento de un determinado estándar. Esta categorización se explicó en detalle en la sección 2.2.1. Para reflejar esta situación de la realidad se utiliza la entidad “ImplementedStandard”, ya que permite, en el caso de que un producto cumpla algún estándar, tener la información acerca de la conformidad de ese cumplimiento.

Por otro lado, se destaca la entidad “Review” que permite que un usuario realice un comentario (y solo uno) acerca de un determinado producto, además de comentar puede calificar el producto con una puntuación de 1 a 5.

Por último, la entidad “Skill” permite plantear la realidad acerca de los conocimientos de los usuarios sobre los productos. Dado un usuario y un producto se debe poder decir si el usuario tiene conocimiento del producto y qué nivel tiene. A su vez, otros usuarios del sistema pueden validar que efectivamente un usuario tiene el nivel de conocimiento que indicó para cierto producto.

## 4. Solución planteada

En esta sección se muestran los aspectos más relevantes de la solución propuesta. En primer lugar, se detalla la arquitectura del sistema mostrando cómo interactúan los componentes que forman parte del mismo. Luego se muestra el diagrama de solución con las diferentes tecnologías utilizadas y una explicación del funcionamiento de cada uno de los componentes del sistema. A continuación, se mencionan algunos aspectos destacados con respecto a la implementación del sistema y por último se describen las decisiones más relevantes que se fueron tomando a lo largo del proyecto.

### 4.1 Arquitectura de la solución

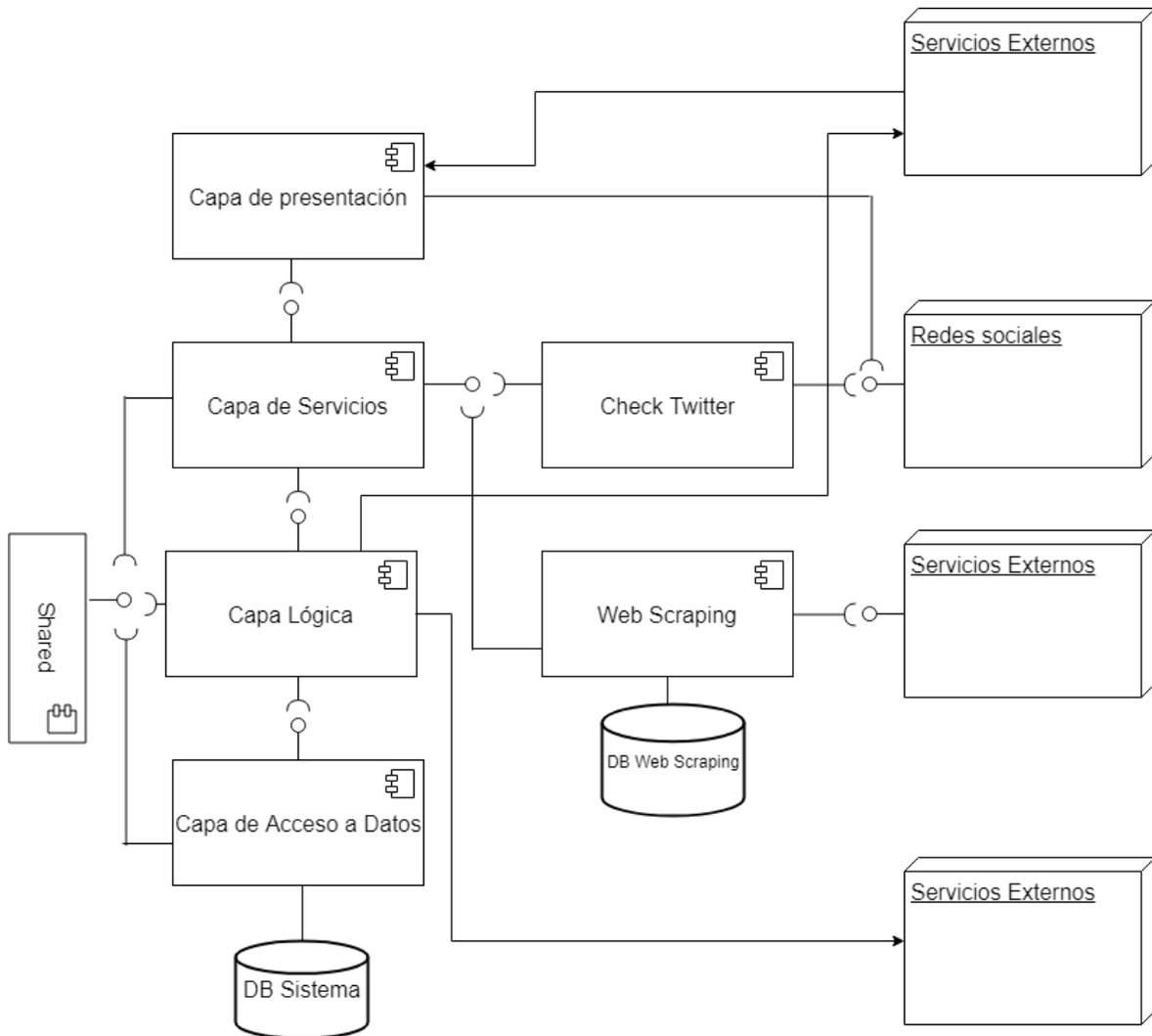


Ilustración 27: Diagrama de arquitectura de la solución

Para la solución se utiliza una arquitectura en capas. La primera capa es la de presentación que es la encargada de desplegar la información a los usuarios. Es el componente consumidor o cliente del sistema ya que se nutre de los servicios ofrecidos por la capa de servicios. Este componente es un cliente liviano para no afectar la usabilidad de la aplicación. El mismo además consume servicios de

redes sociales para el registro y la autenticación de usuarios en la plataforma. También se integra con un componente para poder recibir notificaciones de la capa lógica.

Las capas de servicios, lógica y de acceso a datos son las encargadas del procesamiento y la persistencia de la información. Estas utilizan un componente Shared que es el que tiene el modelo de datos común a todas las capas. Además la capa lógica se integra tanto con un servidor de email como con el componente para poder enviar notificaciones a la capa de presentación.

Toda la información del sistema se almacena en una base de datos, la capa encargada de la persistencia y el acceso a la misma es la capa de acceso a datos, siendo la única con comunicación directa a la base de datos.

Por último se encuentran los componentes de la actualización de los datos, estos se comunican con sistemas externos y luego actualizan la información del sistema consumiendo servicios de la capa de servicios. Además el componente de Web Scraping cuenta con una base de datos propia para guardar información específica de configuración.

Esta arquitectura permite tener separación entre las diferentes funcionalidades, generando más modularidad y reutilización de cada componente. Además facilita las posibles integraciones a futuro de nuevos sistemas externos mediante la capa de servicios.

## 4.2 Diagrama de solución

En el siguiente diagrama se ilustra la arquitectura del sistema con las tecnologías que se utilizaron para desarrollar cada componente y los mecanismos de comunicación que permiten la interacción entre ellos.

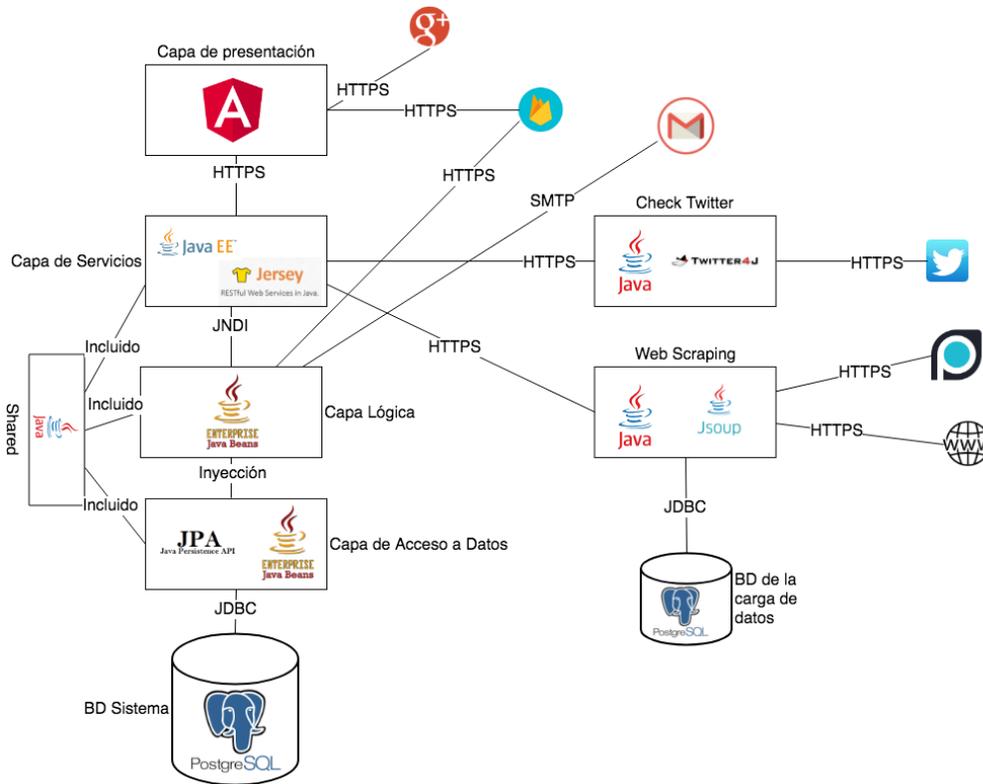


Ilustración 28: Diagrama de la solución con las diferentes tecnologías utilizadas en la misma

En la ilustración 27 se describen los diferentes componentes de la solución con sus respectivas tecnologías. A partir de esa ilustración se realiza la siguiente agrupación de los componentes presentes en la misma, en cuatro grandes partes: el front-end formado por la capa de presentación, el back-end formado por las capas de servicios, lógica, de acceso a datos y el componente shared, la base de datos del sistema y los componentes de actualización formados por Web Scraping y su base de datos y Check Twitter. A continuación se explicará en detalle cada una de estas partes y las tecnologías que se usaron para desarrollarlas.

### 4.2.1 Back-end

Este componente es el encargado de todo el procesamiento y la gestión de los datos necesarios para el observatorio.

Apoyado en este framework, el back-end, es dividido en 3 capas lógicas. Esta arquitectura se utiliza mucho para permitir mayor flexibilidad y menor impacto en los cambios que se van agregando, logrando una solución más escalable y mantenible. A continuación se describe brevemente el propósito de cada capa y algunas de las APIs de Java EE utilizadas en las mismas.

#### 4.2.1.1 *Capa de acceso a datos*

Esta parte de la aplicación es la encargada de la comunicación con la base de datos, este componente contiene tanto el modelo de datos como toda la lógica utilizada para la lectura, escritura, edición y borrado de los mismos en la base de datos. Para la implementación de la solución se partió del modelo de dominio, traduciendo las entidades en clases Java y mediante las anotaciones de la API JPA de Java EE las mismas se mapearon a tablas de una base de datos relacional. Como implementación de la API JPA se utilizó la herramienta de mapeo de objetos relacionales (ORM por su sigla en inglés) Hibernate[43]. Algunos ejemplos importantes de las anotaciones utilizadas en la solución son los siguientes:

- **Id:** el atributo señalado con esta anotación es la clave primaria de la tabla en la base de datos. Puede estar acompañada de `GeneratedValue` que indica, según el valor que se le asigne, la estrategia a utilizar por la base de datos en el caso de que el atributo sea autogenerado. Al elegir el valor `AUTO` se le delega a la implementación de JPA (en este caso Hibernate) la decisión de cómo generar los valores. Hibernate selecciona una estrategia basándose en el manejador de base de datos utilizado. En el caso de Postgres, el valor `AUTO`, determina que el valor se genera utilizando una secuencia, en el caso de no indicarle en la anotación un nombre particular el manejador de la base de datos utiliza la secuencia por defecto.
- **Temporal:** se utiliza para mapear los tipos de Java que indiquen fechas o tiempo a tipos de SQL. Los valores posibles son `DATE`, `TIME` y `TIMESTAMP`.
- **Column:** esta anotación permite configurar varios aspectos de la columna de la base de datos, por ejemplo: el nombre, el largo, si los valores son únicos, se puede configurar un tipo específico de la misma. Un caso de esto último es cuando un campo tiene una longitud muy grande, Postgres permite definir la columna con un tipo especial "Text".
- **OneToMany, OneToMany, ManyToOne, ManyToMany:** estas anotaciones indican cómo se mapean las relaciones entre las diferentes entidades a las tablas de la base de datos. Estas relaciones pueden definirse unidireccionalmente o bidireccionalmente, salvo por la relación `ManyToMany` que es obligatoriamente bidireccional. Estas anotaciones generan en caso de ser necesario una tabla específica para la relación o una columna extra. Además entre otras cosas permite configurar qué operaciones deben realizarse en cascada (el guardado, la edición, el eliminado) y las estrategias de lectura de las mismas (`LAZY` para indicar que al leer el objeto no busque también el relacionado, `EAGER` para especificar lo opuesto).

```

@Table
@Entity
public class Product implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    private String name;
    @Enumerated(EnumType.STRING)
    private LicenceType licenceType;
    private String licence;
    private Boolean openSource;
    private String image;
    private String link;
    @Column(columnDefinition = "TEXT")
    private String description;
    private String version;
    @Temporal(TemporalType.DATE)
    private Date added;
    @Temporal(TemporalType.DATE)
    private Date created;
    private Boolean support;
    private String organization;
    @ManyToOne
    private ProductType type;
    private boolean approved;
    @OneToMany(cascade = CascadeType.ALL)
    private List<ImplementedStandard> implementedStandards;
}

```

Ilustración 29: Extracto de la clase Producto con las anotaciones JPA

Para la lógica misma de la persistencia se utilizaron EJBs stateless ofrecidos a través de interfaces locales. Como la lógica utilizada para las 4 operaciones básicas de la persistencia (alta, lectura, baja y modificación) es análoga, se utiliza una clase abstracta para la centralización de la misma. Esa clase en su constructor recibe una entidad genérica que es la que utiliza en sus operaciones. Por cada entidad del sistema hay un EJB encargado de su persistencia, estos EJBs extienden la clase abstracta genérica para heredar su lógica y evitar de esa forma repetir código.

```

public abstract class AbstractFacade<T> {

    private Class<T> entityClass;

    public AbstractFacade(Class<T> entityClass) {
        this.entityClass = entityClass;
    }

    protected abstract EntityManager getEntityManager();

    public T create(T entity) {
        getEntityManager().persist(entity);
        getEntityManager().flush();
        return entity;
    }

    public void edit(T entity) {
        getEntityManager().merge(entity);
    }

    public void remove(T entity) {
        getEntityManager().remove(getEntityManager().merge(entity));
    }

    public T find(Object id) {
        T response = getEntityManager().find(entityClass, id);
        if(response == null){
            throw new EntityNotFoundException("Object id = " + id.toString());
        } else {
            return response;
        }
    }

    public List<T> findAll() {
        javax.persistence.criteria.CriteriaQuery cq = getEntityManager().getCriteriaBuilder().createQuery();
        cq.select(cq.from(entityClass));
        return getEntityManager().createQuery(cq).getResultList();
    }
}

```

Ilustración 30: Extracto de la clase abstracta que implementa las operaciones de persistencia

```

@Local
public interface ThreadFacadeLocal {

    Thread create(Thread thread);

    void edit(Thread thread);

    void remove(Thread thread);

    Thread find(Object id);

    List<Thread> findAll();

    List<Thread> findRange(int[] range);

    int count();

}

```

Ilustración 31: Interfaz local de la clase encargada de la persistencia de los hilos del foro del sistema

```

@Stateless
public class ThreadFacade extends AbstractFacade<Thread> implements ThreadFacadeLocal {

    @PersistenceContext(unitName = "ProyectoPersistenceUnit")
    private EntityManager em;

    @Override
    protected EntityManager getEntityManager() {
        return em;
    }

    public ThreadFacade() {
        super(Thread.class);
    }
}

```

*Ilustración 32:* Clase que implementa la persistencia de los hilos del foro del sistema

Para lograr la comunicación con la base de datos, en el servidor donde va a correr la aplicación hay que configurar un data source. Esto es lo que contiene toda la información necesaria para la comunicación y debe tener un nombre único en el servidor, ya que va a ser utilizado para la configuración de JPA.

Los EJB para poder comunicarse con la base de datos utilizan un EntityManager, que es la interfaz usada para interactuar con un PersistenceContext. Un PersistenceContext es un conjunto de entidades a persistir, en este caso, son las clases que definieron el modelo. Además de indicar qué clases la conforman, en el PersistenceContext se especifica el data source utilizado. El EntityManager es parte de JPA y ofrece operaciones genéricas para la persistencia de los objetos.

```

<persistence-unit name="ProyectoPersistenceUnit" transaction-type="JTA">
  <jta-data-source>java:/ObservatorioRemoto</jta-data-source>
  <exclude-unlisted-classes>>false</exclude-unlisted-classes>
  <properties>
  </properties>
</persistence-unit>

```

*Ilustración 33:* Archivo de configuración de la unidad de persistencia

JDBC datasource 'ObservatorioRemoto' (enabled)

JDBC datasource configurations.

Disable

Attributes Connection Pool Security Properties Validation Timeouts Statements

Need Help?

Edit

Name:	ObservatorioRemoto
JNDI:	java:/ObservatorioRemoto
Is enabled?:	true
Statistics enabled?:	false
Driver:	postgresql-42.1.3.jar

Ilustración 34: Configuración del data source en el servidor Wildfly

4.2.1.2 Capa de negocio

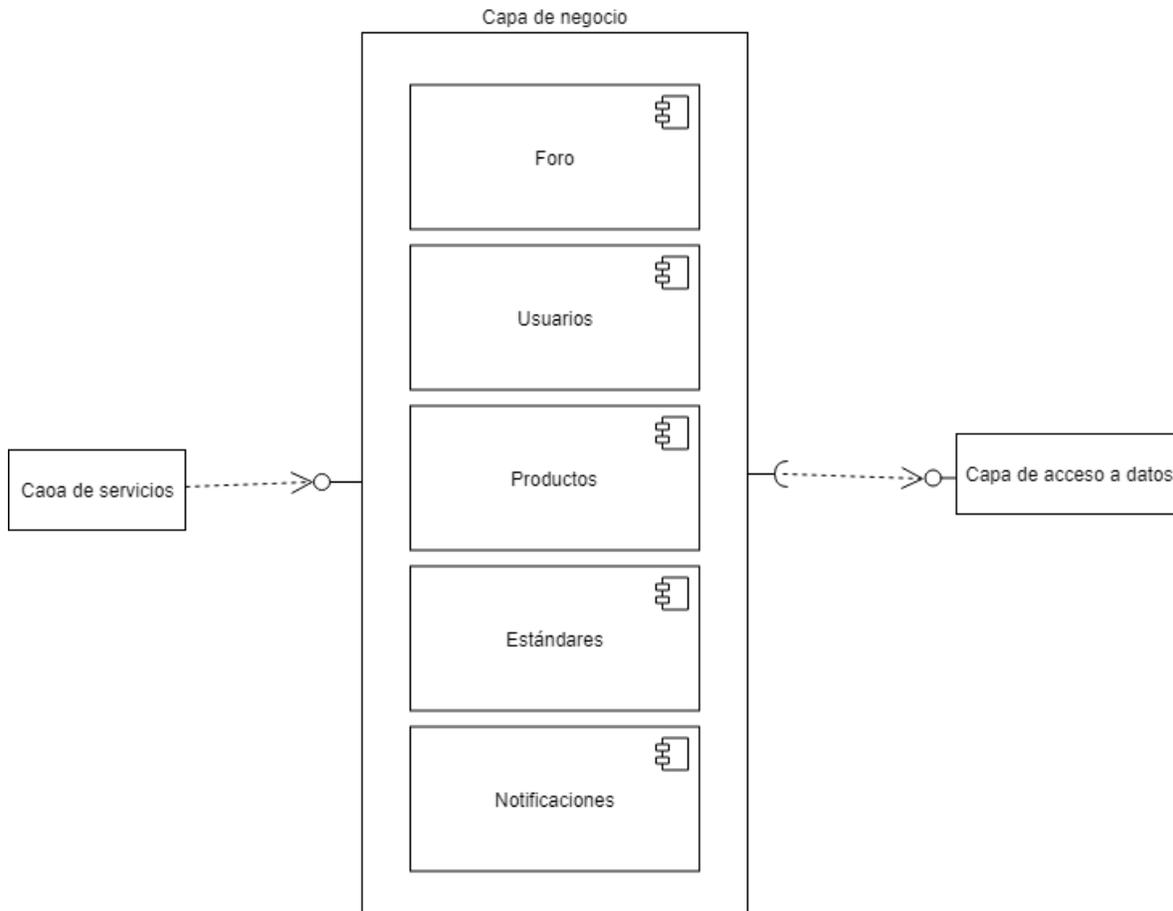


Ilustración 35: Diagrama de componentes de la capa de negocio

La capa de negocio es la encargada del procesamiento y de la traducción entre las peticiones de la capa de servicios y la capa de acceso a datos. Es donde se encuentran definidas todas las reglas de negocio de la solución. Toda esta lógica se encuentra en EJBs stateless ofrecida a través interfaces locales. Esta capa se encuentra dividida por funcionalidades similares como se muestra en la ilustración 35:

- Foro: incluye todo el ciclo de vida de los hilos, los comentarios y las votaciones de los comentarios.
- Usuarios: ciclo de vida de los usuarios, autenticación y gestión de sus conocimientos conjuntamente con las validaciones de los mismos por parte de otros usuarios.
- Productos: gestión de los productos, las sugerencias de nuevos productos, sus estándares implementados y sus calificaciones.
- Estándares: gestiona los estándares
- Notificaciones: es el encargado de enviar notificaciones a los usuarios cuando corresponda. Para esto este componente se integra con la API de mail de Java EE y con la plataforma Firebase [44] que es la encargada de enviar las notificaciones al front-end.

#### 4.2.1.3 Capa de servicios

Esta capa es la encargada de ofrecer la lógica de negocio al front-end. Además permite una fácil integración con otros sistemas, ya sea para recibir datos nuevos como para ofrecer los datos a otras aplicaciones que los necesiten. Estos servicios respetan la arquitectura de servicios web REST [45], centrándose en los recursos a exponer y ofreciendo las 4 operaciones básicas: GET, PUT, POST, DELETE. Para su implementación se utilizó la API de Java EE JAX-RS, en particular se utilizó la librería Jersey que la implementa. En este componente de la solución se agregan filtros para el control de acceso. Específicamente se agrega un filtro para habilitar las peticiones de servidores remotos ya que estas peticiones se encuentran bloqueadas por defecto. La API ofrece tres tipos de operaciones:

- Públicas que no requieren autenticación: son únicamente operaciones de lectura y no presentan ningún control.
- Privadas para los usuarios registrados en el front-end: estas operaciones incluyen las funcionalidades específicas del foro y son sólo accesibles por usuarios logueados a través de su cookie de sesión. Esta cookie contiene dentro el valor de un token jwt.
- Privadas para otros sistemas: como primera integración está el componente de carga de datos. Estas operaciones requieren cabeceras basic auth con el nombre de usuario y la contraseña específica del componente autorizado. El cabezal de autenticación basic auth consiste del nombre de usuario y la contraseña concatenados y codificados. Esta seguridad no suele ser suficiente ya que el encoding no asegura los datos pero en este caso todas las operaciones se realizan mediante https lo que ya encripta toda la comunicación. Estos datos se guardan en la base de datos del sistema para chequear su validez. Existe un sistema por defecto que es el único que puede dar de alta otros sistemas mediante la API.

#### 4.2.2 Base de datos del sistema

La base de datos del sistema es un componente independiente de los otros pero solamente es accedido directamente por la capa de acceso a datos del back-end. Se utiliza una base de datos Postgres relacional. Como mecanismo de seguridad además de la autenticación del usuario dueño también se utiliza control por IP, dejando habilitada únicamente la IP del back-end.

#### 4.2.3 Componente de actualización de datos

El sistema cuenta con dos componentes independientes que gestionan tanto la actualización como la carga de datos al sistema. Para la implementación de los mismos se utiliza Java y funcionan en forma de timer ejecutándose una vez por semana, siendo este tiempo configurable a través de un archivo. A continuación se explicarán en detalle cada uno conjuntamente con las tecnologías utilizadas para implementarlos.

##### 4.2.3.1 Twitter

Esta parte de la solución busca automatizar la búsqueda de actualizaciones de los productos de nuestro sistema. Para su implementación se realizó un proyecto Java con la librería Twitter4J [46], que brinda funcionalidades para la comunicación con la API de la red social. La solución revisa los usuarios de Twitter de los productos buscando por palabras clave que puedan indicar que el mensaje haga referencia a una liberación de una nueva versión del mismo. En caso de detectar posibles candidatos notifica a los administradores, consumiendo la capa de servicios del back-end, dándoles la información del producto y el contenido del tuit.

##### 4.2.3.2 Web Scraping

Este componente es el encargado de buscar en diferentes sitios web nuevos estándares o tecnologías de forma automática. Para esto se utilizó una librería de Java llamada JSOUP[47] que brinda funciones para parsear el HTML a través de sus diferentes etiquetas y atributos.

Debido al dinamismo presente en las páginas web se busca una solución que sea fácilmente actualizable por usuarios no técnicos para lograr el mantenimiento de la solución. Se utilizan tres recursos diferentes para lograr cumplir este objetivo:

- Aplicación extensible mediante librerías: se desarrolla un aplicación que permite cargar nuevas librerías dinámicamente que extiendan la lógica de la misma. Para esto el proyecto central es simplemente el hilo principal con un temporizador para su ejecución semanal que invoca a cada una de las librerías que se encuentren actualmente en el proyecto. Cada librería se encarga de extraer información de un sitio web diferente, mapearla al modelo de datos de la solución, verificar si la información ya existe en el sistema o si es nueva y en caso de que sea nueva enviarla a la aplicación central para que se actualice. Esto permite que si algún desarrollador desea nutrir la aplicación de un sitio nuevo diferente pueda simplemente agregar una librería a la solución sin necesidad de modificar código existente ni realizar un despliegue nuevo.

- Parsehub[48]: esta herramienta permite a usuarios mediante una interfaz gráfica navegar la web y seleccionar la información que les resulte relevante para su extracción automática. De esta forma facilita el mantenimiento del componente por usuarios que no sean desarrolladores. La solución utiliza JSOUP para complementar los casos que no se pueden cubrir con la suscripción gratuita.
- Archivo de configuración: se utiliza un archivo para guardar determinadas configuraciones que puedan cambiar con el tiempo. Entre estas se encuentran los endpoint de los servicios a consumir tanto de la aplicación central como la de Parsehub. También se encuentran expresiones regulares que se utilizan para el mapeo entre los datos obtenidos desde la web y el modelo de datos propio. A continuación se muestra el fragmento del archivo que guarda las expresiones regulares.

```
#Regex used to scrap
implementedStdVersionRegex = (((\d)+(\.\.)*+)(\s+early implementer)*$)
standardVersionGroup = 2
implementedStdReferentVersionRegex = (((\d)+(\.\.)*+)(\s)(official reference implementation)(\s+early implementer)*$)
implementedStdReferentGroup = 2
certifiedComplianceStatus = ^compliant.*(\d\d\d\d\d\d).* (\d\d\d\d).* (\d\d\d\d).*
certifiedComplianceYearGroup = 1
certifiedComplianceMonthGroup = 2
certifiedComplianceDayGroup = 3

#Regex of standards versions types
typesUsedRegex = ^(d-is)$|^ (is)$
deprecatedTypeRegex = ^(d-is)$
```

*Ilustración 36:* Extracto del archivo de configuración del componente Web Scraping

Este componente se utiliza tanto para la actualización de la información como para la carga inicial de la misma. Está dividido en dos grandes partes: la encargada de obtener información de productos y la encargada de obtener información de los estándares.

Antes de realizar la implementación se obtuvieron los archivos robots.txt de cada página a visitar y se verificó que estuviese permitido el acceso. En el caso de la página del OGC se solicita a los robots que utilicen una demora de 10 segundos entre petición y petición. Esto se configura mediante la herramienta Parsehub y en el caso de las peticiones que se hacen mediante JSOUP un timer se asegura que se respete dicho tiempo.

#### 4.2.3.3 Carga y actualización de estándares

Para la obtención de los estándares se utiliza únicamente el sitio web oficial de la página de la OGC. Para lograr la carga de los datos se siguen los siguientes pasos:

1. La carga de datos comienza con la herramienta Parsehub donde se extrae la información necesaria de la página del OGC. Para esto primero ingresa a la página principal donde obtiene todos los nombres de los estándares y los links a la página específica de cada uno.
2. Luego desde la herramienta Parsehub se accede a cada uno de los links específico de los estándares para obtener el resto de su información: descripción y versiones. Las versiones se listan en una tabla bajo el título de descargas. Dentro de esa tabla se encuentran todos los tipos de documentos relacionados al estándar, pero de los listados el que se busca

- extraer es únicamente el de tipo “especificación de la implementación” (IS por su sigla en inglés) y también se guardan las implementaciones deprecadas (DIS por su sigla en inglés).
3. Se continúa en la librería específica encargada del procesamiento de estándares. Esta librería primero consume los servicios de la aplicación central para obtener la información de los estándares ya cargados en el sistema.
  4. Luego consume la API de Parsehub para obtener la última extracción disponible.
  5. Comienza el mapeo de la información obtenida desde Parsehub y el modelo de la solución. Luego de obtenidos los estándares en el formato adecuado se los compara con los encontrados en el sistema central.
  6. Todos los estándares nuevos que encuentra los carga en el sistema, si el estándar tiene toda la información (no le falta descripción ni versión) se guarda definitivamente, de lo contrario se lo guarda como una sugerencia para los administradores.
  7. Todos los estándares con diferencias son registrados y se envía la información de los mismos al sistema central, que luego la envía vía mail a los administradores del sistema para que tomen una decisión.

#### 4.2.3.4 *Carga y actualización de productos*

En el caso de los productos se releva información de las siguientes páginas web: Wikipedia, OGC, OSGeo. Este proceso es más complejo que la carga de los estándares debido a la diversidad de fuentes y a la cantidad de información que requiere el sistema dentro de cada producto. El proceso de carga es muy similar al de los estándares pero con algunas diferencias:

- El mapeo entre el modelo de datos del sistema y los datos obtenidos de Parsehub incluye varios desafíos
  - Clasificar el producto según la clasificación utilizada en la solución. Para esto en el sistema se guarda, en cada tipo de producto, una expresión regular que permite realizar un mapeo entre lo encontrado y los tipos guardados.
  - Obtener los estándares implementados y distinguir entre sus diferentes grados de cumplimiento. Esto se realiza únicamente en el caso del sitio del OGC, ya que los productos registrados como implementadores se encuentran obligatoriamente allí. Para esto también se utiliza una expresión regular guardada en el archivo de configuración que permite distinguir grado de cumplimiento.
  - Asociar los estándares implementados encontrados con los estándares guardados en el sistema.
  - Obtener información de sitios web muy diferentes: cada sitio web a utilizar tiene su propia biblioteca con su lógica independiente que logra realizar el mapeo de los datos obtenidos desde Parsehub. Las diferencias entre las diferentes páginas se pueden apreciar en las imágenes 38, 39 y 40

## Certified and Implementing Products

Register now or update your product listing

Help

Standard: -

Product Provider: -

Compliant Products Only  Reference Implementations Only  All Implementations

Found 63 organizations with 291 products with 969 standard implementations, 969 of which are currently compliant, and 64 are reference implementations

### Action Corporation (view all products)

Top ▲

Ingres Database 11.0	Alex Trofast	Registered: 2017-03-14
Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option 1.1		Certified: 2017-03-30
Simple Features - SQL - Types and Functions 1.1		Certified: 2017-03-30
Ingres Database 10.2	Alex Trofast	Registered: 2014-08-11
Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option 1.1		Certified: 2014-09-05
Simple Features - SQL - Types and Functions 1.1		Certified: 2014-09-05

### airGmap Aerospace Technology Limited (view all products)

Top ▲

OCCS Server 10.1	Fang, Nita	Registered: 2011-07-01
Web Map Service (WMS) Implementation Specification 1.3.0		Certified: 2011-07-01
Web Map Service 1.1.1		Certified: 2011-07-05
OCCS Server 10	Fang, Nita	Registered: 2010-12-01
Web Map Service 1.1.1		Certified: 2010-12-02
OIMS Server 10.1	Fang, Nita	Registered: 2011-07-01
Web Map Service (WMS) Implementation Specification 1.3.0		Certified: 2011-07-01
Web Map Service 1.1.1		Certified: 2011-07-05
OIMS Server 10	Fang, Nita	Registered: 2010-04-08

Ilustración 37: Interfaz gráfica de la página web de la OGC de los productos[2]

Languages

- Français
- 繁體
- Italiano
- Português
- 中文

[Edit links](#)

### Open source software [edit]

The development of open source GIS software has—in terms of software history—a long tradition<sup>[2]</sup> with the appearance of a first system in 1978. Numerous systems are available which cover all sectors of geospatial data handling.

### Desktop GIS [edit]

The following open-source desktop GIS projects are reviewed in Steiniger and Bocher (2008/9)<sup>[3]</sup>

- GRASS GIS – Originally developed by the U.S. Army Corps of Engineers: a complete GIS.
- gvSIG – Written in Java. Runs on Linux, Unix, Mac OS X and Windows.
- ILWIS (Integrated Land and Water Information System) – Integrates image, vector and thematic data
- JUMP GIS / OpenJUMP ((Open) Java Unified Mapping Platform) – The desktop GISs OpenJUMP, SkyJUMP, deeJUMP and Kosmo all emerged from JUMP.<sup>[3]</sup>
- MapWindow GIS – Free desktop application and programming component.
- QGIS (previously known as Quantum GIS) – Runs on Linux, Unix, Mac OS X and Windows.
- SAGA GIS (System for Automated Geoscientific Analysis) – A hybrid GIS software. Has a unique Application Programming Interface (API) and a fast-growing set of geoscientific methods, bundled in exchangeable Module Libraries.
- uDig – API and source code (Java) available.

Besides these, there are other open source GIS tools:

- Capaware – A C++ 3D GIS Framework with a multiple plugin architecture for geographic graphical analysis and visualization.
- FalconView – A mapping system created by the Georgia Tech Research Institute for the Windows family of operating systems. A free, open source version is available.
- Kalypso – Uses Java and GML3. Focuses mainly on numerical simulations in water management.
- TerraView – Handles vector and raster data stored in a relational or geo-relational database, i.e. a frontend for TerraLib.
- Whitebox GAT – Cross-platform, free and open-source GIS software.

### Other geospatial tools [edit]

Apart from desktop GIS, many other types of GIS software exist. A general overview of GIS software projects for each category was done in 2012.<sup>[4]</sup> Below is a similar listing of open source GIS projects.

### Web map servers [edit]

- GeoServer – Written in Java and relies on GeoTools. Allows users to share and edit geospatial data.
- MapGuide Open Source – Runs on Linux or Windows, supports Apache and IIS web servers, and has APIs (PHP, .NET, Java, and JavaScript) for application development.
- Mapnik – C++/Python library for rendering - used by OpenStreetMap.

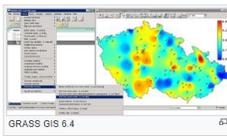




Ilustración 38: interfaz gráfica de la página de Wikipedia de los productos [49]

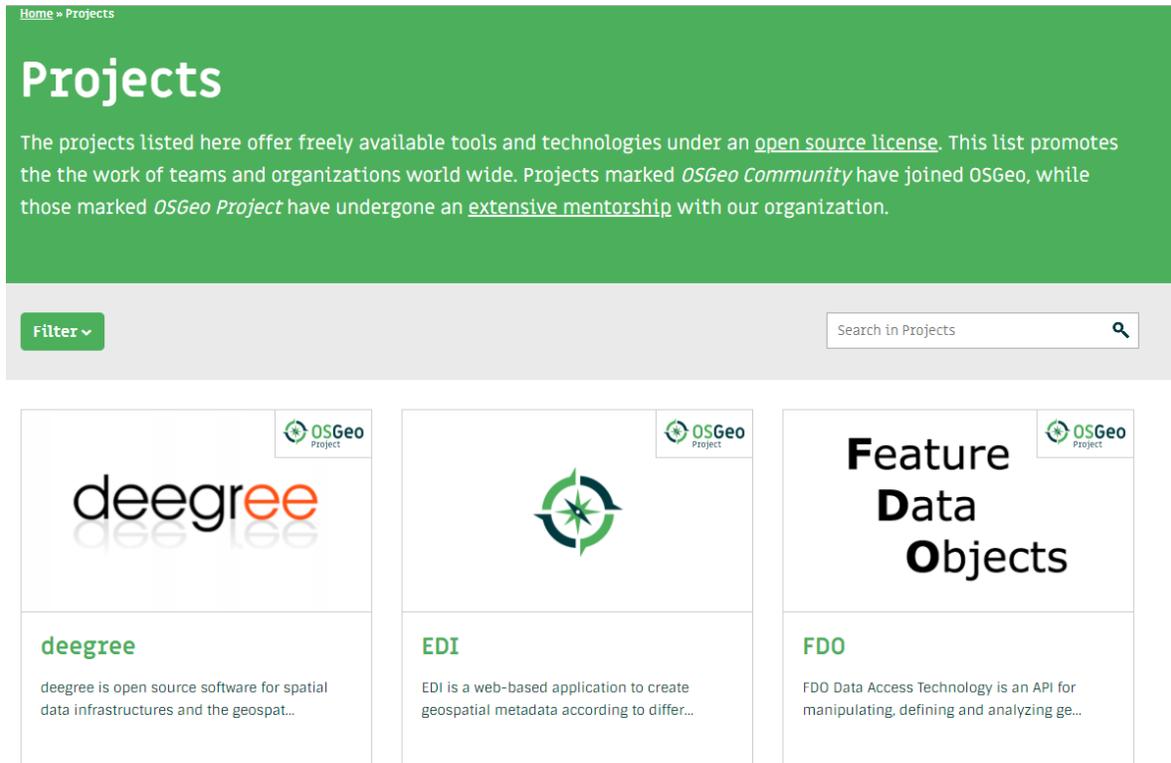


Ilustración 39: Interfaz gráfica de la página de la OSGeo de los productos [50]

A continuación se detalla un ejemplo del flujo de la carga de un producto en particular (Awila) de la web del OGC:

1. Parsehub obtiene la información referente a los estándares implementados, la compañía y el link a los detalles del producto como se muestra en la ilustración 40, en este caso:

Augmented Technologies Ltd (view all products)		Top ▲
Awila 1.4.0	Reynolds, Mike	Registered: 2013-03-01
GeoAPI 3.0 Implementation Standard 3.0		
Geography Markup Language 2.0		
Geography Markup Language 2.1.1		
Web Feature Service (WFS) Implementation Specification 1.1.0		
Web Map Tile Service Implementation Standard 1.0.0		
Web Feature Service 1.0.0		
Web Map Service 1.0		
Web Map Service 1.1.1		

Ilustración 40: Sección de la compañía Augemented Technologies del OGC [2]

- a. Se toma el texto de los estándares implementados y se los mapea a los estándares presentes en el sistema: GeoAPI en su versión 3.0, GML en sus versiones 2.0 y 2.1.1 (deprecada), WFS en sus versiones 1.0.0 (deprecada) y 1.1.0, WMS en sus versiones 1.0 y 1.1.1 (ambas deprecadas) y Web Map Tile Service en su versión 1.0.0. Como se muestra en la ilustración 40, el nombre de los estándares varía incluso dentro de la misma tabla: Web Feature Service (WFS) Implementation Specification y Web Feature Service. Estas inconsistencias dificultaron el mapeo de los mismos con los presentes en el sistema.
  - b. La compañía que lo desarrolla es: “Augmented Technologies Ltd”
2. Utilizando JSOUP se obtienen el resto de los datos del producto como se muestra en la ilustración 41:

### Compliance Navigation

- [Get Certified](#)
- [Register Product](#)
- [Renew Certification](#)

## OGC Implementing/Compliant Product Details

### Awila Product Details

Name:	<b>Awila</b>
Version:	<b>1.4.0</b>
Type:	<b>Client</b>
Implemented:	<b>2013-03-01</b>
URL:	<a href="#">Product Website</a>
Platform(s):	<b>Android</b>
Contact:	<a href="#">Reynolds, Mike - Augmented Technologies Ltd</a>

Implemented Specifications / Interfaces / Alternatives	OGC Compliance Status
GeoAPI 3.0 Implementation Standard 3.0	<a href="#">Implementing</a>
Geography Markup Language 2.0	<a href="#">Implementing</a>
Geography Markup Language 2.1.1	<a href="#">Implementing</a>
Web Feature Service 1.0.0	<a href="#">Implementing</a>
Web Feature Service (WFS) Implementation Specification 1.1.0	<a href="#">Implementing</a>
Web Map Service 1.0	<a href="#">Implementing</a>
Web Map Service 1.1.1	<a href="#">Implementing</a>
Web Map Tile Service Implementation Standard 1.0.0	<a href="#">Implementing</a>

Ilustración 41: Detalle del producto Awila en el sitio del OGC [2]

- a. El nombre, la versión, el link a la página oficial
  - b. Utilizando el tipo y las plataformas se lo mapea a componente de GUI móvil
  - c. En este caso los estándares no cuentan ni con certificación ni el producto es un implementador referente, esto se desprende de la columna “OGC Compliance Status” de la tabla presente en la ilustración 41.
3. Se consulta la base de datos local para verificar si existe una tecnología con estas características que ya haya sido enviada al sistema. En caso que no exista o haya diferencias continúa el proceso.
  4. Luego se consulta la página oficial para buscar información faltante como: soporte (utilizando la fecha de la última modificación) y usuario de twitter. En este caso el link presente en la página del OGC dirige a una página que no existe (<http://www.augmented-technologies.com/products.html>), evitando completar la información básica, terminando esta carga como una sugerencia para que el administrador del sistema pueda realizar las correcciones correspondientes.

#### 4.2.4 Front-end

El componente de front-end refiere a la capa de presentación señalada en el diagrama de la solución (ilustración 28). Se encarga de presentarle la información al usuario de una forma amigable y organizada; con este componente se logra que el usuario pueda acceder a todas las funcionalidades que el sistema tiene disponibles para él de una manera sencilla y fácil de comprender.

Debido a que este sistema apunta al uso masivo y general de la misma, se decidió adoptar el enfoque mobile first para su implementación, haciendo que el sistema sea responsive y así facilitar el acceso a la plataforma desde cualquier lugar por cualquier usuario.

Para poder llevar a cabo esta implementación se optó por utilizar el framework bootstrap 3 debido a su gran popularidad y a las facilidades que brinda para adaptar la interfaz gráfica de los componentes web a cualquier tamaño de pantalla.

Además para el desarrollo del front-end se utilizó el framework Angular 5. Este framework es robusto y permite un desarrollo sin necesidad de tener procesamiento del lado del servidor, logrando componentes ágiles. Debido a la popularidad del mismo, existen muchos componentes desarrollados por terceros (módulos) para realizar determinadas integraciones que facilitan el desarrollo de la aplicación. En este caso se utilizaron los siguientes componentes:

- Para permitir que los usuarios ingresen al sistema utilizando sus cuentas de google se incorporó al sistema el módulo “angular4-social-login”. El funcionamiento del mismo es sencillo, luego de algunas configuraciones para que dicha plataforma acepte a nuestro sistema, una vez que el usuario decide ingresar por Gmail el sistema lo redirige al login de la plataforma y una vez que el usuario se loguea, nuestro sistema puede utilizar sus datos para mantener su sesión en GISAdvisor.
- Otro módulo externo que se incorporó al front-end es el “ng2-cloudinary” que permite la integración con el sistema cloudinary para subir las imágenes de los usuarios y las tecnologías y poder mostrarlas cuando sea necesario.

## 4.3 Implementación

### 4.3.1 Autenticación y Autorización

La autenticación, es el proceso por el cual se identifica un usuario como válido, para posteriormente acceder a ciertos recursos definidos, mientras que la autorización es el proceso sobre el cual se establecen qué tipos de recursos están permitidos o denegados para cierto usuario o grupo de usuarios concreto.

En el back-end, la autenticación del usuario se realiza utilizando el esquema basic auth. Esto se utiliza o bien por los sistemas externos que consumen servicios o durante el login de los usuarios del propio sistema. En este último caso se autentica al usuario y se le asigna una cookie con un JWT. Luego se utiliza esta cookie para la posterior autorización.

En el front-end se utiliza la cookie y Route Guards para realizar el control de acceso.

#### 4.3.1.1 Hashing de contraseñas

El back-end para evitar problemas de seguridad cuando se registra un usuario en el sistema no guarda su contraseña en texto plano, sino que guarda un hash de la misma. El hash es una función criptográfica que se caracteriza por ser su irreversibilidad, es decir, calcular el hash de un valor es sencillo pero dado un hash calcular su valor original es computacionalmente imposible. De esta forma resulta seguro guardar el hash de las contraseñas de los usuarios, ya que si alguien logra acceder a los datos del sistema, igualmente no puede utilizarlos para iniciar sesión porque no tiene forma de conocer la contraseña sin hash. Se utiliza el algoritmo SHA 512 para realizar el hash. Para autenticar un usuario, el back-end parte de la contraseña recibida, le realiza un hash a la misma y la compara con la contraseña guardada en la base de datos, si coinciden el usuario accede de lo contrario se le niega el acceso.

```
private String hashPassword(String passwordToHash) {
    String generatedPassword = null;
    try {
        MessageDigest md = MessageDigest.getInstance("SHA-512");
        byte[] bytes = md.digest(passwordToHash.getBytes("UTF-8"));
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < bytes.length; i++) {
            sb.append(Integer.toString((bytes[i] & 0xff) + 0x100, 16).substring(1));
        }
        generatedPassword = sb.toString();
    } catch (NoSuchAlgorithmException | UnsupportedEncodingException e) {
        Logger.getLogger(UserController.class)
            .getName().log(Level.SEVERE, null, e);
    }
    return generatedPassword;
}
```

Ilustración 42: Función de hashing de la contraseña



#### 4.3.1.4 Cookies

Para el envío de los tokens generados en el back-end se utilizan las cookies de sesión. Estas cookies no tienen fecha de expiración y son enviadas al front-end una vez que un usuario se autentica. En ese momento el navegador almacena la misma automáticamente y la envía en todas las peticiones al back-end, quien extrae de la misma el valor del jwt y continúa con su proceso de autorización. Esta cookie se destruye cuando el usuario realiza la operación de logout. Como el servidor no puede eliminar la cookie de los navegadores, cuando el usuario solicita desloguearse, el servidor sobrescribe la cookie utilizada: le agrega fecha de caducidad con una fecha anterior a la actual y reemplaza el valor del token por un valor no válido.

#### 4.3.1.5 Route Guards

Son interfaces que permiten decirle al enrutador si se debe o no permitir la navegación a una ruta solicitada, toman esta decisión buscando un valor de retorno ya sea verdadero o falso de una clase que implementa la interfaz del Route Guard dado.

Hay cinco tipos diferentes y el comportamiento del enrutador depende de qué Route Guard se utilice: CanActivate, CanActivateChild, CanDeactivate, CanLoad, Resolve.

La interfaz que se utilizó en este caso es “CanActivate” la cual devuelve un booleano que indica si se debe permitir o no la navegación a una ruta.

Para hacer uso de la misma se desarrolló un servicio llamado “AuthGuard” que implementa dicha interfaz el cual verifica si el usuario está autenticado o no, es decir si el JWT está guardado en el almacenamiento local y así se puede utilizar la Guard en cualquier ruta que se quiera proteger.

A su vez también se creó otro servicio llamado “RoleGuard” que también implementa la interfaz mencionada anteriormente, en este caso en vez de verificar únicamente si existe un usuario logueado o no, se verifica el rol de dicho Usuario. Si este rol coincide con el rol acepta el servicio entonces el usuario puede acceder a la ruta deseada, en otro caso el usuario es re direccionado a la página de inicio.

Gracias a dichas implementaciones se puede utilizar la Route Guard escogida en cualquier ruta que se quiera proteger. A continuación se muestran las rutas existentes en el sistema desarrollado y las configuraciones que presenta cada una de las mismas:

```

const routes: Routes = [
  { path: 'home', component: HomeComponent },
  { path: 'notifications', component: NotificationsComponent },
  { path: 'userProfile', component: ProfileComponent, canActivate: [AuthGuard] },
  { path: 'publicProfile/:id', component: PublicProfileComponent, canActivate: [AuthGuard] },
  { path: 'community', component: CommunityComponent, canActivate: [AuthGuard] },

  { path: 'standards', component: StandardsComponent },
  { path: 'standards/:name/:version', component: StandardDetailComponent },
  { path: 'standards/:mode/:name/:version', component: StandardCreateComponent }, // mode update
  {
    path: 'standards/:mode', component: StandardCreateComponent, canActivate: [RoleGuard], // mode insert
    data: {
      expectedRole: 'admin'
    }
  },

  {
    path: 'pendingStandards', component: StandardsComponent, canActivate: [RoleGuard],
    data: {
      expectedRole: 'admin'
    }
  },

  { path: 'products', component: ProductsComponent },
  { path: 'product/:id', component: ProductDetailComponent }, // mode view
  { path: 'products/:mode/:id', component: ProductCreateComponent, canActivate: [AuthGuard] }, // mode update
  { path: 'products/:mode', component: ProductCreateComponent, canActivate: [AuthGuard] }, // mode insert

  {
    path: 'suggestions', component: ProductsComponent, canActivate: [RoleGuard],
    data: {
      expectedRole: 'admin'
    }
  },

  { path: 'compare', component: ProductCompareComponent },
  { path: 'forum', component: ForumComponent },
  { path: 'reports', component: ReportComponent },

  { path: 'thread/:id', component: ThreadComponent },

  { path: '', redirectTo: 'home', pathMatch: 'full' },
  { path: '**', redirectTo: '' }
];

```

Ilustración 44: Configuración de las rutas del front-end

Por ejemplo, se observa el path “userProfile” que muestra la página del perfil del usuario, al agregar el atributo “canActivate: [AuthGuard]” se indica que para acceder a esta página debe haber un usuario logueado en el sistema. Podemos observar, por otro lado, en el path “suggestions” (que muestra las sugerencias de nuevos productos) el uso del atributo “canActivate: [RoleGuard]” junto con el rol esperado, lo que significa que solo los usuarios con rol de administradores pueden acceder a esta página.

Si bien cualquier usuario podría acceder al JWT, cambiar su carga útil y de esa forma “pasar” los Route Guard implementados, cualquier cambio que realice el usuario en la carga útil del token hará que se invalide la firma. Las firmas JSON Web Token se calculan al incluir la carga, por lo que cualquier modificación a la misma hace que el JWT quede invalidado y el usuario no podrá acceder a los recursos que están protegidos con dicho JWT.

## 4.3.2 Comunidad de usuarios

### 4.3.2.1 Idioma

El sistema cuenta con la posibilidad de que el usuario escoja el idioma del mismo. Actualmente el idioma por defecto es el inglés, teniendo la posibilidad de cambiarlo a español. El idioma escogido por el usuario será recordado mientras dure la sesión.

Para llevar a cabo dicha funcionalidad se utilizó la librería “ngx-translate” la cual permite definir traducciones en diferentes idiomas y cambiar entre ellas de forma sencilla. Las traducciones se realizan en archivos del tipo JSON, cada archivo es un idioma por lo que agregar un nuevo idioma al sistema consiste únicamente en agregar un nuevo archivo con las traducciones correspondientes.

En la ilustración 45 se muestra una sección de los archivos JSON para español e inglés, como se puede observar, los archivos tienen el mismo formato pero en el valor de cada etiqueta se ingresa la traducción correspondiente.

<pre> "Home1": "The world's first observatory for GIS developers.", "Home2": "Learn about new products and standards, interact with other users", "Home3": "and share your experiences in GISAdvisor's integrated forum.", "Home": {   "Title": "Home" }, "Architecture": "Reference architecture", "ArchitectureButton": "View Reference architecture",  "NavBar": {   "Register": "Register",   "Login": "Login",   "Logout": "Logout",   "Profile": "User Profile",   "RegisterModal": {     "Title": "Register",     "Name": "Name",     "Email": "Email",     "UserName": "Username",     "Sex": "Female",     "SexM": "Male",     "BirthDate": "Birthdate",     "Pass": "Password",     "Country": "Country",     "ConfirmPass": "Confirm password",     "About": "About me",     "RegisterButton": "Register",     "ErrorName": "Name is required.",     "ErrorEmail": "Email is required.",     "ErrorEmail2": "Email incorrect format.",     "ErrorUserName": "Username is required.",     "ErrorPass": "Password is required.",     "ErrorConfirmPass": "Confirm Password is required.",     "ErrorMatch": "passwords do not match try again."   } } </pre>	<pre> "Home1": "El primer observatorio del mundo para desarrolladores de SIG.", "Home2": "Aprende sobre nuevos productos y estándares, interactúa con otros usuarios", "Home3": "y comparte tus experiencias en el foro integrado de GISAdvisor.", "Home": {   "Title": "Inicio" }, "Architecture": "Arquitectura de referencia", "ArchitectureButton": "Ver arquitectura de referencia",  "NavBar": {   "Register": "Registrarse",   "Login": "Ingresar",   "Logout": "Salir",   "Profile": "Ver Perfil",   "RegisterModal": {     "Title": "Registrar",     "Name": "Nombre",     "Email": "Email",     "UserName": "Nombre de usuario",     "Sex": "Mujer",     "SexM": "Hombre",     "BirthDate": "Fecha de nacimiento",     "Pass": "Contraseña",     "Country": "País",     "ConfirmPass": "Repetir contraseña",     "About": "Descripción",     "RegisterButton": "Registrarse",     "ErrorName": "El nombre es obligatorio.",     "ErrorEmail": "El Email es obligatorio.",     "ErrorEmail2": "Email con formato incorrecto.",     "ErrorUserName": "El nombre de usuario es obligatorio.",     "ErrorPass": "La contraseña es obligatorio.",     "ErrorConfirmPass": "Te falta repetir la contraseña.",     "ErrorMatch": "las contraseñas no coinciden, intente nuevamente."   } } </pre>
---	--

Ilustración 45: Extracto de archivo JSON con textos en los dos idiomas

#### 4.3.2.2 Foro

Como se explicó en el capítulo anterior el foro consiste en realizar un intercambio de información entre los usuarios del sistema así como aportar información para usuarios que no estén registrados en el mismo.



Ilustración 46: Interfaz gráfica del componente de foro de la solución

Las capturas anteriores corresponden a la vista de un usuario no logueado en el sistema.

La solución planteada para llevar a cabo dicho requerimiento fue la siguiente: en primer lugar se creó una pantalla en donde se listan todos los hilos en el sistema, ya sean hilos abiertos o cerrados (la cual corresponde a la primer captura), la misma cuenta con un filtro para poder buscar dentro de todos los hilos existentes y otro para poder filtrar por hilos activos o cerrados. Para el caso de los administradores en cada hilo se muestran botones para cerrar/abrir o eliminar el hilo.

Por otro lado, se creó otra pantalla que muestra el detalle de cada hilo (la cual corresponde a la segunda captura), es decir se muestra la descripción del mismo, en donde el usuario que inicializa el hilo describe el problema/solución que quiere compartir con la comunidad SIG y sus respectivos comentarios. Para lograr que tanto en los comentarios como en la descripción de los hilos se agreguen diferentes estilos se utilizó el componente CKEditor, editor de texto enriquecido WYSIWYG(“what you see is what you get”) que permite escribir contenido directamente dentro de páginas web o aplicaciones en línea, el mismo permite por ejemplo agregar imágenes al foro.

En el caso de los usuarios logueados, ya sea administradores o colaboradores se pueden agregar comentarios a los distintos hilos que se encuentren abiertos así como votar los comentarios de otros usuarios. Los comentarios más votados son los que se muestran más arriba en la lista de respuestas al hilo.

#### 4.3.2.3 Reviews

Las reviews son comentarios que los usuarios logueados realizan sobre cada producto. Un usuario puede únicamente efectuar un comentario por producto el cual va acompañado por una calificación del 1 al 5. Las siguientes capturas de pantallas muestran cómo se ingresan nuevos comentarios a un producto y la lista de comentarios del mismo.

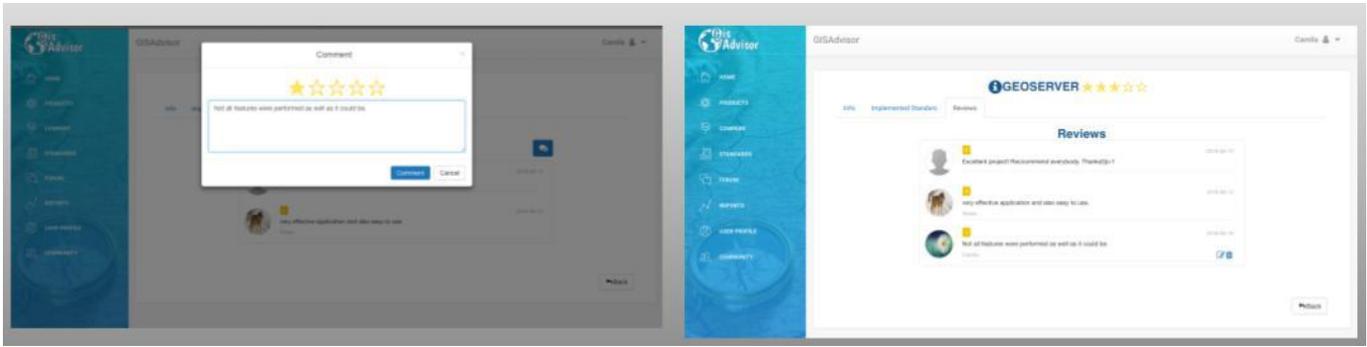


Ilustración 47: Interfaz gráfica de las reviews presentes en la solución

Los comentarios son visibles para todos los usuarios del sistema aun si estos no están registrados en el mismo pero únicamente los usuarios logueados pueden hacer comentarios. En el caso de que el usuario ya haya realizado un comentario, el usuario puede eliminarlo o editarlo. A diferencia de los colaboradores los usuarios administradores pueden eliminar cualquier comentario de cualquier producto sin importar quien lo realizó. Cada producto tiene una calificación genérica la cual se calcula como un promedio de todas las calificaciones que los usuarios ingresan junto con sus comentarios.

#### 4.3.2.4 Conocimientos

Tal como se relevó en los requerimientos del sistema, se debe permitir a los usuarios asociarse conocimientos que se correspondan con los productos del sistema, lo que permite, mostrar a los otros usuarios su nivel de manejo de un determinado producto. A su vez, el resto de la comunidad de usuarios podrán validar esos conocimientos, lo que significa que afirman que el usuario que se lo asignó realmente cuenta con el mismo.

La solución planteada para resolver la sección de este requerimiento que refiere a la posibilidad de un usuario de agregarse conocimientos, se resume en el flujo reflejado en la ilustración 48.

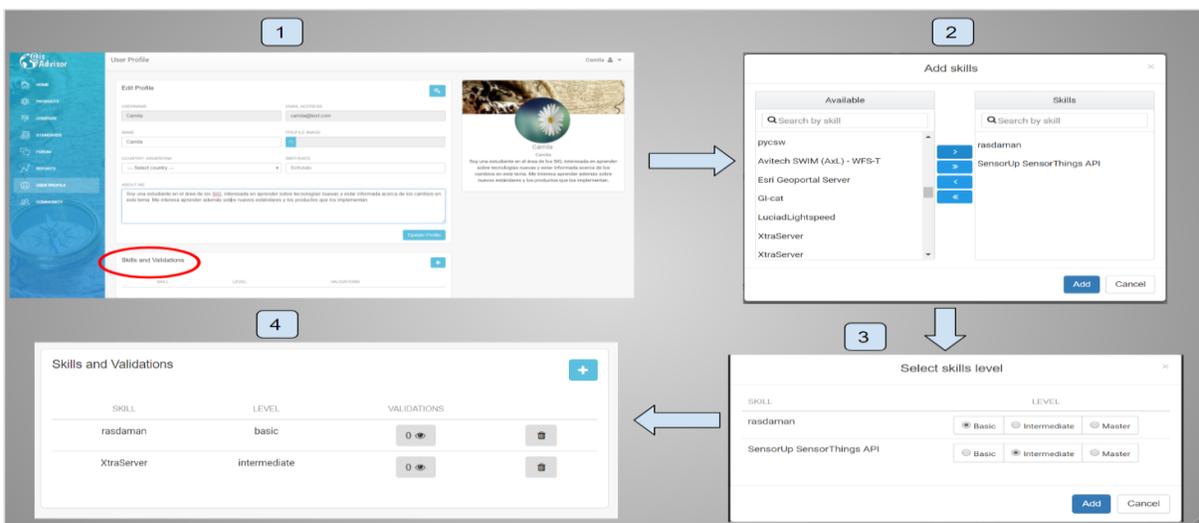


Ilustración 48: flujo de agregado y validación de conocimientos de la solución.

La primera imagen del flujo corresponde a la vista del perfil del usuario, la sección señalada con rojo es una lista donde se muestran los conocimientos del usuario y las validaciones de cada uno. Se le da la opción al usuario de agregar nuevos conocimientos, y al seleccionar esa opción se abre el cuadro indicado en la imagen 2 del flujo. Un aspecto interesante a destacar en cuanto a la implementación de esta funcionalidad es que se permite agregar varios conocimientos al mismo tiempo, todos los que se encuentren en la lista de serán los nuevos conocimientos del usuario. La doble lista que se muestra en la imagen 2 es una picklist, componente de la interfaz de usuario, la cual, se encuentra disponible en el conjunto de componentes primeNG [55] incluido en angular (es nativo de angular). Esta picklist permite, tener de un lado todos los productos disponibles y con los controles entre las dos listas se pueden pasar a la segunda lista los productos que se quieren seleccionar. Para cargar la lista de productos disponibles, en primer lugar se traen todos los productos del sistema, y luego se filtran con los conocimientos que ya tiene asociados el usuario (para evitar duplicados).

Una vez seleccionados todos los productos para los cuales se quiere agregar un conocimiento se pasa a la imagen 3 del flujo, en donde se permite elegir el nivel de conocimiento para cada uno de los productos. Esto se resolvió con inputs del tipo “radio” ya que un usuario puede tener un único nivel por producto. Luego de seleccionar el nivel para cada producto seleccionado se vuelve a mostrar la vista del perfil de usuario, y en la lista de conocimientos ya se muestran los nuevos (como indica la imagen 4 del flujo que se muestra en la ilustración 48). Para resolver la validación de los conocimientos de otro usuario, en primer lugar, el sistema controla que un usuario no pueda validar sus propios conocimientos, lo cual no tendría sentido alguno, y además se controla que un usuario pueda realizar sólo una validación por conocimiento.

Para poder desplegar esta información se crea un perfil público de los usuarios que muestra: nombre, país, rol en el sistema (si es administrador o colaborador), descripción y los conocimientos del mismo. Esta selección de la información a mostrar se hizo para evitar divulgar información sensible o de índole privada. Hay que destacar que una vez que el usuario valida un conocimiento, automáticamente desaparece el control para validarlo. Esto se logra utilizando la directiva \*ngIf de angular, la cual se incluye en el HTML y permite incluir o no cierto elemento en el DOM dependiendo de una condición.

#### 4.3.2.5 *Sugerencia de producto*

Como se explicó anteriormente, los usuarios colaboradores pueden sugerir nuevos productos para el sistema. Luego de que un administrador apruebe una sugerencia, el producto pasa a formar parte del sistema y se les mostrará a los usuarios en la lista de productos.

La solución que se plantea, para la sugerencia de un nuevo producto, es un formulario formado por los campos de las siguientes imágenes:

The screenshot shows the 'Suggest Product' interface. On the left is a navigation menu with options: HOME, PRODUCTS, COMPARE, STANDARDS, FORUM, REPORTS, USER PROFILE, and COMMUNITY. The main form contains the following fields:

- Name:** Text input field with a red border.
- Version:** Text input field with a red border.
- Component Type:** Dropdown menu with a red border and the text '-- Select --'.
- Organization:** Text input field.
- Creation Date:** Text input field containing '2018-06-18'.
- Free/Commercial:** Dropdown menu with a red border and the text '-- Select --'.
- Support:** Dropdown menu with the text 'Without Info'.
- OpenSource:** Dropdown menu with the text 'Without Info'.
- Web Site:** Text input field.
- Description:** Large text area for product details.

Ilustración 49: Interfaz gráfica de la solución para la sugerencia de productos

The screenshot shows the 'Standards' interface. On the left is a navigation menu with options: COMPARE, STANDARDS, FORUM, REPORTS, USER PROFILE, and COMMUNITY. The main form contains two panels:

- Available:** A list of standards with a search bar. The list includes:
  - 3D Portrayal Service - 1.0
  - Augmented Reality Markup Language 2.0 (ARML 2.0) - 1.0
  - Catalogue Services Standard 2.0 Extension Package for ebRIM Application Profile: Earth Observation Products - 1.0.0
  - Catalogue Service - 3.0
  - Catalogue Service - 2.0.2
- Standards:** An empty list with a search bar.

Navigation buttons (right, left, back) are located between the panels. At the bottom right, there are 'Suggest' and 'Back' buttons. The footer includes 'Home', 'Contact us', and 'Proyecto de Grado © 2018 GisAdvisor, Verónica, Cristina y Sabrina.'

Ilustración 50: Interfaz gráfica de la solución para la sugerencia de productos

Se señalan con un borde rojo los campos que son obligatorios para sugerir un nuevo producto y el resto de los campos son opcionales. En la ilustración 50 se visualiza la sección correspondiente a los estándares que implementa el producto. En la lista de la derecha van a estar los estándares implementados (puede dejarse vacía). También se habilitó la posibilidad de agregar el grado de conformidad de los estándares implementados por el producto a través de un pop-up. Si el nivel es referente o certificado se pide que se ingrese la fecha de certificación.

#### 4.3.3 Gestión de productos

##### 4.3.3.1 Filtrado según arquitectura de referencia

Con el fin de informar al usuario la arquitectura utilizada por el sistema se agregó una funcionalidad en la pantalla de inicio que grafica dicha arquitectura (sección 2.2) y a su vez al seleccionar una capa permite filtrar los productos que la implementan.

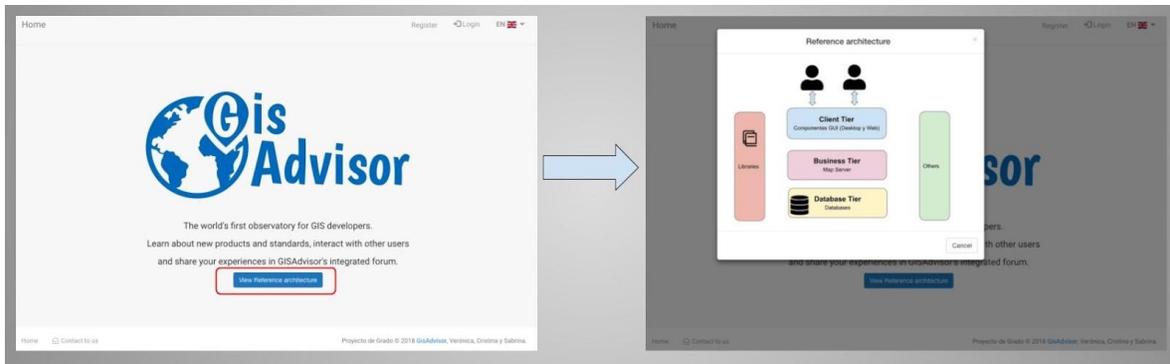


Ilustración 51: Presentación de la arquitectura de referencia del sistema

#### 4.3.3.2 Libros

Para cada producto del sistema existe una sección integrada con Google Books donde el usuario tiene la posibilidad de acceder a diferentes libros asociados al nombre del producto.

Para lograr esta integración, se utilizó la API de Google Books que permite buscar y acceder al contenido de los libros. La ilustración 52 muestra el ejemplo del uso de esta funcionalidad para el caso del producto GeoServer.

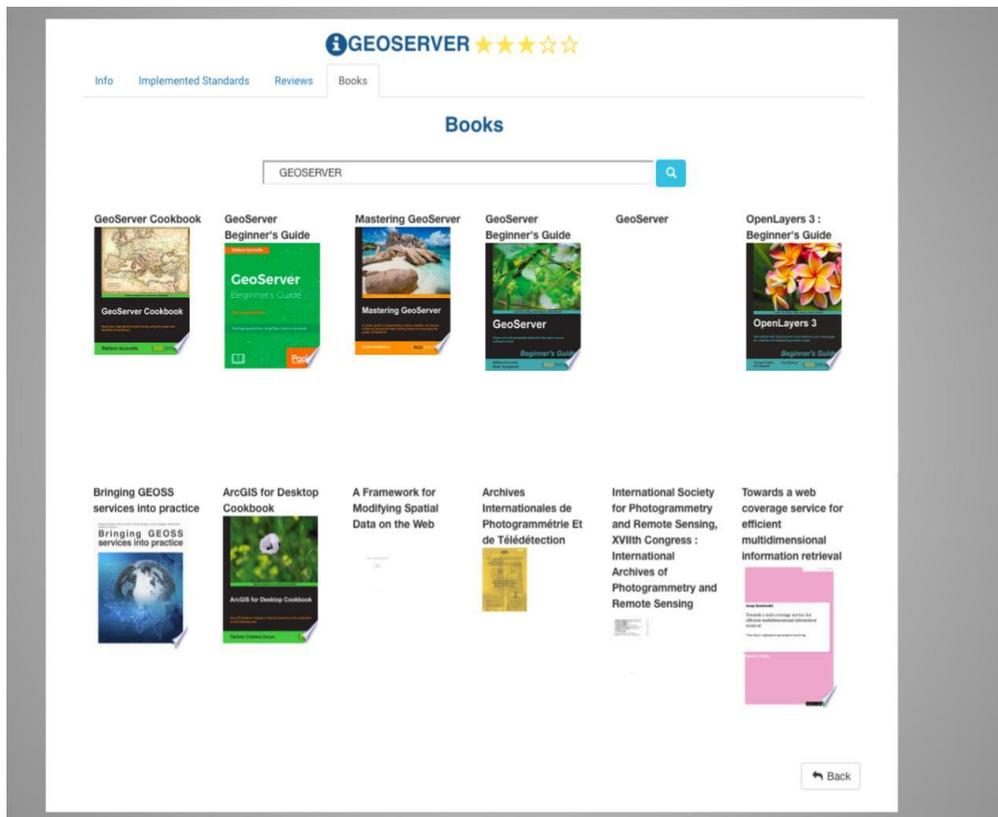


Ilustración 52: Funcionalidad de despliegue de libros recomendados para el producto seleccionado

### 4.3.3.3 Comparación

Otra de las funcionalidades que presenta el sistema es la comparación de productos, la misma permite elegir dos productos en base a la capa seleccionada. Comparando las diferentes propiedades así como los estándares implementados, mostrando en verde los que coinciden y en naranja los que no.

**Compare Products**

COMPARE PRODUCTS TO SEE WHICH IS MORE SUITABLE FOR YOU

pycsw - 2.0

VS

GI-cat - 12.0

[Compare](#)

	PYCSW	GI-CAT
	☆☆☆☆☆	☆☆☆☆☆
Free		
Open Source	No info	No info
Has Support	No info	No info
Web Site	<a href="http://pycsw.org/">http://pycsw.org/</a>	<a href="http://essi-lab.eu/gi-cat">http://essi-lab.eu/gi-cat</a>
Layer	Map Servers - NA	Map Servers - NA
Implemented Standards <input type="checkbox"/>	3	2
	<span style="color: green;">Catalogue Service - 2.0.2</span> <span style="color: orange;">Catalogue Service - 3.0</span> <span style="color: orange;">Catalogue Service - 3.0</span>	<span style="color: green;">Catalogue Service - 2.0.2</span> <span style="color: orange;">Catalogue Service - 1.0.0</span>

Ilustración 53: Vista de comparación de productos

Asimismo los usuarios administradores cuentan con la funcionalidad de seleccionar dos sugerencias de productos para ser comparadas entre sí.

Suggestions cana  EN

Filters

[Compare](#)

NAME	VERSION	TYPE	WEB SITE	
<input type="checkbox"/> GDAL/OGR	2.1	GUI Components	<a href="http://gdal.org">http://gdal.org</a>	✓ ✗
<input checked="" type="checkbox"/> Esri Geoportal Server	1.2.5	Map Servers	<a href="http://www.esri.com/software/arcgis/geoportal/index.html">http://www.esri.com/software/arcgis/geoportal/index.html</a>	✓ ✗
<input checked="" type="checkbox"/> XtraServer	4.11	Map Servers	<a href="http://www.interactive-instruments.de/en/xtraserver/">http://www.interactive-instruments.de/en/xtraserver/</a>	✓ ✗
<input type="checkbox"/> deegree webservises	3.4	Map Servers	<a href="http://www.lat-lon.de/">http://www.lat-lon.de/</a>	✓ ✗
<input type="checkbox"/> GDAL/OGR	2.1	GUI Components	<a href="http://gdal.org">http://gdal.org</a>	✓ ✗
<input type="checkbox"/> pycsw	2.0	Map Servers	<a href="http://pycsw.org/">http://pycsw.org/</a>	✓ ✗
<input type="checkbox"/> rassdaman	9.0	GUI Components	<a href="http://www.rasdaman.com">http://www.rasdaman.com</a>	✓ ✗

Ilustración 54: Vista de productos sugeridos con posibilidad de comparación

#### 4.3.4 Reportes

El sistema cuenta con una sección de reportes en la cual existen los siguientes tipos:

- **Usuarios por país:** Se muestra un mapa que expone la cantidad de usuarios registrados en cada país.
- **Ranking de Productos:** Se muestran los productos con mayor calificación en el sistema.
- **Cantidad de visitas por Producto:** Se despliegan los productos que existen en el sistema con sus respectivas cantidades de visitas realizadas por usuarios registrados.
- **Estándares implementados:** Se muestran los estándares más implementados por los productos existentes en el sistema.
- **Mis intereses:** Este reporte únicamente está disponible para usuarios que estén registrados en el sistema. Lo que se muestra en el mismo son la cantidad de visitas que el usuario realizó por producto. La idea de este reporte es generar un historial de visitas de productos por usuario con el fin de tener registradas las preferencias del mismo.

Un aspecto a destacar, en lo que respecta a la implementación de los reportes, es que para el reporte de usuarios por país se utilizó el componente Google Map Charts el cual utiliza la API de Google Maps para mostrar el mapa. Por otro lado, el resto de los reportes hacen uso del módulo FusionCharts el cual permite que se puedan desplegar la información en diferentes tipos de gráficas las cuales son elegidas por el usuario: Pie, Line, Vertical Bar o Horizontal Bar. Para dichos reportes también se agregó la posibilidad de elegir cuantos registros retorna la gráfica.

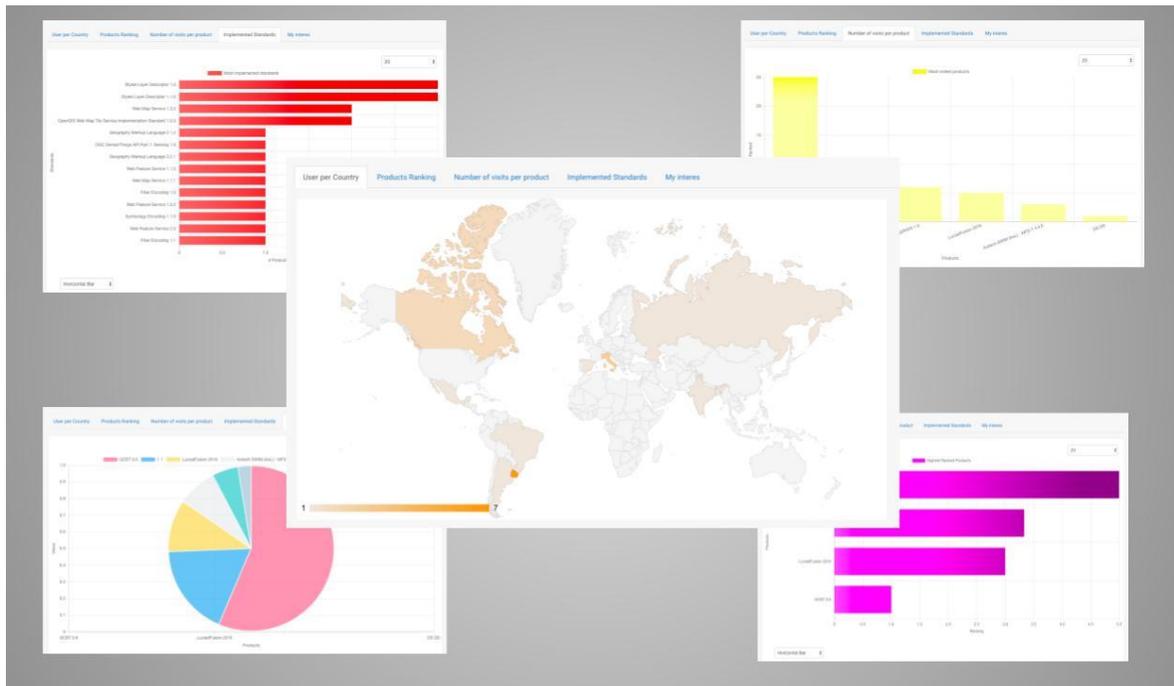


Ilustración 55: Reportes del sistema

#### 4.3.5 Notificaciones

El sistema cuenta con mecanismos de notificación a los usuarios. Estos mecanismos son: mediante emails y notificaciones push. El envío de mails se realiza en el back-end del sistema comunicándose con el servidor de mail directamente.

Los eventos que generan una notificación y los usuarios que las reciben son los siguientes:

- Administradores: sugerencias de nuevos productos por usuarios colaboradores
- Administradores: sugerencias de productos o estándares por parte del componente de Web Scraping
- Creador de un hilo del foro: nuevo comentario sobre el mismo
- Creador de comentario en un foro: nuevo voto para su comentario
- Usuario con determinado conocimiento: nueva validación del mismo

Para el envío de emails el sistema utiliza la API de mail de Java que se comunica con el servidor de mail y envía mensajes. En este caso se utiliza un servidor Gmail y se configura en el servidor la conexión al mismo, luego desde la aplicación se accede a esta configuración.

Type	
SMTP	
<< < 1-1 of 1 > >>	
<a href="#">Need Help?</a>	
<a href="#">Edit</a>	
Socket Binding:	mail-smtp
Use SSL?:	true
Username:	gisadvisor.info@gmail.com
Password:	*****

*Ilustración 56: Configuración del servidor de correos en el Wildfly*

Para las notificaciones push se necesita un componente externo para poder enviarlas. La forma de comunicación push a través de internet es en la que es inicializada por el servidor hacia el cliente, al revés de la más comúnmente utilizada pull. Esto presenta un desafío, ya que el front-end se ejecuta en los navegadores de cada usuario, haciendo difícil poder identificar cada navegador para enviar las notificaciones correspondientes. Para esta comunicación se utiliza el componente de mensajería de la plataforma firebase [56].

Para lograr la integración Firebase ofrece una librería para angular y una API para ser consumida por el back-end. En el front-end usando esta librería se realiza el siguiente flujo:

1. Al iniciar sesión un usuario por primera vez el navegador le solicita permiso para recibir notificaciones desde esta aplicación web
2. En caso de que los conceda la librería genera un identificador único para el navegador del cliente

3. El front-end envía este identificador al back-end para que lo asocie al usuario logueado y lo persista en la base de datos
4. El back-end luego envía mensajes a través de la API de Firebase indicando el identificador del cliente que hay que notificar.
5. Firebase recibe este mensaje y lo envía al navegador correspondiente, esto genera que aunque la aplicación no se esté ejecutando se mostrará una notificación en el mismo.

El front-end le ofrece una interfaz gráfica al usuario para que pueda acceder a todas las notificaciones que le fueron enviadas teniendo la posibilidad de marcarlas como leídas/no leídas así como también de eliminarlas. A su vez la barra de opciones del sistema ubicada en la parte superior de la pantalla muestra la cantidad de notificaciones que el usuario tiene sin leer.

#### 4.3.6 Actualización de datos

En esta sección se describe más en detalle alguno de los pasos del proceso de actualización de los datos.

##### 4.3.6.1 Componente Web Scraping

Este componente está compuesto de varias librerías (una por cada sitio web que se utiliza) y por un proyecto central que se encarga de cargar las librerías y ejecutar las funciones de scrapeo de cada una para luego realizar las llamadas a la API correspondientes. La implementación de la librería correspondiente a los estándares OGC se detallará a continuación:

Configuración de Parsehub: para esta configuración se parte de la página general de los estándares. Allí se recorre la lista de estándares y se obtiene el link a cada uno para luego dirigirse a cada una de sus páginas

**OGC® Standards and Supporting Documents**

OGC(R) standards are technical documents that detail interfaces or encodings. Software developers use these documents to build open interfaces and encodings into their products and services. These standards are the main "products" of the Open Geospatial Consortium and have been developed by the membership to address specific interoperability challenges. Ideally, when OGC standards are implemented in products or online services by two different software engineers working independently, the resulting components plug and play, that is, they work together without further debugging.

OGC maintains two tracks of standards: the Full standard track and the Community standard track. Each is summarized below.

Full standard track: the Full standard track is a consensus process to develop and approve a standard inside the OGC Technical Committee. In this track, a Standards Working Group is created and that group writes the standard and supports the approval process in the Technical Committee. There are two levels in this track:

1. OGC standard: this is the traditional OGC standard resulting in an implementable and testable standard or a conceptual model from which implementation standards can be developed; and
2. OGC standard with Compliance Suite: this is an OGC standard with a proven ability to be implemented. To reach this level, an OGC standard must have at least three reference implementations and there must be an OGC Compliance Program compliance test suite for all mandatory features of the standard.

- ▼ OGC® Standards
  - 3dP
  - ARML2.0
  - Cat: ebRIM App Profile: Earth Observation Products
  - Catalogue Service
  - CDB
  - CityGML
  - Coordinate Transformation
  - Filter Encoding
  - GML in JPEG 2000
  - GeoAPI
  - GeoPackage
  - GeoSciML
  - GeoSPARQL
  - Geography Markup Language
  - GeoRSS
  - Geospatial eXtensible Access Control Markup Language (GeoXACML)
  - Geospatial User Feedback (GUF)
  - GroundwaterML
  - I3S
  - IndoorGML
  - KML
  - LandInfra/InfraGML
  - Location Services (OpenLS)
  - Moving Features

Ilustración 57: Interfaz gráfica de la página general de los estándares del OGC [57]

Luego en la página específica del estándar se obtiene el primer párrafo luego del título “Overview” como la descripción. Para obtener esto de forma correcta se configura el selector de Parsehub con XPath, ya que el HTML de las páginas de los estándares no era uniforme:

```
//h2[contains(text(), "Overview")]/following::p[1] | //h2[contains(text(), "Overview")]/following::text() [1]
```

Por último para obtener información de todas las versiones también se configura una expresión XPath que obtiene la primera columna de la tabla que viene después del título “Downloads”

```
//h2[contains(text(), 'Downloads')]/following-sibling::table[1]/tbody/tr/td[count(//table/tbody/tr/th[.="Version"]/preceding-sibling::th)+1]
```

**OGC®**  
Making location count.

About ▾ Standards ▾ Innovation ▾ News & Events ▾ Membership ▾ Resources ▾

## 3D Portrayal Service

- 1) Overview
- 2) Downloads
- 3) Official Schemas
- 4) Related News

### 1) Overview

The 3D Portrayal Service Standard is a geospatial 3D content delivery implementation specification. It focuses on what is to be delivered in which manner to enable interoperable 3D portrayal.

It does not define or endorse particular content transmission formats, but specifies how geospatial 3D content is described, selected, and delivered. It does not prescribe how aforementioned content is to be organized and represented, but provides a framework to determine whether 3D content is interoperable at the content representation level. More details are available in Design of this standard.

### 2) Downloads

Version	Document Title (click to download)	Document #	Type
1.0	<a href="#">OGC 3D Portrayal Service 1.0</a>	15-001r4	IS

- OGC® Standards
  - 3dP
  - ARML2.0
  - Cat: ebRIM App Profile: Earth Observation Products
  - Catalogue Service
  - CDB
  - CityGML
  - Coordinate Transformation
  - Filter Encoding
  - GML in JPEG 2000
  - GeoAPI
  - GeoPackage
  - GeoSciML
  - GeoSPARQL
  - Geography Markup Language
  - GeoRSS
  - Geospatial eXtensible Access Control Markup Language (GeoXACML)
  - Geospatial User Feedback (GUF)
  - GroundwaterML
  - i3s
  - IndoorGML
  - KML

Ilustración 58: Interfaz gráfica del estándar 3D Portrayal Service del OGC [58]

Mapeo de datos al modelo del sistema: los datos obtenidos de Parsehub son una lista de estándares con los siguientes campos:

```
{
  "name": "3dP",
  "url": "http://www.opengeospatial.org/standards/3dp",
  "description": "The 3D Portrayal Service Standard is a geospatial 3D content delivery implementation",
  "versions": [
    {
      "version": "1.0",
      "type": "IS"
    }
  ]
}
```

Ilustración 59: Extracto de los datos obtenidos de Parsehub en el proyecto de estándares del OGC

El campo a traducir es el campo de las versiones. Para esto la librería busca solamente los de tipo IS o DIS. En caso de no existir notifica al administrador del error, al igual que si alguno de los otros campos viene vacío. La expresión regular que indica qué tipos aceptar se encuentra guardada en el archivo de configuración.

```
#Regex of standards versions types
typesUsedRegex = ^(d-is)$|^^(is)$
deprecatedTypeRegex = ^(d-is)$
```

*Ilustración 60:* Expresiones regulares utilizadas en el componente de Web Scraping para el mapeo de las versiones de los estándares

#### 4.4 Decisiones tomadas

Durante el desarrollo del proyecto se tomaron decisiones en cuanto a la implementación de las funcionalidades finales del prototipo como las tecnologías utilizadas para su desarrollo. Estas decisiones se explican en mayor detalle en las próximas secciones.

##### 4.4.1 Alcance de la solución

Como primera decisión dentro de la implementación de la solución se decide modificar levemente la arquitectura de referencia explicada en la sección 2.2. En un principio para categorizar las TIG se consideró utilizar únicamente las categorías: Map Server, GeoDBMS, Componentes GUI (Desktop y Web) y librerías, que son las categorías planteadas en la arquitectura de referencia. Pero luego de realizar algunos relevamientos de información se decidió agregar también la categorización GUI Mobile ya que esta categoría ha crecido en los últimos años por lo que resulta importante para la comunidad poder hacer un seguimiento de sus productos. Además se agrega la categoría “Otros” que permite contemplar casos especiales en las que un mismo producto contiene varias capas de la arquitectura de referencia. Por último viendo la categorización utilizada por la OSGEO surgió la necesidad de agregar la categoría “Catalogue Servers” para contemplar los proyectos del tipo “Metadata Catalogues”.

Otra decisión relevante que se tomó fue con respecto al alcance de la funcionalidad de la actualización de los datos del observatorio. Para poder cumplir los objetivos planteados, el sistema debe contar con un sistema de actualización de los datos robusto, ya que este componente de la solución es el más importante para los usuarios finales. La actualización totalmente automática resultó ambiciosa para el cronograma del proyecto. Esto es porque la misma implica análisis de una gran cantidad de datos de fuentes diferentes en tiempo real, y traducción de los mismos al modelo de datos del planteado para este sistema. Por eso se decidió, que el sistema de actualización funcione mediante notificaciones a los administradores del sistema en caso de detectar novedades y fundamentalmente se decidió realizar una plataforma colaborativa con diferentes tipos de usuario que colaboren en la actualización de los datos.

Además se decide incorporar funcionalidades de foro para que la comunidad de usuarios tenga un medio para comunicarse por temas relacionados con las TIG.

#### 4.4.2 Tecnologías utilizadas

Para la implementación de la solución se tomaron varias decisiones en cuanto a las tecnologías a utilizarse para el desarrollo de cada componente. A continuación se detallan las más relevantes.

##### 4.4.2.1 *Java EE*

Para la implementación del back-end se optó por la tecnología Java EE, las principales razones para esta elección fueron:

- Se buscó utilizar un framework abierto y gratuito.
- Es una tecnología que se puede correr en todos los sistemas operativos
- Se optó por Java EE Full porque brinda mayor modularidad a través de los EJB
- Además la versión EE full ofrece muchas APIs con diferentes propósitos como lo son la persistencia de los datos y los servicios web REST

##### 4.4.2.2 *Parsehub*

Parsehub es una herramienta para automatizar el Web Scraping de forma amigable a través de una interfaz gráfica. Nace como una extensión de Firefox pero ahora es una aplicación de escritorio. Tiene varias funcionalidades de gran utilidad:

- Permite seleccionar partes del sitio web haciendo click en la sección y luego sugiere partes similares para continuar con la selección.
- Permite configurar la selección de forma manual utilizando expresiones con CSS o expresiones XPath.
- Permite realizar selecciones relativas a otra selección ya hecha simplemente clickeando en lo ya seleccionado y luego clickeando en lo nuevo. También sugiere otros elementos parecidos para agregar.
- Exporta la información obtenida tanto en JSON como en CSV.
- Tiene una API que permite automatizar las ejecuciones y obtener los resultados de cada una.

standard_name	standard_url
3dP	http://www.opengeospatial.org/standards/3dp
ARML2.0	http://www.opengeospatial.org/standards/aml
Cat: ebRIM App Profile: Earth Observation Products	http://www.opengeospatial.org/standards/cat2eext4ebrim
Catalogue Service	http://www.opengeospatial.org/standards/cat

Ilustración 61: Interfaz gráfica de la herramienta Parsehub [48]

Esta herramienta cuenta con una versión gratuita limitada, que es la que se utiliza en el proyecto. Esta versión incluye:

- Velocidad limitada: 200 páginas en 40 minutos
- 200 páginas por ejecución
- 5 proyectos públicos
- Soporte limitado
- Retención de los datos por 14 días

Estas funcionalidades resultan suficientes para la solución, ya que permiten comunicación con el componente Web Scraping mediante la API. La cantidad de proyectos es mayor a la utilizada (estándares OGC, productos OGC, productos OSGeo y productos Wikipedia). Además la velocidad no resulta relevante ya que estos procesos se van a ejecutar semanalmente, así como tampoco resulta importante la retención de los datos, ya que los mismos son migrados a datos del sistema luego de la ejecución. La única limitante que aparece en esta versión es la cantidad de páginas en una misma ejecución, ya que el proyecto de los productos de la OGC visita muchas páginas más (en el orden de las 5000). Para cubrir este caso se utiliza la librería JSOUP de Java.

#### 4.4.2.3 Bootstrap[59]

Es un framework de front-end gratuito para lograr un desarrollo web más rápido y fácil dado que incluye plantillas de diseño basadas en HTML y CSS para tipografía, formularios, botones, tablas, navegación, modales, imagen y muchos otros lo cual hace que el desarrollo web sea más fácil, práctico y rápido.

Los motivos por los cuales se decidió incluir dicho framework en el proyecto son:

- **Menor tiempo:** se puede ahorrar mucho tiempo y esfuerzo utilizando las plantillas y clases de diseño predefinidas de Bootstrap y concentrarse en otros trabajos de desarrollo.
- **Responsive:** las plantillas con las que cuenta Bootstrap se adaptan al tamaño de la pantalla en la que está viendo la página, como escritorios, tabletas o teléfonos celulares.
- **Diseño uniforme:** todos los componentes de Bootstrap comparten las mismas plantillas y estilos de diseño a través de una biblioteca central, de modo que los diseños de las páginas web , que utilizan dichos componentes, son consistentes a lo largo de su desarrollo.
- **Fácil de usar:** Bootstrap es muy fácil de usar por lo que cualquier persona con conocimientos básicos de HTML y CSS puede comenzar a desarrollarlo con Bootstrap.
- **Compatibilidad:** es compatible con todos los navegadores” Mozilla Firefox, Google Chrome, Safari, Internet Explorer y Opera.
- **Código abierto:** es de uso gratuito.

#### 4.4.2.4 Wildfly [60]

Wildfly es un servidor de aplicaciones de código abierto de la familia de servidores JBoss. Es excepcionalmente rápido, liviano y es una implementación potente de las especificaciones de Java EE. Wildfly fue originalmente creado por JBoss pero actualmente se encuentra en desarrollo continuo por parte de Red Hat. Este servidor es bastante usado en el desarrollo Java actual, según una encuesta realizada en el año 2017 solo se encuentra en uso por debajo del servidor Tomcat. Este último, desarrollado por Apache, en su versión básica no implementa el perfil full de la plataforma Java EE, solamente implementa el perfil web, por más de que si se encuentran otros de esta familia que lo hacen como Tom EE el uso masivo de este servidor se da en su versión básica. [61]

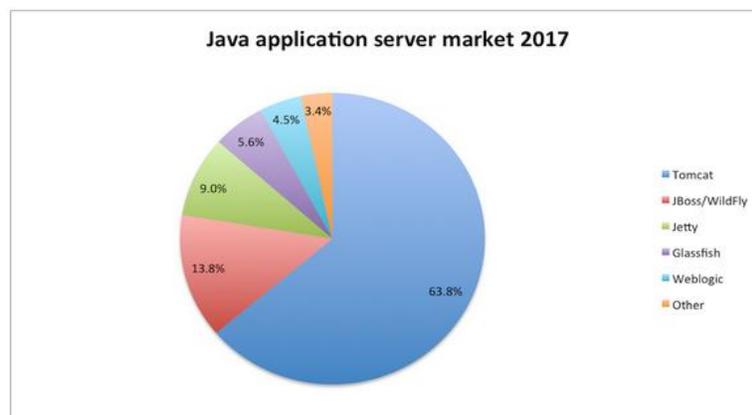


Ilustración 62: Uso de servidores de aplicaciones para Java en el 2017[62]

#### 4.4.2.5 *Angular 5*

Angular es un framework JavaScript, gratuito y Open Source, creado por Google y destinado a facilitar la creación de aplicaciones web modernas de tipo SPA (Single Page Application).

Una aplicación web SPA creada con Angular es una web de una sola página, en la cual la navegación entre secciones y páginas de la aplicación, así como la carga de datos, se realiza de manera dinámica, casi instantánea, asíncronamente haciendo llamadas al servidor (back-end con un API REST) y sin refrescar la página.

Otro aspecto a destacar es que Angular es desarrollada en TypeScript, el cual es un lenguaje de programación libre y de código abierto desarrollado por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipado estático y objetos basados en clases. Tiene una sintaxis intuitiva, los navegadores no convertirán TypeScript a JavaScript si no que TypeScript cuenta con un transpiler que convierte el código TypeScript a JavaScript y se puede elegir si usar ES5 [63] o ES6.

Si bien existen varios frameworks y librerías para agilizar el desarrollo web se decidió utilizar Angular dado que algunas de las integrantes del equipo habían utilizado AngularJS para algunas asignaturas en facultad por lo que contaban con cierta experiencia en el framework y resultó ser del agrado de los mismos por lo que se decidió seguir con la misma línea y utilizar Angular 5. De todas formas, a pesar de compartir el mismo nombre Angular 5 es un framework nuevo y posee conceptos y formas de trabajar diferentes a las de AngularJS lo cual implicaba una curva de aprendizaje bastante empinada, de igual manera se aceptó el desafío dado que Angular 5 presenta grandes cantidades de ventajas a la hora de realizar una aplicación web entre ellas:

- Está orientado a móviles.
- Al utilizar TypeScript el código es mantenible.
- Libre uso.
- El código es muy consistente lo cual implica que cualquier el código pueda ser entendido muy rápidamente.
- Utiliza componentes web, siendo estos es una porción de código que es posible reutilizar en otros proyectos de Angular sin apenas esfuerzo, lo que permite un desarrollo de aplicaciones mucho más ágil.
- Es respaldado por Google, esto implica que existe una comunidad grande de usuarios para soporte.

#### 4.4.2.6 *Firebase Cloud Messaging*

Esta herramienta desarrollada por Google se utiliza para las notificaciones push del sistema. Permite de forma muy sencilla implementar este tipo de comunicación desde el servidor hacia el cliente, haciendo de intermediario en el envío de los mensajes.

Dentro de sus principales características se destacan:

- Gratuito
- Ofrece API para la integración del back-end
- Ofrece librerías para la integración de diversos clientes: IOS, Android, JavaScript, C++ y Unity

- Cuenta con una consola web desde la cual se pueden enviar notificaciones de forma manual y realizar algunas otras configuraciones de las mismas.
- Permite enviar mensajes a usuarios específicos, a grupos de usuarios e incluso permite a los usuarios suscribirse a diferentes temas y luego enviar mensajes por tema.
- Vasta documentación y se utiliza por una gran cantidad de usuarios que brindan soporte en diferentes comunidades y foros.

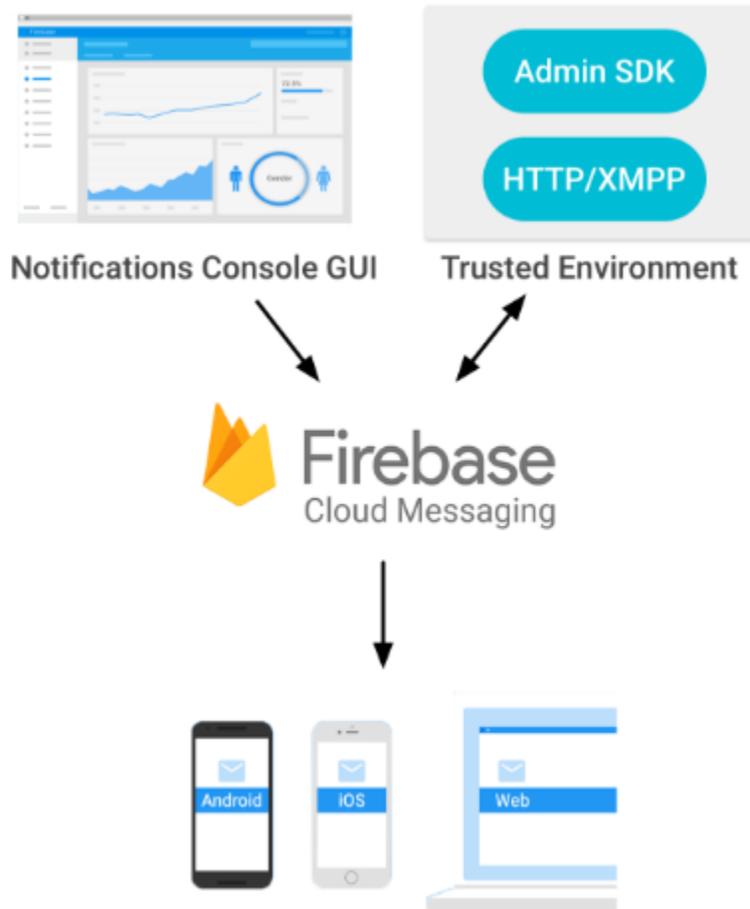


Ilustración 63: Diagrama de funcionamiento de Firebase [56]

## 5. Pruebas del sistema y gestión del proyecto

Esta sección describe el proceso de pruebas realizado así como la gestión del proyecto.

### 5.1 Pruebas del sistema

El proceso de pruebas se puede dividir en tres grandes secciones: pruebas del front-end, pruebas del back-end y carga y actualización de los datos. No es recomendable que las pruebas funcionales del sistema las realicen las mismas personas que se encargaron de la implementación. Aprovechando que el equipo de trabajo se dividió por componente, se designa a las desarrolladoras que no fueron encargadas del componente a probar, la realización de las pruebas funcionales del mismo.

#### 5.1.1 Pruebas del front-end

En el front-end se realizaron dos tipos de pruebas:

- Funcionales: se ejecutaron varios casos de prueba de forma manual para verificar que las funcionalidades estuvieran correctamente implementadas. Estas pruebas se ejecutaron luego de cada desarrollo de una funcionalidad nueva, los errores detectados se corrigieron antes de continuar con otras funcionalidades.
- De validación: estas pruebas fueron realizadas por personas que no pertenecen al equipo de desarrollo, como lo fueron los tutores del proyecto y otros alumnos avanzados de la carrera de ingeniería en computación. Se buscó probar la experiencia de usuario, verificar su amigabilidad y performance. Estas pruebas se realizaron una vez finalizado el desarrollo completo de la plataforma. Los errores fueron corregidos y se realizó una última validación con los usuarios luego de eso.

#### 5.1.2 Pruebas del back-end

Para las pruebas del back-end se utilizó la herramienta Postman[64]. Esta herramienta es un cliente de servicios web, que permite guardar colecciones de peticiones, automatizar sus ejecuciones y utilizando JavaScript escribir tests de las mismas.

Se realizaron pruebas después de cada liberación de una nueva funcionalidad. Luego se corrigieron los errores detectados en las mismas antes de continuar con otras funcionalidades nuevas. Algunas de estas pruebas están detalladas en el apéndice 2 de este informe.

Otras pruebas realizadas más a bajo nivel fue la revisión de código. En el momento de subir una nueva funcionalidad al repositorio GitLab, se realizaba un “merge request” a la rama principal del mismo. Esta solicitud debía ser aceptada por otra persona diferente a la que realizó el desarrollo y para aceptarlo se revisaba el código modificado y agregado. Algunos criterios a tener en cuenta en estas revisiones son: claridad en el código, presencia de comentarios, funciones cortas, entre otros.

#### 5.1.3 Carga y actualización de datos

Para terminar la validación de la solución se procede a realizar la carga definitiva de los datos. Para realizar esta carga se utilizan los componentes Twitter y Web Scraping. Además de la carga de los

datos también se prueban las funcionalidades de actualización. Para esto se realizaron las siguientes pruebas:

- Carga de estándares: se logran cargar automáticamente 50 de 58 estándares. Los 8 estándares faltantes no se cargan directamente debido a que el campo descripción no se pudo completar. De estos 8 hay 6 que no contienen información de la descripción del estándar en la página de la OGC, solamente 2 no son obtenidas directamente desde Parsehub debido a una irregularidad en el HTML del sitio. El sistema igualmente notifica correctamente a los administradores de estos 8 estándares para que el mismo los pueda completar.
- Actualización de estándares: para probar la actualización se borran estándares de la base de datos y se ejecuta el proceso de actualización. El mismo logró encontrar correctamente los estándares faltantes y notificar a los administradores del sistema de los mismos.
- Carga de productos OGC
- Carga de productos OSGeo
- Carga de productos Wikipedia

## 5.2 Gestión del proyecto

En esta sección se describe el proceso realizado en el desarrollo del proyecto identificando las diferentes etapas del mismo así como las desviaciones que surgieron. En primer lugar se plantea la planificación inicial con el fin de comparar la misma con dicho proceso.

### 5.2.1 Planificación inicial

El proyecto tuvo comienzo en el mes de Abril de 2017 en el marco de la asignatura “Proyecto de Grado”. En primer lugar se realizó una reunión con la tutora para plantear los objetivos del proyecto y el alcance del mismo. Además se proporcionó un informe acerca del relevamiento y caracterización de productos SIG de acuerdo al cumplimiento de estándares OGC realizado por estudiantes del “Módulo Taller-- Tecnologías de la Información Geográfica” con el objetivo de contar con una carga inicial de datos.

Luego de tener claro los objetivos del proyecto se planificaron las diferentes etapas y las duraciones de las mismas, se realizaron cronogramas por bimestre (los cuales se encuentran en el anexo del informe). En el siguiente diagrama Gantt se pueden observar las diferentes etapas que se planificaron inicialmente:

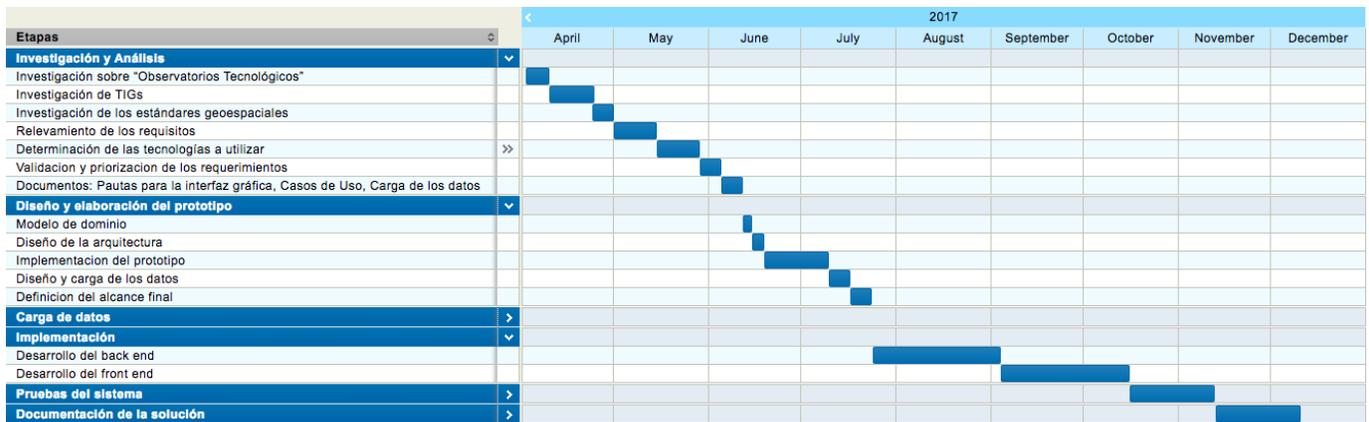


Ilustración 64: Cronograma inicial del desarrollo del proyecto

## 5.2.2 Etapas desarrolladas

### 5.2.2.1 Investigación y Análisis

En esta etapa se realizó una investigación sobre algunos conceptos clave para el entendimiento de los requerimientos del sistema. En primer lugar se investigó sobre los observatorios tecnológicos con el fin de comprender qué son y cuáles son sus principales funcionalidades, a su vez se buscaron ejemplos de observatorios existentes en la actualidad con el fin de tener algunas referencias para el proyecto. Asimismo se investigó acerca las principales características y funcionalidades de las TIG haciendo una distinción por capa y sobre los estándares geoespaciales existentes, profundizando en los más utilizados y sus diferentes implementaciones.

Posteriormente se realizó un relevamiento de requerimientos, los cuales fueron validados con los tutores, y se realizó una priorización de los mismos. Luego, se establecieron los diferentes casos de usos que presentan el sistema y las pautas para la interfaz gráfica.

Vale la pena destacar que para cada uno de los ítems mencionados anteriormente se creó un documento correspondiente, los mismos se encuentran en el anexo del informe.

Por último, se determinaron las tecnologías a utilizar en el desarrollo del sistema. Inicialmente, en conjunto con los tutores, se escogieron las tecnologías base, para luego realizar un estudio y comparación de las que se utilizarían en la actualización de datos, siendo esta la funcionalidad más importante del sistema. Cabe destacar que en todos los casos se optó por tecnologías de libre uso.

### 5.2.2.2 Diseño y elaboración del prototipo

En primera instancia, se realizó un modelo de dominio con el fin de poder representar los conceptos claves del sistema para luego realizar un diseño de la arquitectura de la aplicación que permita brindar una solución eficiente. Para lograr dicho diseño se realizaron diferentes reuniones con los tutores en donde se debatieron distintas posibilidades con el fin de mejorar la solución planteada.

También se seleccionó cuáles eran los datos sobre los que se basaría el sistema realizando un diseño sobre la carga de los mismos.

Una vez seleccionadas las tecnologías y contando con un diseño de la arquitectura se comenzó a realizar un prototipo para validar que la solución planteada era apropiada para el sistema, en particular para la actualización de datos.

Luego de tener la línea base de la arquitectura implementada en base a la priorización de requerimientos, se definió el alcance del proyecto definiendo las funcionalidades a desarrollar.

Por último, a principios de diciembre, una vez finalizado el prototipo y validado con los tutores se realizó una presentación con los mismos y algunos invitados de ellos en donde se mostró el avance del proyecto y las ideas sobre la arquitectura planteada.

### *5.2.2.3 Carga de datos*

El proceso de la carga de datos consta del diseño y desarrollo de los componentes de actualización explicados en la sección 4.2.3. Estos componentes fueron los encargados de realizar una carga inicial de todos los datos que luego fueron mejorados de forma manual desde la interfaz web. Este proceso se fue dando de forma paulatina durante el proyecto. Para esto, se comenzó con el caso de los estándares que eran una cantidad acotada y que se obtenían todos de la misma fuente. Luego de esta carga inicial se procedió a realizar su validación y correcciones correspondientes.

Luego comenzó el proceso de la carga de los productos. Para esto, primero se realiza únicamente la carga de los productos registrados en el OGC, como implementadores referentes de alguno de los estándares geoespaciales, para tener una cantidad más acotada de los mismos. Una vez realizada se procede a validar y corregir la misma. Luego se realiza la carga de los productos con implementaciones certificadas de igual manera que la anterior y por último se cargan los productos con implementaciones sin certificar ni referentes. De esa forma se termina la carga de datos de los productos registrados en OGC.

Por último se continúa con los sitios de OSGeo y de Wikipedia, en ese orden.

### *5.2.2.4 Implementación*

En esta etapa se desarrollaron las funcionalidades definidas. En primer lugar se desarrollaron las funcionalidades del back-end, reutilizando los componentes ya desarrollados para el prototipo inicial previamente validado: componente de web Scraping y componente encargado de la interacción con twitter para notificar sobre nuevas versiones de las herramientas.

Finalmente, se comenzó con el desarrollo de los componentes de front-end, los cuales demandaron más tiempo del planificado, dado que la curva de aprendizaje fue mayor a la que se estimó ya que las tecnologías utilizadas no eran de conocimiento de los integrantes del equipo. El componente de foro, el de login con redes sociales y el de notificaciones push fueron los que más tiempo consumieron a consecuencia de ser los más complejos.

### *5.2.2.5 Pruebas de sistema*

La finalización de las etapas de implementación y la carga de datos dio paso a esta nueva etapa, la última del proceso de construcción del sistema. En la misma se llevaron a cabo pruebas a nivel local.

Además se realizó una instalación del front-end en la plataforma Firebase con el fin de que el sistema pudiera ser accedido por usuarios sin necesidad de una instalación previa para lograr de esta forma pruebas a nivel de usuario.

### 5.2.2.6 Documentación de la solución

Finalmente se realizó la documentación del proyecto realizado con el fin de explicar la solución planteada. La documentación incluye un marco teórico para poder comprender el contexto del problema a solucionar, así como las tecnologías utilizadas y la arquitectura propuesta. Este marco teórico también incluye el resultado del objetivo del proyecto relacionado a la investigación de observatorios tecnológicos en general.

### 5.2.3 Desviación del proyecto

En un principio se planificó que el proyecto se llevará a cabo en 9 meses pero a lo largo del mismo fueron surgiendo algunas desviaciones, las cuales hicieron que se extendiera la duración planificada inicialmente. En el siguiente diagrama (dividido en dos imágenes para mejorar su visualización) se muestra el tiempo utilizado en las diferentes tareas y posteriormente se detallan las diferentes desviaciones.

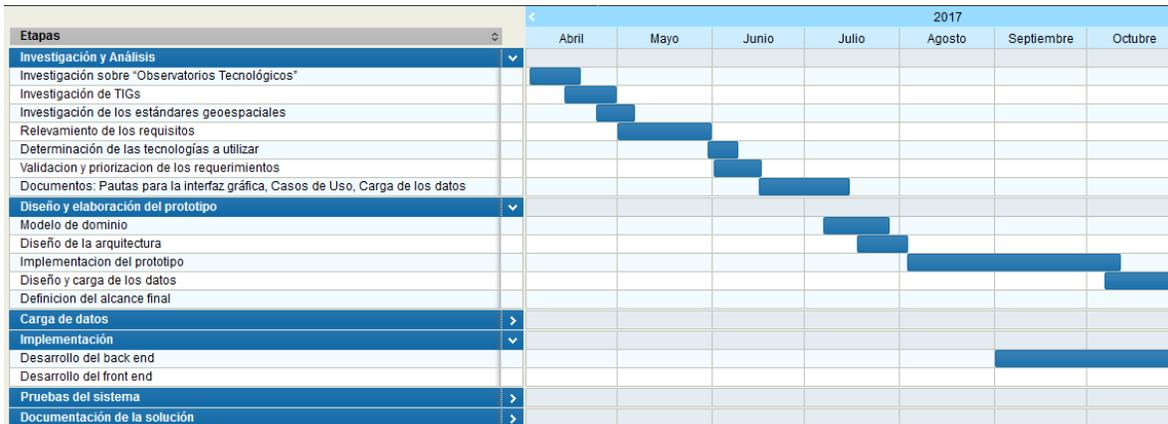


Ilustración 65: Cronograma final 1 del desarrollo del proyecto

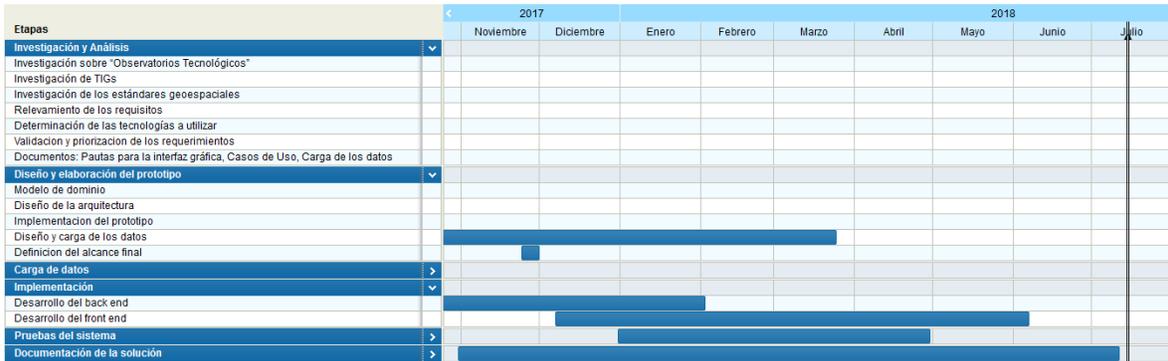


Ilustración 66: Cronograma final 2 del desarrollo del proyecto

### 5.2.3.1 *Desarrollo del front-end*

Como se mencionó anteriormente la tecnología utilizada para el desarrollo del front-end era desconocida por las integrantes del equipo, lo cual generó un gran desafío para éstas. A su vez provocó que el tiempo de aprendizaje de la misma fuese mucho mayor al estimado. Por otro lado, a medida que se fue desarrollando el front-end fueron surgiendo algunos arreglos y mejoras en los componentes de back-end, lo cual se vio reflejado en el tiempo planificado.

### 5.2.3.2 *Carga de datos*

La carga de datos fue el principal motivo por el que la duración del proyecto se vio retrasada, dado que a medida que se iba llevando a cabo la misma fueron surgiendo diferentes problemas.

- Actualización de la herramienta Parsehub: en el medio del proceso de carga se actualizó la herramienta utilizada, cambió su funcionamiento y generó retrasos al seguir configurando los diferentes proyectos.
- Cambio de la página de OSGeo: la página de la OSGeo fue modificada durante el proceso. Los cambios no fueron solamente en los estilos de la misma sino también estructurales, por lo que se tuvo que rehacer el proyecto de Parsehub para obtener los datos.
- Estándares implementados del OGC: en la información brindada por esta página aparecen los estándares implementados de los productos. Estos aparecen con nombres diferentes a los que utiliza la propia página en la sección de los estándares. El mapeo entre estos nombres y los de los estándares fue trabajoso y retrasó bastante el cronograma, especialmente debido al volumen de datos que se manejan (más de 6000 estándares implementados).
- Productos iguales en fuentes distintas: al utilizar varias fuentes para la carga de los productos, surge el problema de reconocer si se estaba cargando dos veces el mismo producto.

### 5.2.4 *Herramientas utilizadas*

Para llevar a cabo la gestión del proyecto se utilizaron varias herramientas con el fin de lograr una buena organización entre los integrantes del equipo, a continuación se describirán las más relevantes:

Para la gestión de las tareas pendientes se utilizó Trello. Trello es un software de administración de proyectos con interfaz web el cual está disponible tanto para Android como para iOS utilizado para organizar proyectos.

Trello permite crear distintos paneles de control (llamados tableros) que hacen referencia al proyecto donde se está trabajando. En ellos, se pueden configurar distintas columnas, en este caso se crearon las siguientes columnas para las cuales se fueron asignando diferentes tipos de tareas:

- **To Do:** tareas pendientes.
- **In Progress:** tareas que se estaban siendo desarrolladas por algún integrante del equipo.
- **Testing:** tareas que estaban siendo testeadas.
- **Done:** tareas que se habían desarrollado y testeados.

- **Bug:** tareas en las que se habían encontrado un error.

Cada una fue asignada a un miembro del equipo y clasificada en “front-end” y “back-end”. A su vez se permite agregar tiempos límites a las tareas lo cual permitió que se pudieran priorizar las mismas. Por lo tanto, dicha herramienta fue de mucha utilidad para el equipo dado que permitió una buena organización a la hora de implementar los requerimientos del sistema.

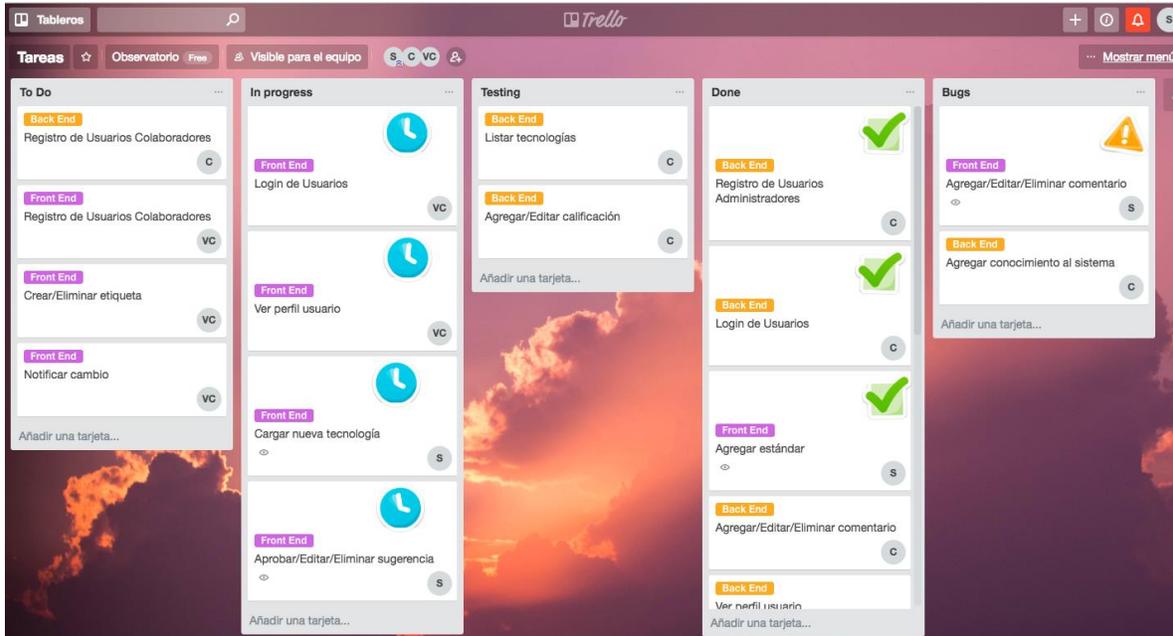


Ilustración 67: Tablero del proyecto GISAdvisor

Además se creó un grupo en WhatsApp con las integrantes del equipo para mejorar la organización, ya sea para agendar las diferentes reuniones así como para consultar dudas o debatir diferentes soluciones para algunos problemas. Dicha aplicación móvil fue de gran utilidad para el equipo ya que fue el principal medio de comunicación y al ser mensajes instantáneos, permitió hacer la misma más fluida.

Para la gestión de la documentación se utilizó la herramienta Google Drive. Esta herramienta se utilizó principalmente para la elaboración de la documentación del sistema así como para compartir los mismos con los tutores con el fin de tener una devolución de estos. Al ser una herramienta online permite que todas las integrantes del equipo puedan trabajar en paralelo y así agilizar la elaboración de los documentos.

Por último para la gestión del código de la aplicación se utilizó GitLab. Éste es un gestor de repositorios en el cual se crearon cuatro repositorios, uno para el back-end, uno para el front-end y uno para cada componente de actualización. En los mismos, se fue subiendo el código a medida que se iba desarrollado, lo cual permitió que todas las integrantes pudieran trabajar sobre el mismo y tener copias de seguridad de este último. Dicha herramienta también permite tener un versionado del código y de esta forma ir logrando diferentes versiones del prototipo.

## 6. Conclusiones y trabajo a futuro

### 6.1 Conclusiones

El objetivo general del proyecto fue el diseño y la implementación de un observatorio de TIG con el fin de centralizar información, para facilitarles la búsqueda y comparación de las mismas a las personas que las utilizan, ya sean desarrolladores o simplemente usuarios TIG en general.

Se logró cumplir este objetivo a través de una investigación de otros observatorios tecnológicos que permitió definir los requisitos iniciales del sistema a implementar.

El proyecto también contó con investigación de las TIG existentes en la actualidad. Se realizó un primer relevamiento para conocer las diferentes fuentes y las principales características de la información a manejar. Así como también se investigó la arquitectura de referencia que terminó determinando la categorización a utilizar para las TIG. Esta información permitió concluir la necesidad de desarrollar un sistema colaborativo y terminar de definir los requisitos finales del sistema. Luego en la presentación de avance del proyecto surgieron nuevos requerimientos como fueron el agregado de libros sugeridos en cada producto del sistema.

Para la realización del prototipo se desarrollaron tres componentes de forma tal de poder dividir las responsabilidades. Uno de los componentes es el encargado de la carga inicial y la actualización de parte de la información del sistema utilizando Web Scraping y consultando la red social Twitter. Por otro lado, se implementaron un componente de front-end y uno de back-end conteniendo toda la lógica de negocio y el almacenamiento de los datos del sistema.

En cuanto a tecnologías utilizadas para la implementación se concluye que la técnica de Web Scraping, si bien es sencilla de utilizar, no es fácil realizar una solución escalable utilizando la misma debido al dinamismo de los sitios web. Para esto se utilizó la librería de Java JSOUP, la cual cuenta con buena documentación y soporte de la comunidad de usuarios. Además se utilizó la herramienta Parsehub para facilitar el mantenimiento del sistema. Esta última tiene algunas limitaciones para el uso gratuito, pero resulta muy amigable y completa para la integración. Por último la lógica se dividió en diferentes librerías según la fuente permitiendo modificar o agregar nuevas fuentes sin generar un gran impacto en el sistema.

Por otra parte se utilizó Angular 5 para el desarrollo del front-end. Esta era desconocida para los integrantes del proyecto pero se destaca como una tecnología muy potente para la realización de aplicaciones web. También cuenta con una gran comunidad de usuarios que colaboran en la resolución de problemas y existe vasta documentación sobre la misma. El componente de front-end soporta una comunidad de usuarios internacional gracias a los dos idiomas disponibles en el mismo: inglés y español.

El diseño de una solución escalable utilizando Web Scraping y la utilización de Angular 5 presentaron los desafíos más grandes para la implementación. En el caso de Angular esto fue debido al desconocimiento y a la gran diferencia que existe entre ésta y otras tecnologías similares que se conocían.

El sistema contó con testing funcional total desarrollado durante y luego de finalizada la etapa de desarrollo. Además luego de finalizada la misma se pasó a realizar la carga inicial de los datos del sistema. Esto también fue un gran desafío debido a la calidad de los datos que se manejaron. Se cargaron datos de forma automática, algunos de ellos fueron mejorados a través del sistema mismo por parte de administradores. También se realizaron pruebas de la actualización de los mismos mediante el borrado de algunos y el cambio en las versiones de otros.

## 6.2 Trabajo a futuro

Si bien, el sistema implementa las funcionalidades definidas en el alcance, se consideraron una serie de modificaciones para una eventual mejora del producto las cuales se describirán a continuación:

- En lo que respecta a la creación/sugerencia de nuevos productos, se podría mejorar la experiencia de usuario, sugiriendo los estándares implementados por el producto que se está creando/sugiriendo basándose en el tipo seleccionado, es decir, la capa de la arquitectura a la que pertenece.
- Agrupación de productos: cuando se realizó la carga de datos automática utilizando la página del OGC, varios productos listados como diferentes en realidad son los mismos pero divididos por funcionalidad. Para facilitar el uso se podría agregar la posibilidad para los administradores de seleccionar estos productos y agruparlos como un único producto.
- Utilizando el reporte “Mis intereses” en donde se muestran cuáles son los intereses de cada usuario, se puede sugerir otros productos que estén relacionados con los que el usuario visitó.
- Mejoras en la solución de Twitter: se podría agregar procesamiento del lenguaje natural a la lógica de la aplicación para reconocer más fácilmente cuándo un tuit está anunciando el lanzamiento de una nueva versión.
- Mejoras a la solución de Web Scraping: incorporar Web Scraping semántico. Esto no fue posible debido a la falta de sitios web que ofrecieran su versión en RDF, pero en un futuro podría incorporarse.

## 7. Referencias

- [1] P. Blišťan, Ľ. Kovanič, y M. Kovaničová, «The Importance of Geographic Information Systems Education at Universities in the Process of Building a European Knowledge-based Society», *Procedia - Soc. Behav. Sci.*, vol. 191, pp. 2458-2462, jun. 2015.
- [2] «Certified and Implementing Products | OGC». [En línea]. Disponible en: <http://www.opengeospatial.org/resource/products>. [Accedido: 20-jun-2018].
- [3] L. González, G. Llambías, y P. Pazos, «6th International Policy and Research Conference on Social Security», 2010.
- [4] M. A. Bernabé Poveda, C. M. López Vázquez, y Ó. Abarca, *Fundamentos de las infraestructuras de datos espaciales*. Madrid: UPM Press, 2012.
- [5] Juan E. Gutiérrez Palacios, *Los Sistemas de Información Geográfica*. .
- [6] «information system | Definition, Examples, & Facts | Britannica.com». [En línea]. Disponible en: <https://www.britannica.com/topic/information-system>. [Accedido: 06-mar-2018].
- [7] N. J. Gupta, «Fundamentals of Geographical Information System (GIS), Map Sources, and Digital Map Preparation», en *Geospatial Technologies and Climate Change*, J. Sundaresan, K. M. Santosh, A. Déri, R. Roggema, y R. Singh, Eds. Cham: Springer International Publishing, 2014, pp. 247-260.
- [8] J. L. Awange y J. B. Kyalo Kiema, «Fundamentals of GIS», en *Environmental Geoinformatics*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 191-200.
- [9] F. A. Sarría, «Sistemas de Información Geográfica», p. 239.
- [10] Rosmarie Sommer, Ruth Wenger, y Susanne Wymann von Dach, «Tecnologías de Información Geográfica para el manejo de recursos naturales», *inforesources*, 2007. [En línea]. Disponible en: [https://boris.unibe.ch/71064/3/focus07\\_3\\_s.pdf](https://boris.unibe.ch/71064/3/focus07_3_s.pdf). [Accedido: 09-jul-2018].
- [11] «Global Geographic Information System (GIS) Market | P&S Market Research». [En línea]. Disponible en: <https://www.psmarketresearch.com/market-analysis/geographic-information-system-market>. [Accedido: 06-mar-2018].
- [12] «About OSGeo», *OSGeo*. [En línea]. Disponible en: <https://www.osgeo.org/about/>. [Accedido: 05-mar-2018].
- [13] «W3C Semantic Web Activity Homepage». [En línea]. Disponible en: <https://www.w3.org/2001/sw/>. [Accedido: 22-mar-2018].
- [14] «About us». [En línea]. Disponible en: <https://www.iso.org/about-us.html>. [Accedido: 18-oct-2017].
- [15] «Welcome to the OGC | OGC». [En línea]. Disponible en: <http://www.opengeospatial.org/>. [Accedido: 18-oct-2017].
- [16] M. G. F. DE, «GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN TRABAJO FIN DE GRADO», p. 87.
- [17] NATALIA CALLE y MARTÍN STEGLICH, «Relevamiento y caracterización de productos GIS de acuerdo al cumplimiento de estándares OGC», Facultad de Ingeniería, 2016.

- [18] «Technology Research | Gartner Inc.» [En línea]. Disponible en: <https://www.gartner.com/technology/home.jsp>. [Accedido: 18-oct-2017].
- [19] «Forrester». [En línea]. Disponible en: <https://www.forrester.com/home/0,6092,1-0,FF.html>. [Accedido: 18-oct-2017].
- [20] «Research at Google». [En línea]. Disponible en: <https://research.google.com/>. [Accedido: 18-oct-2017].
- [21] «Observatorio Tecnológico», *Agesic*. [En línea]. Disponible en: <https://www.agesic.gub.uy/innovaportal/v/3704/1/agesic/observatorio-tecnologico.html?idPadre=4099>.
- [22] «OVTT | Proyecto». [En línea]. Disponible en: <http://www.ovtt.org/proyecto>.
- [23] «About EITO». [En línea]. Disponible en: [https://www.eito.com/epages/63182014.sf/en\\_GB/?ObjectPath=/Shops/63182014/Categorises/AboutUs/About\\_EITO](https://www.eito.com/epages/63182014.sf/en_GB/?ObjectPath=/Shops/63182014/Categorises/AboutUs/About_EITO). [Accedido: 18-oct-2017].
- [24] «Digital Observatory for Protected Areas». [En línea]. Disponible en: <http://dopa.jrc.ec.europa.eu/en>. [Accedido: 07-mar-2018].
- [25] «Observatorio Tecnológico», *ITU*. [En línea]. Disponible en: <http://www.itu.int:80/es/ITU-T/techwatch/Pages/default.aspx>.
- [26] «Slashdot: News for nerds, stuff that matters». [En línea]. Disponible en: <https://slashdot.org>. [Accedido: 20-jun-2018].
- [27] «Stack Overflow - Where Developers Learn, Share, & Build Careers», *Stack Overflow*. [En línea]. Disponible en: <https://stackoverflow.com/>. [Accedido: 20-jun-2018].
- [28] «CPUBoss», *CPUBoss*. [En línea]. Disponible en: <http://cpuboss.com/>. [Accedido: 20-jun-2018].
- [29] «Versus | Compara cualquier cosa, desde teléfonos móviles hasta ciudades», *VERSUS*. [En línea]. Disponible en: <https://versus.com/es>. [Accedido: 20-jun-2018].
- [30] «Delivery de Comida Online - Elegí, Pedí y Ya», *PedidosYa*. [En línea]. Disponible en: <https://www.pedidosya.com.uy>. [Accedido: 20-jun-2018].
- [31] «Uber - Get a Ride Near You - Earn Money by Driving | Uber». [En línea]. Disponible en: <https://www.uber.com/en-UY/>. [Accedido: 20-jun-2018].
- [32] G. Boeing y P. Waddell, «New Insights into Rental Housing Markets across the United States: Web Scraping and Analyzing Craigslist Rental Listings», *J. Plan. Educ. Res.*, vol. 37, n.º 4, pp. 457-476, dic. 2017.
- [33] «Web Scraping Vs Web Crawling -», *Pro Web Scraping*, 16-oct-2015. .
- [34] «The Web Robots Pages». [En línea]. Disponible en: <http://www.robotstxt.org/robotstxt.html>. [Accedido: 20-jun-2018].
- [35] «Web Scraping | Columbia University Mailman School of Public Health». [En línea]. Disponible en: <https://www.mailman.columbia.edu/research/population-health-methods/web-scraping>. [Accedido: 20-jun-2018].

- [36] «Web Scraping: Best Practices to Follow | | PromptCloud». [En línea]. Disponible en: <https://www.promptcloud.com/blog/web-scraping-best-practices>. [Accedido: 20-jun-2018].
- [37] «What is FreeFormat | GooSeeker». [En línea]. Disponible en: <http://www.gooseeker.com/en/node/knowledgebase/freeformat>. [Accedido: 22-mar-2018].
- [38] «Oracle Developers Portal». [En línea]. Disponible en: <https://developer.oracle.com/>. [Accedido: 22-mar-2018].
- [39] «Differences between Java EE and Java SE - Your First Cup: An Introduction to the Java EE Platform». [En línea]. Disponible en: <https://docs.oracle.com/javaee/6/firstcup/doc/gkhoy.html>. [Accedido: 21-mar-2018].
- [40] Eric Jendrock, Ricardo Cervera-Navarro, Ian Evans, Kim Haase, y William Markito, «The Java EE 7 Tutorial», Setiembre 2014.
- [41] «1.7 Java EE 7 APIs - Java Platform, Enterprise Edition: The Java EE Tutorial (Release 7)». [En línea]. Disponible en: <https://docs.oracle.com/javaee/7/tutorial/overview007.htm>. [Accedido: 20-jun-2018].
- [42] «Introducing the Java EE Web Profile - JAXenter». [En línea]. Disponible en: <https://jaxenter.com/introducing-the-java-ee-web-profile-103275.html>. [Accedido: 20-jun-2018].
- [43] «Hibernate. Everything data. - Hibernate». [En línea]. Disponible en: <http://hibernate.org/>. [Accedido: 20-jun-2018].
- [44] «Firebase». [En línea]. Disponible en: <https://firebase.google.com/>. [Accedido: 09-jul-2018].
- [45] «What Are RESTful Web Services? - The Java EE 6 Tutorial». [En línea]. Disponible en: <https://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html>. [Accedido: 20-jun-2018].
- [46] «Twitter4J - A Java library for the Twitter API». [En línea]. Disponible en: <http://twitter4j.org/en/index.html>. [Accedido: 18-jul-2018].
- [47] «jsoup Java HTML Parser, with best of DOM, CSS, and jquery». [En línea]. Disponible en: <https://jsoup.org/>. [Accedido: 20-jun-2018].
- [48] «Free web scraping - Download the most powerful web scraper | ParseHub». [En línea]. Disponible en: <https://www.parsehub.com/>. [Accedido: 20-jun-2018].
- [49] «List of geographic information systems software», *Wikipedia*. 29-sep-2017.
- [50] «Projects Archive», *OSGeo*. [En línea]. Disponible en: <https://www.osgeo.org/projects/>. [Accedido: 20-jun-2018].
- [51] J. Reschke <julian.reschke@greenbytes.de>, «The “Basic” HTTP Authentication Scheme». [En línea]. Disponible en: <https://tools.ietf.org/html/rfc7617>. [Accedido: 20-jun-2018].
- [52] J. Bradley, N. Sakimura, y M. Jones, «JSON Web Token (JWT)». [En línea]. Disponible en: <https://tools.ietf.org/html/rfc7519>. [Accedido: 20-jun-2018].
- [53] J. Bradley, N. Sakimura, y M. Jones, «JSON Web Signature (JWS)». [En línea]. Disponible en: <https://tools.ietf.org/html/rfc7515>. [Accedido: 20-jun-2018].
- [54] «JSON Web Tokens - jwt.io». [En línea]. Disponible en: <https://jwt.io/>. [Accedido: 20-jun-2018].

- [55] «PrimeNG». [En línea]. Disponible en: <https://www.primefaces.org/primeng/#/>. [Accedido: 20-jun-2018].
- [56] «Firebase Cloud Messaging», *Firebase*. [En línea]. Disponible en: <https://firebase.google.com/docs/cloud-messaging/?hl=es-419>. [Accedido: 09-jul-2018].
- [57] «OGC® Standards and Supporting Documents | OGC». [En línea]. Disponible en: <http://www.opengeospatial.org/standards/>. [Accedido: 20-jun-2018].
- [58] «3D Portrayal Service | OGC». [En línea]. Disponible en: <http://www.opengeospatial.org/standards/3dp>. [Accedido: 20-jun-2018].
- [59] «Bootstrap · The most popular HTML, CSS, and JS library in the world.» [En línea]. Disponible en: <https://getbootstrap.com/>. [Accedido: 20-jun-2018].
- [60] «WildFly Homepage · WildFly». [En línea]. Disponible en: <http://wildfly.org/>. [Accedido: 22-mar-2018].
- [61] «Getting Started Guide - WildFly 10 - Project Documentation Editor». [En línea]. Disponible en: <https://docs.jboss.org/author/display/WFLY10/Getting+Started+Guide>. [Accedido: 22-mar-2018].
- [62] «Most popular Java application servers: 2017 edition – Plumber». [En línea]. Disponible en: <https://plumbr.io/blog/java/most-popular-java-application-servers-2017-edition>. [Accedido: 22-mar-2018].
- [63] «TypeScript - JavaScript that scales.» [En línea]. Disponible en: <https://www.typescriptlang.org/>. [Accedido: 20-jun-2018].
- [64] «Postman | API Development Environment». [En línea]. Disponible en: <https://www.getpostman.com/>. [Accedido: 20-jun-2018].

## Apéndice 1: Documento de requisitos

**Fecha:** 30/05/2017

**Versión:** 2.0

**Responsables:** Verónica Cardozo, Sabrina Sosa, Cristina Vilaboa

### Sistema a construir

El proyecto consiste en la implementación de una aplicación web como observatorio para Tecnologías de Información Geográfica (TIG), las cuales denominaremos como “productos”. La misma contará con facilidades para la búsqueda y comparación entre las diferentes tecnologías que implementan estándares geoespaciales.

La aplicación tiene dos desafíos importantes: la amigabilidad de la interfaz de usuario y la recolección de los datos para la misma. Para lograr que la información se mantenga actualizada se propone habilitar a todos los usuarios a agregar contenido que luego sería controlado por los administradores del sistema.

### Usuarios

Este producto está destinado a la comunidad de desarrolladores TIG y a los usuarios TIG en general. Estos usuarios tienen una cantidad de tiempo importante en la realización de su trabajo, dedicado a la investigación de las tecnologías disponibles para hacerlo, así como si son pagas, gratuitas, si implementan determinados estándares o no. Esta información no está centralizada, lo que hace que este proceso sea aún más trabajoso.

Los usuarios son personas técnicas con conocimientos en el área de los Sistemas de Información Geográfica (SIG), y por ende tienen buenos conocimientos en computación. Igualmente el sistema deberá ser amigable y deberá contar con buena documentación de usuario.

Se diferencian los siguientes tipos de usuario en la aplicación:

- **Usuario administrador:** Son un conjunto reducido de usuarios autenticados que son los encargados de la gestión de los estándares, los productos y tienen la función de moderadores en el foro.
- **Usuario colaborador:** Se autentican en el sistema y pueden sugerir nuevos productos al sistema, calificar diferentes productos y participar en el foro.
- **Usuario anónimo:** son los que no iniciaron sesión en el sistema. Pueden consultar la información disponible de estándares y productos.

### Requisitos funcionales

1. **Login de Usuarios:** ingresando el mail y la contraseña utilizados para registrarse, el usuario puede ingresar al sitio.
2. **Registro de Usuarios Colaboradores:** ingresan mail, nombre, país, sexo, fecha de nacimiento y contraseña.

3. **Registro de Usuarios Administradores:** los administradores del sistema pueden crear nuevos usuarios administradores ingresando: mail, nombre, país, sexo, fecha de nacimiento y contraseña de los mismos.
4. **Agregar conocimiento al usuario:** los usuarios podrán agregar un conocimiento de los disponibles en el sistema a su perfil.
5. **Agregar/Editar/Eliminar comentario:** el usuario puede agregar comentarios relacionados a alguna de los productos. Los colaboradores pueden editar o eliminar sus propios comentarios. Los administradores pueden editar o eliminar cualquier comentario.
6. **Agregar/Editar calificación:** el usuario puede agregar/editar una calificación a alguno de los productos.
7. **Filtrar productos:** el usuario puede filtrar los diferentes productos según los siguientes criterios: nombre, estándares implementados, pago/gratuito, clasificación según arquitectura de referencia u open source o no.
8. **Listar productos:** se lista los productos según los siguientes criterios: mejor calificadas, más recientes, alfabético por el nombre o más visitadas.
9. **Comparar productos:** el usuario selecciona dos productos pertenecientes a la misma capa y se comparan según sus características principales.
10. **Sugerir producto:** el usuario colaborador puede crear nuevos productos que serán revisados por algún administrador antes de ser dados de alta en el sistema.
11. **Aprobar/Editar/Eliminar sugerencia:** los usuarios administradores pueden aprobar, editar o eliminar cualquiera de las sugerencias realizadas por los colaboradores.
12. **Cargar nuevo producto:** los administradores pueden dar de alta a un nuevo producto en el sistema ingresando: nombre, versión, fecha versión, descripción, imagen, link a la página oficial, estándares implementados, pago/gratuito, open source o no, soporte y clasificación según arquitectura de referencia.
13. **Agregar estándar:** los administradores pueden dar de alta un nuevo estándar ingresando: nombre, descripción, versión y página web.
14. **Ver perfil usuario:** los usuarios pueden acceder al perfil de los otros usuarios existentes en el sistema.
15. **Validar conocimiento usuario:** los usuarios podrán validar conocimientos de otro usuario perteneciente al sistema.
16. **Agregar tema de discusión al Foro:** los usuarios podrán agregar nuevos temas de discusión al foro ingresando un título para el mismo.
17. **Agregar comentario a tema de discusión del Foro:** los usuarios podrán agregar comentarios a los temas de discusión existentes del foro.
18. **Eliminar comentario/tema de discusión del foro:** los administradores podrán eliminar tanto comentarios como temas de discusión del foro.
19. **Cerrar tema de discusión del foro:** los administradores podrán cerrar un tema de discusión del foro deshabilitando la posibilidad de que se agreguen más comentarios al mismo pero dejándolo visible para los usuarios.

20. **Notificar cambio:** los sistemas externos encargados de la actualización de la información notifican al sistema en caso de detectar un nuevo estándar o producto o una nueva versión de los mismos.

### Requisitos no funcionales

1. **Carga de datos:** los datos van a tener una carga inicial por parte de las desarrolladoras y luego contará con varios sistemas para su actualización:
  - a. Contará con un sistema externo encargado de revisar si hay nuevas liberaciones de productos o estándares en la web mediante Web Scraping.
  - b. Contará con otro sistema externo encargado de revisar las cuentas oficiales de twitter de los productos revisando si se encuentran nuevas versiones de los mismos disponibles.
  - c. Por último será una plataforma colaborativa por lo que los usuarios del sistema tendrán la posibilidad de agregar datos al mismo que serán luego aprobados por los administradores del sistema.
2. **Performance:** el sistema debe soportar varios usuarios de forma concurrente haciendo consultas a la base de datos del sistema.
3. **Soporte Web:** se debe garantizar una buena experiencia de usuario (usabilidad y estética) para esto se debe seguir el standard W3C. Además el sitio debe funcionar de forma correcta en los browsers más usados, como ser Google Chrome (versión 6 en adelante), Mozilla Firefox (versión 4 en adelante), Internet Explorer (versión 10 en adelante), Opera (versión 10 en adelante) y Safari (versión 8 en adelante).
4. **Soporte para dispositivos móviles:** el sistema debe garantizar una buena experiencia de usuario en los dispositivos móviles, teniendo un responsive design.

### Restricciones

1. **Prototipo para evaluar arquitectura:** al final del segundo bimestre se espera contar con un prototipo para evaluar la arquitectura diseñada para la aplicación.
2. **Liberación final:** al final del tercer bimestre se debe liberar la versión final del sistema a realizar.

### Interfaces

- **Interfaces de usuario:** el sistema tendrá una única interfaz de usuario que será web, mediante la cual se podrán realizar los requisitos funcionales.
- **Interfaces con software:**
  - Una interfaz que permita interactuar con el sistema de correo, para el envío de las notificaciones a los usuarios.
  - Una interfaz que permita interactuar con redes sociales de sitios oficiales, con el fin de mantener actualizadas las tecnologías
  - Una interfaz que permita interactuar con otros sitios web oficiales para implementar el Web Scraping.

- Una interfaz que permita interactuar con las redes sociales de los usuarios para permitir su registro en el sistema.
- Una interfaz que exponga servicios de consulta de los datos por internet para permitirle a otros sistemas la integración con la misma.

## Apéndice 2: Casos de prueba del sistema

Se realizan una serie de casos de prueba, estos fueron divididos por funcionalidades similares, a continuación se detallan algunos ejemplos de los mismos:

- Control de acceso

	Operación de la API	Datos de entrada	Salida esperada
1	POST /users	<ul style="list-style-type: none"> <li>● mail</li> <li>● contraseña</li> <li>● nombre</li> <li>● país</li> <li>● sexo</li> <li>● rol : colaborador</li> <li>● nickname</li> </ul>	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: mismos datos que los ingresados, menos la contraseña y se le agrega el identificador del usuario.</li> </ul>
2	GET /users/login	Encabezado Basic Auth con mail y contraseña del usuario dado de alta en 1	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: igual que el anterior</li> <li>● Cookie de sesión jwt</li> </ul>
3	GET /users/view/id	identificador del usuario ingresado en 1 y cookie de sesión recibida en 2	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: igual que el anterior</li> </ul>
4	POST /users/admin	<ul style="list-style-type: none"> <li>● mail</li> <li>● contraseña</li> <li>● nombre</li> <li>● país</li> <li>● sexo</li> <li>● rol : administrador</li> <li>● nickname</li> <li>● cookie de sesión recibida en 2</li> </ul>	<ul style="list-style-type: none"> <li>● Código: 403 Forbidden</li> </ul>
5	GET /users/logout	<ul style="list-style-type: none"> <li>● cookie de sesión recibida en 2</li> </ul>	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cookie de sesión jwt con valores inválidos</li> </ul>
6	GET /users/view/id	Identificador del usuario recibido en 1	<ul style="list-style-type: none"> <li>● Código: 403 Forbidden</li> </ul>
7	GET /users/login	Encabezado Basic Auth con mail y contraseña de usuario administrador	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: datos del usuario</li> </ul>

			administrador <ul style="list-style-type: none"> <li>● Cookie de sesión jwt</li> </ul>
8	POST /users/admin	<ul style="list-style-type: none"> <li>● mail</li> <li>● contraseña</li> <li>● nombre</li> <li>● país</li> <li>● sexo</li> <li>● rol : administrador</li> <li>● nickname</li> <li>● cookie de sesión recibida en 6</li> </ul>	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: datos del usuario administrador</li> </ul>
9	GET /users/view/id	identificador del usuario recibido en 7 y cookie de sesión recibida en 2	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: igual que el anterior</li> </ul>

- Participación en el foro:

	Operación de la API	Datos de entrada	Salida esperada
1	GET /users/login	Encabezado Basic Auth con mail y contraseña de usuario colaborador	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: datos del usuario</li> <li>● Cookie de sesión jwt</li> </ul>
2	POST /forum/threads	<ul style="list-style-type: none"> <li>● título</li> <li>● descripción</li> <li>● cookie obtenida en 1</li> </ul>	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: datos del hilo creado</li> <li>● Cookie de sesión jwt</li> </ul>
3	GET /forum/threads	<ul style="list-style-type: none"> <li>● cookie obtenida en 1</li> </ul>	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: listado de hilos del sistema incluyendo el hilo creado en 2</li> </ul>
4	GET /users/login	Encabezado Basic Auth con mail y contraseña de usuario colaborador diferente al usado en 1	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: datos del usuario</li> <li>● Cookie de sesión jwt</li> </ul>
5	POST /forum/comments	<ul style="list-style-type: none"> <li>● contenido</li> <li>● id del hilo obtenido en 2</li> </ul>	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: datos del comentario creado</li> </ul>

		<ul style="list-style-type: none"> <li>● cookie obtenida en 6</li> </ul>	
6	GET /forum/thread/comments/id	identificador del hilo obtenido en 2 y cookie de sesión recibida en 6	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: listado de comentarios pertenecientes a ese hilo, incluyendo el agregado en 7</li> </ul>
7	PUT forum/comments/vote/id	<ul style="list-style-type: none"> <li>● id del comentario recibido en 7</li> <li>● cookie obtenida en 1</li> </ul>	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> </ul>
8	GET /users/login	Encabezado Basic Auth con mail y contraseña de usuario administrador	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: datos del usuario</li> <li>● Cookie de sesión jwt</li> </ul>
9	PUT /forum/threads/close/id	<ul style="list-style-type: none"> <li>● id del hilo obtenido en 2</li> <li>● cookie obtenida en 13</li> </ul>	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> </ul>
10	POST /forum/comments	<ul style="list-style-type: none"> <li>● contenido</li> <li>● id del hilo obtenido en 2</li> <li>● cookie obtenida en 1</li> </ul>	<ul style="list-style-type: none"> <li>● Código: 400 Bad request</li> </ul>

● Gestión de productos:

	Operación de la API	Datos de entrada	Salida esperada
1	GET /users/login	Encabezado Basic Auth con mail y contraseña de usuario administrador	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: datos del usuario administrador</li> <li>● Cookie de sesión jwt</li> </ul>
2	POST /prods	<ul style="list-style-type: none"> <li>● nombre</li> <li>● tipo de licencia</li> <li>● licencia</li> <li>● link</li> <li>● tipo</li> <li>● versión</li> <li>● estándares implementados</li> </ul>	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: datos del producto incluyendo el identificador</li> </ul>

		<ul style="list-style-type: none"> <li>● cookie obtenida en 1</li> </ul>	
3	GET /prods		<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: lista de productos incluyendo el agregado en 3</li> </ul>
4	GET /users/login	Encabezado Basic Auth con mail y contraseña de usuario colaborador	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: datos del usuario colaborador</li> <li>● Cookie de sesión jwt</li> </ul>
5	POST /prods	<ul style="list-style-type: none"> <li>● nombre</li> <li>● tipo de licencia</li> <li>● licencia</li> <li>● link</li> <li>● tipo</li> <li>● versión</li> <li>● estándares implementados</li> <li>● aprobada: false</li> <li>● cookie obtenida en 4</li> </ul>	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: datos del producto incluyendo el identificador</li> </ul>
6	GET /prods		<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: lista de productos sin el producto agregado en 5</li> </ul>
7	PUT /prods/approve/id	<ul style="list-style-type: none"> <li>● cookie obtenida en 1</li> </ul>	<ul style="list-style-type: none"> <li>● Código: 200 OK</li> </ul>
8	GET /prods		<ul style="list-style-type: none"> <li>● Código: 200 OK</li> <li>● Cuerpo: lista de productos con el producto agregado en 5</li> </ul>