



Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Posgrado y Educación Continua

Especialización en Infraestructura de Datos Espaciales

Trabajo Final

TÍTULO:

Implementación de un prototipo del Servicio Web de Procesamiento (WPS) en el marco de una Infraestructura de Datos Espaciales (IDE).

NOMBRE DEL ALUMNO:

Alicia Noemí Duarte – Ingeniera en Sistemas de Información.

DIRECTORA:

Mgter. María Alejandra Barrera.

CO-DIRECTOR:

Ing. Esp. Ariel Bouhier.

DEDICATORIA

Este trabajo se lo dedico con todo mi corazón a mi hermoso Néstor, compañero de camino en esta vida, por apoyarme incondicionalmente en todas mis iniciativas. Quedarán en nuestros recuerdos los paseos por Rosario luego de los días de cursado presencial y también compartiendo los espacios de nuestra pequeña casa durante el cursado virtual en pandemia. Gracias por enseñarme día a día a ser mejor persona, recordarme en todo momento a vivir la vida y por sobre todas las cosas brindarme comprensión, cariño y amor.

RESUMEN

Los servicios son las funcionalidades accesibles mediante una red, que una Infraestructura de Datos Espaciales (IDE) ofrece al usuario para aplicar sobre los datos geográficos. Estas funcionalidades se organizan en servicios, entre los que se destaca el Servicio Web de Procesamiento (WPS), como un estándar del Open Geospatial Consortium (OGC) para el procesado de datos, que permite publicar, localizar y usar procesos geoespaciales.

Como objetivo principal de este trabajo, se planteó la implementación de un prototipo del WPS en el marco de una IDE, para determinar la incorporación de dicho servicio, como solución a diversos requerimientos de análisis espacial por parte de usuarios para la toma de decisiones. La pregunta a responder por el prototipo fue: ¿Cuáles son las escuelas cercanas a los barrios populares para brindarle cobertura WiFi?

Como caso de estudio, y con el objeto de comprobar la aplicabilidad de la propuesta, se implementó un WPS a la Infraestructura de Datos Espaciales de Santa Fe (IDESF), que en la actualidad no brinda este servicio. Esto ayudará en la solución de requerimientos de usuarios y productores de Información Geográfica de la provincia, en tareas relacionadas a las consultas a través de áreas de influencia, entre otras.

Palabras claves: IDE, WPS, Servicios Web, WMS, WFS, OGC, GeoServer, vector, OpenLayers, QGIS.

ÍNDICE GENERAL

DEDICATORIA.....	ii
RESUMEN.....	iii
Lista de Figuras.....	vi
Lista de Tablas.....	vii
1. INTRODUCCIÓN.....	1
2. OBJETIVOS.....	4
2.1 Objetivo General.....	4
2.2 Objetivos Específicos.....	4
3. MARCO TEÓRICO.....	4
3.1 Infraestructura de Datos Espaciales	4
3.1.1 Componentes de una IDE.....	5
3.1.2 Tipos de servicios de una IDE.....	7
3.1.3 Interoperabilidad.....	10
3.1.4 Actores de una IDE.....	10
3.1.5 Open Geospatial Consortium (OGC).....	11
3.1.6 Comité Técnico ISO/TC 211 para la información Geográfica.....	12
3.2 Web Service Processing.....	12
4. METODOLOGÍA.....	15
4.1 Descripción del área de estudio.....	15
4.2 Materiales y métodos.....	16
4.2.1 Datos.....	16
4.2.2 Formato y características de los datos	17
4.2.3 Software.....	21
4.2.4 Generador de consultas.....	27

4.2.5 Sistema de Referencia de Coordenadas (SRC).....	30
4.2.7 Recolección, preparación y publicación de la información.....	33
4.3 Procesos para la implementación del prototipo WPS.....	34
5. RESULTADOS.....	44
5.1 Prototipo WPS en el marco de la IDESF.....	44
5.2 Clientes WPS.....	47
5.2.1 Integración con OpenLayers como cliente ligero WPS.....	47
5.2.2 QGIS como cliente pesado WPS.....	52
5.3 Incorporación de nuevos procesos WPS.....	55
6. DISCUSIÓN, CONCLUSIONES Y TRABAJOS FUTUROS.....	60
6.1 Discusión.....	60
6.2 Conclusiones.....	62
6.3 Trabajos Futuros.....	63
7. BIBLIOGRAFÍA.....	65
ANEXO 1. Requerimiento resuelto con QGIS.....	67
ANEXO 2. WPS – Petición GetCapabilities.....	68
ANEXO 3. WPS – Petición DescribeProcess.....	69
ANEXO 4. XML del encadenamiento de procesos del prototipo WPS.....	74
ANEXO 5. Variable de JavaScript donde se guarda el XML del prototipo WPS.....	76
ANEXO 6. Archivo pom.xml.....	77

Lista de Figuras

Figura 01: Esquema de componentes de una IDE.....	5
Figura 02: Operación GetCapabilities.....	13
Figura 03: Operación DescribeProcess.....	14
Figura 04: Operación Execute.	14
Figura 05: Santa Fe - División político administrativa.....	15
Figura 06: Acceso al Generador de consultas WPS de GeoServer.....	27
Figura 07: Generador de consultas WPS de GeoServer.....	28
Figura 08: Generador de consultas WPS - Proceso gs:Bounds.....	28
Figura 09: Generador de consultas WPS – Proceso gs:Bounds – Ventana Resultado.....	29
Figura 10: Proceso gs:Bounds - Solicitud POST WPS.....	29
Figura 11: Proyección Gauss-Krüger.....	32
Figura 12: Consulta WPS gs:BufferFeatureCollection.....	38
Figura 13: XML del proceso gs:BufferFeatureCollection.....	38
Figura 14: Consulta WPS gs:InclusionFeatureCollection.....	39
Figura 15: XML del proceso gs:InclusionFeatureCollection.....	40
Figura 16: Consulta WPS gs:Import.....	41
Figura 17: XML del proceso gs:Import.....	42
Figura 18: Previsualización de la capa.....	43
Figura 19: Prototipo WPS en OpenLayers – Capa base.....	49
Figura 20: Prototipo WPS en OpenLayers – Capas de partida.....	49
Figura 21: Prototipo WPS en OpenLayers – Escuelas dentro de Buffer BP.....	50
Figura 22: Prototipo WPS en OpenLayers – Escuelas dentro de Buffer BP-Listado.....	51
Figura 23: Prototipo WPS en OpenLayers – Buffer de Barrios Populares.....	52

Figura 24: QGIS – Plugin WPS-Client.....	53
Figura 25: QGIS – Plugin WPS-Client – Proceso gs:BufferFeatureCollection.....	53
Figura 26: QGIS – Plugin WPS-Client – Mensaje de error.....	54
Figura 27: Estructura del proyecto.	56
Figura 28: Archivo JAR del proceso.....	58
Figura 29: Archivo JAR del proceso en GeoServer.....	59
Figura 30: Generador de consultas WPS – Proceso gs:IdesfWPS.....	59
Figura 31: Generador de consultas WPS – Proceso gs:IdesfWPS – Resultado.....	60

Lista de Tablas

Tabla 1: Procesos candidatos para el prototipo.....	36
Tabla 2: Procesos para calcular distancias.....	43

1. INTRODUCCIÓN

Una Infraestructura de Datos Espaciales (IDE), como Sistema de Información Geográfica distribuido, es algo más que un servidor en funcionamiento que está publicando mapas y datos en Internet. Desde el punto de vista tecnológico hay tres componentes fundamentales de toda IDE: los datos, metadatos y servicios (Iniesto, Núñez et al, 2021).

Las IDE se basan en el modelo de arquitectura informática denominado cliente-servidor. Esta tecnología es la que generalmente utilizan las aplicaciones de Internet, donde un software llamado genéricamente *cliente* (usuario), que funciona en una computadora local, se comunica y realiza una petición a una computadora remota (*servidor*) que le responde con la información o el servicio solicitado. (Bernabé, López et al, 2012).

Los servicios IDE ofrecen funcionalidades accesibles, generalmente, a través de un navegador de Internet. Estas funcionalidades se organizan en servicios de visualización de mapas, de descarga, de consulta, etc (Iniesto, Núñez et al, 2021). Entre los más importantes se pueden mencionar:

- WMS (Web Map Service): permite publicar y consultar mapas.
- WFS (Web Feature Service): permite descargar y analizar datos vectoriales.
- WCS (Web Coverage Service): permite descargar y analizar datos raster.
- CSW (Catalogue Service Web): permite buscar datos disponibles.
- WPS (Web Processing Service): permite procesar datos geoespaciales.

Las IDE se caracterizan por el alto nivel de estandarización que presentan, la mayoría de los servicios geoespaciales desplegados en ellas utilizan interfaces definidas por el Open Geospatial Consortium¹ (OGC). Creado en 1994, el OGC es un consorcio internacional sin fines de lucro conformado por miembros comerciales, gubernamentales, académicos, entre otros. El proceso de consenso

1 <https://www.ogc.org>

impulsado por los miembros de OGC desarrolla estándares geoespaciales abiertos, con el objetivo de hacer que la información y los servicios geoespaciales sean encontrables, accesibles, interoperables y reutilizables. El OGC tiene como visión un mundo en el que todos se beneficien de la IG y que los servicios estén disponibles a través de cualquier red, aplicación o sistema.²

El Servicio Web de Procesamiento o Web Processing Service (WPS) es un estándar OGC para el procesamiento de datos. Los estándares OGC hacen posible que la información espacial pueda distribuirse en Internet.

La función de un servicio WPS es ejecutar algoritmos, cálculos y modelos de computación de distinta complejidad que operan con datos georreferenciados. Los datos requeridos por el servicio, sean vectoriales o raster, pueden estar en el mismo servidor o en servidores remotos y pueden ser accedidos a través de algún tipo de servicio estandarizado (por ejemplo, WFS o WCS). Los algoritmos pueden ser tan sencillos como restar un conjunto de números referenciados espacialmente de otros (por ejemplo, la determinación de la diferencia en los casos de COVID19 entre dos olas de contagio) o tan complicados como un modelo de simulación de incendios forestales. (OGC, 2022).

El estándar WPS permite publicar, localizar y usar procesos geoespaciales. A través de la interfaz del WPS es posible describir los datos de entrada y de salida, así como los modos de ejecución y de acceso a la información producida. El WPS permite que procesos específicos sobre datos geoespaciales se puedan publicar como servicios web para ser usados por clientes distribuidos en la web.

Las IDE al incorporar el estándar WPS pueden ofrecer servicios y recursos de cómputo robustos, como así también, servir como un medio para compartir datos espaciales. Lo cual lo convierte en un componente fundamental de una IDE a través del cual, usuarios y productores de IG pueden compartir medios para realizar geoprocesamiento de forma fiable.

En este sentido la Infraestructura de Datos Espaciales de la provincia de Santa Fe (IDESF), se define como el conjunto de políticas, estándares, procedimientos y

² <https://www.ogc.org/about>

recursos tecnológicos que facilitan la producción, obtención, uso y acceso de información geográficamente referenciada de cobertura provincial. Uno de los objetivos de la IDESF es brindar servicios de IG, facilitando a los usuarios su acceso y uso.³

La IDESF en la actualidad no brinda el Servicio WPS, por lo que este trabajo pretende ser una iniciativa para la implementación del mismo en la IDE provincial. Esto ayudará en la solución de requerimientos de usuarios y productores de Información Geográfica (IG), en tareas tales como, realizar consultas a través de áreas de influencia.

Las implementaciones disponibles para WPS según la página web del OGC⁴ son varias, entre ellas podemos mencionar los softwares: 52 North, PyWPS, Deegree, ZOO-Project y GeoServer.

En este caso se trabajó con el WPS del software GeoServer, debido a que la IDESF utiliza este software como Servidor de Mapas, lo que representa una ventaja usar la implementación asociada al mismo. GeoServer implementa el servicio WPS a través de una extensión que debe ser previamente instalada.

Como propósito del prototipo WPS, se consideró un requerimiento solicitado en el marco del programa "Santa Fe + Conectada"⁵ del gobierno de la provincia de Santa Fe: Determinar las escuelas que se encuentran a una distancia de hasta 1000 metros por fuera de las áreas comprendidas por los barrios populares, de las ciudades de Santa Fe y Rosario, para facilitarle cobertura Wifi.

IDESF asesoró al área productora de IG en la resolución del requerimiento, utilizando QGIS (ver Anexo 1). Entonces, surgió el interrogante si el mismo se podría resolver a través del servicio WPS, facilitando a un usuario sin conocimientos de manejo de QGIS obtener el resultado requerido sin grandes esfuerzos y de manera rápida, ante solicitudes que generalmente son urgentes.

3 <https://www.santafe.gob.ar/idesf/>

4 <https://www.ogc.org/resource/products/byspec>

5 <https://www.santafe.gob.ar/index.php/web/content/view/full/236957> El programa provincial "Santa Fe + Conectada" permitirá ampliar la inclusión digital y reducir la brecha tecnológica que la pandemia puso en evidencia, a través del tendido de más de 4000km de fibra óptica llegando a las 365 localidades de la Provincia.

2. OBJETIVOS

2.1 Objetivo General

Implementar un prototipo, en el marco de la IDESF, como iniciativa en la incorporación del servicio de procesamiento, para brindar a usuarios y productores de IG la posibilidad de procesar información geoespacial para la toma de decisiones en el territorio provincial.

2.2 Objetivos Específicos

- Determinar los procesos específicos de la herramienta a utilizar para la implementación del WPS (entradas, salidas, procesamiento, modo de solicitar la ejecución de los mismos y el modo de manejar la información de salida).
- Aplicar la propuesta del servicio a la IDESF para verificar su funcionamiento.
- Investigar la factibilidad de crear e incorporar nuevos procesos WPS a la herramienta a utilizar.

3. MARCO TEÓRICO

3.1 Infraestructura de Datos Espaciales

Las Infraestructuras de Datos Espaciales (IDE) se han consolidado como la estrategia tecnológica más adecuada para abordar las exigencias de la sociedad de la información, para facilitar las posibilidades de acceso de ciudadanos, instituciones y empresas a la IG y servicios de geoprocesamiento a través de la red (Iniesto, Núñez et al, 2021).

Una IDE es un conjunto de tecnologías, políticas y acuerdos institucionales, destinados a facilitar el acceso a la información espacial, que permite la búsqueda, visualización, análisis y aplicación de datos espaciales a todos los niveles (administración, empresas, sectores sin fines de lucro, la universidad y los ciudadanos) (Bernabé, López et al, 2012).

Según la Infraestructura de Datos Espaciales de la República Argentina (IDERA), las IDE surgen con el objetivo de poner en común, datos espaciales y servicios para que sean utilizados por cualquier tipo de usuario y en todo tipo de aplicaciones.

La IDE facilita el acceso a la información haciendo uso de un mínimo conjunto de estándares, protocolos y especificaciones.

3.1.1 Componentes de una IDE

Desde el punto de vista tecnológico hay tres componentes fundamentales de toda IDE: datos, metadatos, servicios. Pero no debe olvidarse de otro componente cuyo papel es primordial, que es la organización. (Iniesto, Núñez et al, 2021).



Figura 01: Esquema de componentes de una IDE.

Fuente: Iniesto, Núñez et al, 2021.

Datos

Actualmente, existe un consenso internacional por el cual se clasifican los datos espaciales que pueden manipular las IDE según la IG de base o temática. Los datos de referencia sirven como información básica, contienen IG necesaria para que cualquier usuario y aplicación pueda referenciar sus datos y son de propósito

general, descriptivos de la geografía de la zona que sirven para referenciar. Los datos temáticos son aquellos que basados en IG de referencia, singularizan o desarrollan algún aspecto concreto de la información contenida en aquella o incorporan información adicional específica y están orientados a un sector de aplicación específico. Son los valores de las distintas capas de IG, como por ejemplo: Clima, Hidrología, Vegetación, etc.

Metadatos

Los metadatos son datos que describen los conjuntos de datos geográficos y los servicios de IG y que hace posible localizarlos, inventariarlos y utilizarlos (LISIGE, 2010), es decir, son los datos sobre los propios datos o servicios, que informan sobre el contenido del conjunto de datos o las operaciones del servicio y los datos acoplados al servicio, incluyendo su nombre y resumen, cuándo fue creado el conjunto de datos o el servicio y sus distintas actualizaciones, cómo se han obtenido, cómo se puede acceder a ellos, la zona o extensión geográfica que cubren, su calidad y validez, el formato, el propietario, las condiciones que rigen el acceso a los datos o servicios, y otras características que sirven para caracterizar datos y servicios de forma que usuarios (y aplicaciones) puedan localizarlos, acceder a ellos y explotarlos adecuadamente.

La norma que regula los metadatos de la IG es la norma ISO 19115 “Geographic Information – Metadata”. La norma específica para servicios es la ISO 19119 “Geographic information – Services”.

Servicios

Los servicios son las funcionalidades accesibles mediante una red que una IDE ofrece al usuario para aplicar sobre los datos geográficos o sus metadatos, se mencionan a menudo como servicios web.

Éstos, en función de las operaciones que nos permiten realizar, se organizan en servicios de visualización de mapas (WMS, WMTS), de localización (CSW), de descarga (WFS, WCS), de transformación de coordenadas (WCTS), de procesamiento (WPS), entre otros.

Organización

La organización es el componente más complejo y el que hace que el resto funcione y se mantenga, incluye el personal humano, una estructura organizativa y de reparto del trabajo, la estandarización para garantizar la interoperabilidad dentro y fuera de la organización. Establece políticas, directivas, reglas y acuerdos entre los productores de datos, proporcionando una plataforma estructurada de datos, productos y servicios.

Todos los componentes son necesarios, pero la organización es de especial importancia en una IDE porque ordena, regula, estructura y armoniza todos los demás, lo que garantiza la eficiencia, como también, el uso de los datos y servicios geográficos de una IDE.

3.1.2 Tipos de servicios de una IDE

La publicación de la información se hace, a través de servicios, respetando estándares internacionales y normativas acordadas en el marco de la propia IDE.

La definición técnica que la Norma ISO 19119:2005 presenta para Servicios es: *“servicio es una parte distinguible de la funcionalidad proporcionada por una entidad a través de una interfaz”*. (Bernabé, López et al, 2012).

Existen dos organizaciones que trabajan conjuntamente en los datos geográficos: el OGC que define especificaciones de servicios e ISO TC/211 que emite las normas de servicios. Los estándares OGC más relevantes se han aprobado después como normas de ISO/TC211.

Entre los servicios más importantes especificados por el OGC, se puede mencionar:

- Servicio Web de Mapa (WMS): Permite la visualización de una imagen cartográfica generada a partir de una o varias fuentes: mapa digital, datos de un SIG, ortofoto, etc., provenientes de uno o varios servidores. Este servicio se solicita a través del navegador web o aplicación SIG del usuario que envía una petición en forma de URL. Esta petición se recibe y procesa por el servidor WMS que, como respuesta, devuelve al usuario una imagen

en formato jpeg, gif, png, etc. La definición de un formato u otro garantiza la transparencia de las capas de información, permitiendo la combinación de capas procedentes de diferentes servicios WMS. Este permite también opcionalmente consultar los atributos alfanuméricos de la información que se visualiza.

- Servicio Web de Teselas de Mapa (WMTS): Servicio muy similar al WMS, excepto que discretiza el espacio en un conjunto de niveles de zoom predefinidos y para cada uno de ellos define una matriz regular de teselas. Ello permite sacar provecho de los mecanismos de caché de servidor que existen actualmente en la web, por lo que las interacciones con servidores utilizadas muy frecuentemente deben resultar más ágiles que los servidores WMS en igualdad de condiciones. Sin embargo, los clientes WMTS resultan un poco más complejos dado que, generalmente, deben realizar diversas peticiones de teselas adyacentes hasta llenar el área de navegación con cada acción del usuario.
- Servicio Web de Objetos (WFS): Permite acceder a los datos mediante el empleo del formato GML. Este servicio permite acceder y consultar los atributos de un objeto (feature) geográfico como un río, una ciudad o un lago, representado en modo vectorial. Un WFS permite no sólo visualizar la información tal y como lo permite un WMS, sino que también permite acceder a la información y descargarla. Este servicio dispone como operación optativa la que permite manipular (editar, borrar, crear) la información almacenada en base de datos.
- Servicio Web de Coberturas (WCS): Es el servicio análogo al WFS pero para datos raster, como imágenes satelitales y modelos digitales del terreno. Permite no sólo visualizar la información raster, como lo permite un WMS, sino además permite consultar el valor del o los atributos almacenados en cada píxel.
- Servicios Web de Catálogo (CSW): Este servicio permite buscar la IG que se necesita en base a los metadatos que la definen. Permite la publicación

y búsqueda de información que describe datos, servicios, aplicaciones y en general, todo tipo de recursos de la IDE. Los servicios de catálogo, que están basados en el acceso a los metadatos de los datos y los servicios, son necesarios para proporcionar capacidades de búsqueda y solicitud de los recursos dentro de una IDE.

- Servicio de Nomenclátor (Gazetteer): Es un caso específico del servicio WFS ya que ofrece la posibilidad de localizar un objeto geográfico de nombre dado y consultar los atributos que tenga asociados. El servicio une cada nombre geográfico con su localización en base a coordenadas. Es un servicio web muy importante, ya que es el modo más natural de seleccionar la zona que el usuario quiere ver o consultar. Sus principales funcionalidades son: devolver la geometría compleja (polígonos y líneas) de las entidades ligadas con el nombre del topónimo buscado, combinar topónimos con otros criterios de búsqueda (búsquedas espaciales, por tipo de entidad, etc.) y localizar información literal mediante textos o búsquedas espaciales.
- Descriptor de Estilos de Capa (SLD): Es una extensión de la especificación WMS que permite a los usuarios utilizar estilos de simbolización propios, permitiendo definir cómo se va a representar la IG a través de la web.
- WCTS (Servicio Web de Transformación de Coordenadas): Permite transformar puntos y archivos de datos geográficos de un Sistema de Referencia de Coordenadas (CRS) a otro.
- Servicio Web de Procesamiento (WPS): Permite definir utilidades de geoprocésamiento en la web, utilizando una entrada de datos, preferiblemente estándar, realizando un proceso (como un cálculo de rutas, de un buffer, etc.) y devolviendo una salida bien definida.
- Web Map Context (WMC): Permite guardar en un formato estándar y en XML la situación activa de visualización en un momento determinado (qué WMS se están viendo, qué capas, qué zona, a qué escala, en qué CRS) y reproducirla en cualquier visualizador estándar.

3.1.3 Interoperabilidad

La interoperabilidad es la condición mediante la cual, sistemas heterogéneos pueden intercambiar procesos o datos (Inierto, Núñez et al, 2021).

La Norma ISO 19119 establece que: La interoperabilidad es la capacidad para comunicar, ejecutar programas, o transferir datos entre varias unidades funcionales sin necesitar que el usuario tenga conocimiento de las características de esas unidades.

La obligación de lograr la interoperabilidad conduce a la necesidad de abordar la creación y adopción de estándares. Los propósitos de los estándares en la geoinformación son:

- Proporcionar una buena especificación semántica (tener bien definido el significado de los datos que se quieren intercambiar).
- Proporcionar especificaciones sobre formatos.
- Reducir costos en la implementación y mantenimiento del software.
- Reducir costos al evitar duplicaciones. Cada productor de datos se encarga de mantener sus datos actualizados.
- Disponer de plataformas y formatos no propietarios.
- Mejorar la colaboración.

3.1.4 Actores de una IDE

En una IDE, entendida como sistema distribuido en la red, están involucrados distintas instituciones, organismos y usuarios (administraciones públicas, universidades, empresas privadas, etc.), tanto públicos como privados, que llamaremos actores, cada uno con un tipo de participación y responsabilidad diferente, cada uno con su rol o función (Inierto, Núñez et al, 2021).

- Productores de datos: producen y mantienen datos geográficos (mapas, MDT, imágenes, ortofotos, etc.), además de difundirlos a la sociedad a través de servicios de visualización, de descarga, de consulta, etc. Dependiendo de los tipos de datos estos actores pueden ser organismos públicos (institutos geográficos, catastros, institutos de estadísticas y censos) u organizaciones privadas.
- Desarrolladores de software, generan los programas y aplicaciones que permiten publicar un servicio o implementar un Geoportal, desde el que puedan verse y utilizarse los datos.
- Intermediarios (brokers), adaptan e integran las soluciones y componentes existentes para proporcionar un sistema completo y a la medida para usuarios y organizaciones no expertos.
- Universidades, investigan e innovan. Desarrollan algoritmos, métodos, programas y soluciones que no existen en el mercado, para que la tecnología progrese y evolucione.
- Usuarios, utilizan los servicios que proporciona una IDE para solucionar sus problemas. Demandan información. Puede ser un ciudadano individual, un organismo público, una empresa privada, una universidad, una asociación o cualquier agente social. El usuario es el actor más importante de una IDE. Todo se hace por él, para él y pensando en él. Cada vez se le da más importancia a su opinión, su capacidad de decisión y su grado de satisfacción.

3.1.5 Open Geospatial Consortium (OGC)

El Open Geospatial Consortium (OGC) es una organización internacional sin fines de lucro, conformada por miembros comerciales, gubernamentales, académicos y empresas de los sectores público y privado, dedicado al desarrollo de estándares de servicios basados en localización y geoinformación. Su objetivo es promover el desarrollo y uso de estándares y tecnologías abiertas en el campo de la IG, para

lograr que los servicios estén disponibles a través de cualquier red, aplicación o sistema. Para ello define, por consenso, especificaciones de interoperabilidad que están disponibles para uso global (Bernabé, López et al, 2012).

3.1.6 Comité Técnico ISO/TC 211 para la información Geográfica

El Comité Técnico ISO/TC 211 para la IG, tiene como objetivo desarrollar un conjunto estructurado de normas internacionales sobre «los métodos, herramientas y servicios para la gestión de datos, adquisición, procesamiento, análisis, acceso y disponibilidad de la IG, haciendo posible la interoperabilidad geoespacial». Estas normas se agrupan en la denominada familia ISO 19100, constituida por normas internacionales relacionadas con objetos o fenómenos que se asocian a localizaciones sobre la superficie terrestre (Bernabé, López et al, 2012).

3.2 Web Service Processing

Web Processing Service (WPS) es una especificación del OGC que define una interfaz estandarizada para facilitar el acceso a servicios de procesamiento a través de la Web, proporcionando reglas para las entradas y las salidas de dichos servicios (OGC, 2022). WPS ofrece y permite ejecutar procesos geoespaciales (geoprocesos) simples o complejos. Se entiende por proceso cualquier algoritmo, cálculo o modelo, que opera sobre datos espacialmente referenciados tanto en formato raster como vectorial. De este modo, un WPS puede ofrecer cualquier tipo de funcionalidad SIG a través de la red (ej. generación de áreas de influencia, algoritmos de análisis, etc.). (IDEE, 2021).

La primera versión del WPS se publicó en 2004 y fue aprobado como un estándar oficial por OGC, para el procesamiento de datos remotos, en junio de 2007 con la versión 1.0.0 (Schut, 2007). En el año 2015 se publicó WPS 2.0, siendo la versión actual hasta la fecha.

Los datos de entrada del WPS pueden encontrarse almacenados en el propio servidor o pueden ser accedidos remotamente utilizando otros servicios del OGC, tales como WFS o WCS.

El objetivo principal del WPS es definir un protocolo de comunicación basado en XML (Extensible Markup Language) para el geoprocesamiento remoto. Las aplicaciones del lado del cliente interactúan con el WPS incorporando parámetros a la dirección URL del servicio. La especificación WPS del OGC define tres operaciones obligatorias, que pueden invocarse desde un cliente: *GetCapabilities*, *DescribeProcess* y *Execute*.

Operación GetCapabilities

La operación GetCapabilities permite obtener los metadatos referentes al servicio y la definición de todos los procesos que brinda el servidor. La respuesta está contenida en un documento XML.

Parámetro	Descripción	Obligatoriedad
Request=GetCapabilities	Nombre de la operación que se realiza en la petición.	Obligatorio
Service=WPS	Tipo de servicio sobre el cual se realiza la petición: WPS.	Obligatorio
AcceptVersions	Versión de la especificación del OGC. Las actuales son 1.0.0. y 2.0	<i>Opcional</i>
language	Indica el idioma en el que se desea recibir la respuesta. La referencia al idioma debe utilizar los códigos RFC4646 ² .	<i>Opcional</i>
Extension	Elemento contenedor de otras especificaciones	<i>Opcional</i>

Figura 02: Operación *GetCapabilities*.

Fuente: Iniesto, Núñez et al, 2021.

Operación DescribeProcess

La operación DescribeProcess devuelve como respuesta un archivo XML que contiene una definición detallada del proceso solicitado, cómo ser número y tipo de entradas/salidas, como así también formatos válidos para éstas.

Parámetro	Descripción	Obligatoriedad
Request=DescribeProcess	Nombre de la operación que se realiza en la petición.	Obligatorio
Service=WPS	Tipo de servicio sobre el cual se realiza la petición: WPS.	Obligatorio
Version³	Versión de la especificación del OGC.	Obligatorio
language	Indica el idioma en el que se desea recibir la respuesta. La referencia al lenguaje debe utilizar los códigos RFC4646. Debe ser un idioma que figure en el elemento <i>Languages</i> del documento de capacidades.	<i>Opcional</i>
Identifier	Indica 1 ó más identificadores. Especifica los procesos de los que se solicita la descripción. El valor de este parámetro será el obtenido en la sección "ProcessOfferings" del documento de metadatos del servicio.	Obligatorio

Figura 03: Operación DescribeProcess.

Fuente: Iniesto, Núñez et al, 2021.

Operación Execute

La operación Execute se utiliza para solicitar la ejecución de un proceso implementado por el servicio, proporcionando los valores de los parámetros de entradas para obtener como respuesta los datos de salida de dicho proceso.

Parámetro	Descripción	Obligatoriedad
Request=Execute	Nombre de la operación que se realiza en la petición.	Obligatorio
Service=WPS	Tipo de servicio sobre el cual se realiza la petición: WPS.	Obligatorio
Version	Versión de la especificación del OGC.	Obligatorio
Language	Indica el idioma en el que se desea recibir la respuesta. La referencia al idioma debe utilizar los códigos RFC4646. Debe ser un idioma que figure en el elemento <i>Languages</i> del documento de capacidades.	<i>Opcional</i>
Identifier	Indica 1 ó más identificadores. Especifica los procesos de los que se solicita la descripción. El valor de este parámetro será el obtenido en la sección "ProcessOfferings" del documento de metadatos del servicio.	Obligatorio
Mode	Texto que contiene el modo de ejecución: <i>sync</i> (síncrono), <i>async</i> (asíncrono) o <i>auto</i> (automático, delega la decisión al servidor)	Obligatorio
Response	Define el tipo de respuesta del WPS pudiendo ser "datos puros" o un documento XML.	Obligatorio
Input	Datos de entrada proporcionados al proceso a ejecutar	<i>Opcional</i>
Output	Descripción de la información de salida esperada como resultado de la ejecución del proceso, incluyendo el formato deseado y el modo de transmisión para cada dato.	<i>Opcional</i>

Figura 04: Operación Execute.

Fuente: Iniesto, Núñez et al, 2021.

Los protocolos web utilizados para las operaciones del servicio WPS son los pares clave valor en la URL (petición HTTP GET) y las peticiones en codificación XML (petición HTTP POST).

4. METODOLOGÍA

En este capítulo se presenta el área de estudio, como así también se especifican los datos, software y herramientas empleados para lograr el objetivo del presente trabajo. Se documenta el procedimiento realizado para la implementación del prototipo del WPS en el marco de la IDESF.

4.1 Descripción del área de estudio

La provincia de Santa Fe está ubicada en la región Centro-Este de la República Argentina y su ciudad capital es Santa Fe de la Vera Cruz. La superficie⁶ total de la provincia es de 133.007 km², está dividida político-administrativamente por 19 departamentos y 365 distritos⁷.

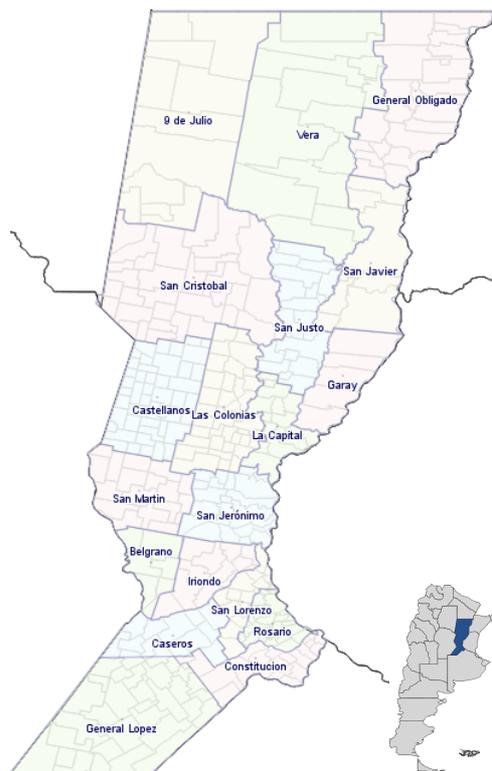


Figura 05: Santa Fe - División político administrativa.

Elaboración propia.

6 IGN – Instituto Geográfico Nacional

7 SCIT – Servicio de Catastro e Información Territorial de la Provincia de Santa Fe

Santa Fe es una extensa llanura que oscila entre los 10 y los 125 metros sobre el nivel del mar. Esta oscilación permite distinguir dos regiones geográficas: la Llanura Chaqueña y la Llanura Pampeana.

Geográficamente limita al Este con Entre Ríos y Corrientes, al Norte con Chaco, al Oeste con Santiago del Estero y Córdoba, y al Sur con Buenos Aires.

A nivel nacional, Santa Fe con 3.194.537 habitantes⁸ (según Censo Nacional de Población, Hogares y Viviendas de 2010) es la tercera provincia con más habitantes, luego de Buenos Aires y Córdoba. El 75% de la población total se concentra en 6 de los 19 departamentos, y el 37% en uno de ellos, el departamento Rosario.

Según los últimos datos censales, la provincia de Santa Fe ocupa el cuarto lugar en cuanto a densidad poblacional, con 24 hab/km², variando desde 1,8 en el departamento 9 de Julio a 631,5 en el departamento de Rosario. Esta diferencia revela las distintas realidades del norte y el sur de la provincia. La población es fundamentalmente urbana⁹, el 90,85% habita en ciudades, siendo las más habitadas Rosario y Santa Fe.

4.2 Materiales y métodos

En este apartado se especifican los datos necesarios para la implementación del prototipo, definiendo formato y características de los mismos. Además, se incluyen las herramientas utilizadas, todas ellas de código abierto (Open Source) y el sistema de referencia de coordenadas con el cual se trabajó.

4.2.1 Datos

Las capas necesarias para el prototipo son las correspondientes a las escuelas estatales de la provincia de Santa Fe y los barrios populares de las ciudades de Santa Fe y Rosario.

8 Anuario Estadístico de la República Argentina 2017

9 INDEC. Censo Nacional de Población, Hogares y Viviendas 2010. Población urbana refiere a la población que reside en áreas de 2000 o más habitantes.

La capa de escuelas estatales se descargó de la página de descargas del Geoportal de la IDESF¹⁰:

Link de descarga: <https://www.santafe.gob.ar/idesf/descargas>

Nombre: escuelas_estatal

Formato: Shapefile (.shp)

Tipo de geometría: puntos

EPSG: 22185

Metadatos:

<https://www.santafe.gob.ar/idesf/catalogo/srv/spa/metadata.show?uuid=0ec97816-52e0-49f7-b0e7-dd0661389e55>

Responsable: Ministerio de Educación de la Provincia de Santa Fe.

La capa correspondiente a los Barrios Populares se descargó desde Nación¹¹:

Link de descarga:

https://datosabiertos.desarrollosocial.gob.ar/dataset/0d022767-9390-486a-bff4-ba53b85d730e/resource/0396a8a3-8998-4fc1-8503-e6706667b5fa/download/2022-07-13_info_publica.gpkg

Nombre: 2022-07-13_info_publica

Formato: GeoPackage (.gpkg)

Tipo de geometría: polígonos

EPSG: 4326

Metadatos:

https://datos.gob.ar/dataset/desarrollo-social-registro-nacional-barrios-populares/archivo/desarrollo-social_0396a8a3-8998-4fc1-8503-e6706667b5fa

Responsable: Secretaría de Integración Socio Urbana del Ministerio de Desarrollo Social.

4.2.2 Formato y características de los datos

Shapefile

Shapefile¹² (.shp) es un formato vectorial de datos espaciales, desarrollado por la compañía ESRI (Environmental System Research Institute), originalmente se creó

¹⁰ <https://www.santafe.gob.ar/idesf/geoportal/>

¹¹ <https://datos.gob.ar/dataset/desarrollo-social-registro-nacional-barrios-populares>

¹² <https://www.esri.com/content/dam/esrisites/sitecore-archive/Files/Pdfs/library/whitepapers/pdfs/shapefile.pdf>

para la utilización con su producto ArcView GIS, pero se ha convertido en formato estándar “de facto” para el intercambio de IG.

Un shapefile es un formato vectorial sencillo y no topológico que se utiliza para almacenar la ubicación de entidades geográficas y la información de los atributos asociados a ellas. Las entidades geográficas de un shapefile se pueden representar por medio de puntos, líneas o polígonos y cada shapefile sólo puede tener un tipo de estos elementos.

El formato shapefile define la geometría y los atributos de entidades a la que se hace referencia geográfica en tres o más archivos con extensiones de archivo específicas. Los siguientes archivos son necesarios:

- .shp - archivo principal que almacena la geometría de las entidades.
- .shx - archivo que almacena el índice de las entidades geométricas.
- .dbf - archivo que almacena la información de los atributos de las entidades.

Entre la geometría y los atributos existe una relación de uno a uno, basada en el número de registro. Cada archivo debe tener el mismo prefijo, por ejemplo rutas.shp, rutas.shx y rutas.dbf corresponden a una capa de rutas en formato shapefile.

Para mejorar el funcionamiento en las operaciones de consulta se pueden incorporar algunos opcionales:

- .sbn y .sbx – archivos que almacenan el índice espacial de las entidades.
- .fbn y .fbx – archivos que almacenan el índice espacial de las entidades para los shapefiles que son sólo de lectura.
- .ain y .aih – archivos que almacenan el índice de atributo de los campos activos en una tabla o una tabla de atributos del tema.
- .prj - archivo que guarda la información referida al sistema de coordenadas en formato ESRI WKT.
- .xml – archivo que almacena los metadatos del shapefile.

- .cpg – archivo opcional que permite especificar el set de caracteres, para identificar la codificación de caracteres usado en el archivo .dbf.

GeoPackage

GeoPackage¹³ (.gpkg) es un formato abierto, basado en estándares, independiente de la plataforma, portátil, autodescriptivo y compacto para transferir información geoespacial (GeoPackage, 2022). Fue aprobado por el OGC en febrero de 2014. Está construido sobre la base de SQLite, de modo que lo hace interoperable en todos los entornos.

Entre sus principales características podemos mencionar: se trata de un solo archivo, permite almacenar varias capas de información en un mismo archivo (vectoriales y raster), se puede incluir tablas de atributos no espaciales, es soportado por varios software SIG y permite almacenar gran cantidad de datos (140 Tb).

XML (Extensible Markup Language)

El Lenguaje de Marcado Extensible¹⁴ (Extensible Markup Language XML) fue creado en 1996 por el Word Wide Web Consortium (W3C), partiendo de las especificaciones del Lenguaje Estándar Generalizado de Marcas (SGML-ISO 8879). La primera definición que apareció fue la de: «sistema para definir, validar y compartir formatos de documentos en la web» (Inierto, Núñez et al, 2021). También se conoce como “metalenguaje”, debido a que permite la creación de otros lenguajes de marcado específicos a un dominio (Bernabé, López et al, 2012).

El XML es un formato de texto simple y muy flexible que permite el intercambio de una amplia variedad de datos en la web principalmente. Diseñado para almacenar y transmitir datos, sobre todo alfanuméricos. Se compone de elementos XML, cada uno de los cuales consta de una etiqueta de inicio (<title>), de una etiqueta

13 <https://www.geopackage.org/>

14 <https://www.w3.org/XML/>

de fin (</title>) y de los datos comprendidos entre ambas etiquetas (el contenido), permite codificar documentos en un formato estructurado que es interpretable tanto por máquinas (programas) como por las personas.

Se dice que un documento está bien formado cuando tiene todos los elementos bien anidados y sus elementos cumplen las especificaciones XML.

XML, es el lenguaje estándar mediante el cual se articula la comunicación entre los diferentes componentes de una IDE cuando se trata de intercambiar información alfanumérica. La mayoría de estándares OGC utilizan como base de codificación el estándar XML de W3C, que define un formato de texto controlado por un documento de esquema XSD (Bernabé, López et al, 2012).

GML (Geography Markup Language)

El Lenguaje de Mercado Geográfico¹⁵ (GML) es el lenguaje XML para la IG. GML sirve como lenguaje de modelado para sistemas geográficos, así como también un formato de intercambio abierto para transacciones geográficas en Internet (OGC, 2022). GML fue desarrollado por OGC para la codificación, almacenamiento, y transporte de la IG en Internet. GML también es un estándar ISO (ISO 19136:2007) (OGC, 2022). Como cualquier codificación XML, GML representa la IG en formato texto.

GML se basa en el modelo geográfico abstracto desarrollado por OGC. Este modelo describe el mundo en término de entidades geográficas llamadas objetos geográficos, y son comprensibles tanto por las computadoras como por los usuarios. Esencialmente, un objeto geográfico no es más que una lista de propiedades y geometrías. Las propiedades tienen normalmente un nombre, un tipo, un valor y una descripción. Las geometrías están compuestas de geometrías básicas tales como puntos, líneas, curvas, superficies y polígonos.

Es la codificación por defecto usada por los servicios OWS, tales como WFS (este protocolo, lee y escribe datos GML) (Inierto, Núñez et al, 2021).

¹⁵ <https://www.ogc.org/standards/gml>

GeoJSON

GeoJSON¹⁶ es un formato para codificar una variedad de estructuras de datos geográficos. En 2015, el Grupo de Trabajo de Ingeniería de Internet (IETF-Internet Engineering Task Force), junto con los autores de la especificación original, formaron un Grupo de Trabajo para estandarizar GeoJSON. El RFC 7946 se publicó en agosto de 2016 y es la nueva especificación estándar del formato GeoJSON, que reemplaza la especificación GeoJSON de 2008 (GeoJSON, 2022).

GeoJSON es un formato de intercambio de datos geoespaciales de código abierto basado en JSON (JavaScript Object Notation), que representa entidades geográficas sencillas y sus atributos no espaciales. Es adoptado e implementado por servidores de mapas, a pesar de no ser un estándar del OGC. El formato es ampliamente utilizado en aplicaciones de mapas en entornos web al permitir el intercambio de datos de manera rápida, ligera y sencilla. GeoJSON usa el sistema de referencia de coordenadas geográficas WGS84 y unidades en grados decimales.

Actualmente, GeoJSON desempeña un papel importante y creciente en muchas bases de datos espaciales, API web y plataformas de datos abiertos. En consecuencia, los implementadores exigen cada vez más la estandarización formal, mejoras en la especificación, orientación sobre la extensibilidad y los medios para utilizar conjuntos de datos GeoJSON más grandes.

4.2.3 Software

GeoServer

GeoServer¹⁷ es un servidor de software de código abierto implementado en Java, que permite a los usuarios compartir, gestionar y editar datos geoespaciales. Ejecuta sobre plataformas Linux, Mac OSX, Microsoft Windows. GeoServer puede leer datos de diversas fuentes de datos espaciales y generar resultados en varios

¹⁶ <https://geojson.org/>

¹⁷ <https://geoserver.org/>

formatos. Diseñado para ser interoperable, permite la publicación de datos de diversas fuentes de datos espaciales, utilizando estándares abiertos.

Fue iniciado en 2001 por The Open Planning Project (TOPP), una incubadora de tecnología sin fines de lucro con sede en Nueva York. Actualmente en su segunda década, GeoServer continúa con su misión de hacer que los datos espaciales sean más accesibles para todos.

GeoServer implementa los estándares del OGC, tales como WMS, WFS y WCS. GeoServer forma un componente central de la Web Geoespacial.

GeoServer, servidor de información geoespacial abierto, es software libre. Puede ser redistribuido y/o modificado bajo los términos de la Licencia Pública General de GNU publicada por la Free Software Foundation; ya sea la versión 2 de la Licencia, o cualquier versión posterior (colectivamente, "GPL").

Extensión WPS de GeoServer

Web Processing Service (WPS) es un servicio OGC para la publicación de procesos, algoritmos y cálculos geoespaciales. El servicio WPS no forma parte de GeoServer de forma predeterminada, pero está disponible como una extensión, que proporciona la operación "execute" para procesamiento de datos y análisis geoespacial.

La principal ventaja del WPS de GeoServer sobre un WPS independiente es la integración directa con otros servicios de GeoServer y el catálogo de datos. Esto significa que es posible crear procesos basados en datos servidos en GeoServer, en lugar de enviar toda la fuente de datos en la solicitud. También es posible que los resultados de un proceso se almacenen como una nueva capa en el catálogo de GeoServer. De esta manera, WPS actúa como una herramienta de análisis geoespacial remoto completo, capaz de leer y escribir datos desde y hacia GeoServer.

La extensión GeoServer WPS incluye un generador de solicitudes (WPS Request Builder) para probar los procesos wps a través de la interfaz de administración

web de GeoServer. Esta herramienta también se puede utilizar para demostrar procesos y construir ejemplos propios.

El WPS brindado por el software GeoServer aparece en la lista de implementaciones actuales, publicado por el OGC, para la especificación WPS 1.0.0¹⁸.

PostgreSQL

PostgreSQL¹⁹ es un potente sistema de base de datos relacional de objetos de código abierto, publicado bajo la licencia de mismo nombre²⁰, que utiliza y amplía el lenguaje SQL. Los orígenes de PostgreSQL se remontan a 1986 como parte del proyecto POSTGRES en la Universidad de California en Berkeley.

PostgreSQL se destaca por su arquitectura comprobada, confiabilidad, integridad de datos, conjunto sólido de funciones, extensibilidad y la dedicación de la comunidad de código abierto detrás del software para ofrecer soluciones innovadoras y de alto rendimiento. PostgreSQL se ejecuta en todos los principales sistemas operativos, y tiene potentes complementos como la popular extensión de base de datos geoespacial PostGIS.

Como muchos otros proyectos Open Source, el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyados por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

PostgreSQL intenta cumplir con el estándar SQL, admite muchas de las características requeridas por dicho estándar, aunque a veces con una sintaxis o función ligeramente diferente. A partir del lanzamiento de la versión 14 en septiembre de 2021, PostgreSQL cumple con al menos 170 de las 179 características obligatorias para la conformidad con SQL:2016 Core.

18 <https://www.ogc.org/resource/products/byspec>

19 <https://www.postgresql.org/>

20 <https://opensource.org/licenses/postgresql>

PostGIS

PostGIS²¹ es una extensión espacial para la base de datos relacional PostgreSQL. Le añade soporte para las siguientes funcionalidades: tipos de datos, índices y funciones espaciales, convirtiéndola en una base de datos espacial.

PostGIS fue creado por la empresa canadiense Refrations Research Inc²², en mayo de 2001 lanzó la primera versión del software. PostGIS actualmente es un proyecto de la Open Source Geospatial Foundation (OSGeo) y es desarrollado por un grupo de colaboradores dirigidos por un Comité Directivo del Proyecto.

PostGIS ha sido certificado en 2006 por el OGC lo que garantiza la interoperabilidad, facilitando el intercambio de IG.

PostGIS es un proyecto de código abierto y su código fuente se publica bajo la GNU General Public License (GPLv2²³ o posterior).

pgAdmin

pgAdmin²⁴ es una herramienta de gestión para PostgreSQL, presenta una interfaz gráfica que facilita la creación, el mantenimiento y el uso de objetos de la base de datos. Es un proyecto de software libre multiplataforma (Windows, Linux, macOS) lanzado bajo la licencia PostgreSQL²⁵. El software está disponible en formato fuente y binario.

pgAdmin 4 es una reescritura completa de pgAdmin (su versión predecesora es pgAdmin III desarrollada en C++), construida usando los lenguajes de programación Python y Javascript/jQuery.

21 <https://postgis.net/>

22 <http://www.refrations.net/>

23 <http://opensource.org/licenses/gpl-2.0.php>

24 <https://www.pgadmin.org/>

25 <https://www.pgadmin.org/licence/>

Librería – OpenLayers

OpenLayers²⁶ facilita la incorporación de un mapa dinámico en cualquier página web. Puede mostrar mosaicos de mapas, datos vectoriales y marcadores cargados desde cualquier fuente. OpenLayers ha sido desarrollado para promover el uso de IG de todo tipo. Es una librería Open Source basada en Javascript del lado del cliente (en el modelo cliente/servidor), publicada bajo la licencia FreeBSD (OpenLayers, 2022).

OpenLayers es una iniciativa muy difundida de la OSGeo. Interactúa tanto con servicios OGC, utilizando métodos estándares para acceder a los datos geográficos (entre ellos WMS y WFS), como con servidores de mapas comerciales (Google, ESRI, etc.) (Iniesto, Núñez et al, 2021), permitiendo así combinar información de distintos servidores. Renderiza datos vectoriales de GeoJSON, TopoJSON, KML, GML, y otros formatos (OpenLayers, 2022).

Es modular, ofrece alto rendimiento y tiene muchas funciones para mostrar e interactuar con los mapas y datos geoespaciales. Una característica importante es que OpenLayers permite la interacción con servidores WPS (Zader, 2016).

QGIS

QGIS²⁷ es un proyecto oficial de OSGeo. Corre sobre plataformas Linux, Unix, Mac OSX, Microsoft Windows. Soporta numerosos formatos y funcionalidades, datos vector y raster a través de las bibliotecas GDAL y OGR, así como bases de datos, por ejemplo PostgreSQL/PostGIS.

El proyecto nació en mayo de 2002 y se estableció como un proyecto en SourceForge en junio del mismo año. QGIS se desarrolla usando el kit de herramientas Qt y C++. Esto significa que es ligero y tiene una interfaz gráfica de usuario (GUI) agradable y fácil de usar.

²⁶ <https://openlayers.org/>

²⁷ <https://www.qgis.org/>

Este SIG de Código Abierto se distribuye bajo la Licencia Pública General GNU (GPL). El desarrollo de QGIS bajo esta licencia significa que se puede revisar y modificar el código fuente, garantizando el acceso libre de costo.

QGIS tiene una infraestructura de complemento, que permite ampliar las herramientas y capacidades estándar con las que cuenta a través del uso y desarrollo de complementos.

Complemento QGIS - Plugin WPS Client

El plugin WPS Client²⁸ permite conectar QGIS con servidores WPS para ejecutar procesos. Se puede instalar el complemento directamente desde el Administrador de complementos de QGIS.

Eclipse y Maven

Eclipse²⁹ es un Entorno de Desarrollo Integrado (IDE), de Código abierto y Multiplataforma. Preferentemente se utiliza para desarrollar lo que se conoce como Aplicaciones de Cliente Enriquecido, entorno de desarrollo integrado, opuesto a las aplicaciones Cliente-liviano, entorno de desarrollo integrado basadas en navegadores. Es una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java.

Maven³⁰ es una herramienta para construir y gestionar cualquier proyecto basado en Java. Posee la capacidad de realizar la compilación del código y su empaquetado, es decir, hace posible la creación de software con dependencias incluidas dentro de la estructura del JAR.

Maven construye un proyecto utilizando un Project Object Model (POM) y un conjunto de complementos. El POM es un archivo propio de todo proyecto Maven,

28 <https://plugins.qgis.org/plugins/wps/>

29 <https://www.eclipse.org/>

30 <https://maven.apache.org/>

donde se definen todas las dependencias del proyecto (librerías externas utilizadas) (Apache Maven, 2022).

Eclipse trae un Maven embebido, pero para disponer de la última versión tal vez sea necesario descargar el plugin desde su página oficial, ya que la versión embebida no suele ser la más actualizada.

4.2.4 Generador de consultas

La extensión WPS de GeoServer incluye un generador de solicitudes para probar procesos WPS a través de su interfaz de administración web. Esta herramienta facilita la construcción de ejemplos personalizados (GeoServer, 2022).

El *Generador de consultas WPS*³¹ (WPS Request Builder) se encuentra en la sección “Demos de GeoServer” (opción “Demos”, del panel izquierdo, de la interfaz de administración web de GeoServer).



Figura 06: Acceso al Generador de consultas WPS de GeoServer.

El *Generador de consultas WPS* consta principalmente de un cuadro de selección donde se listan todos los procesos disponibles y dos botones, uno para enviar la solicitud WPS y otro que genera la solicitud POST del mismo (ver Figura 07).

31 <https://docs.geoserver.org/2.21.x/en/user/services/wps/requestbuilder.html>

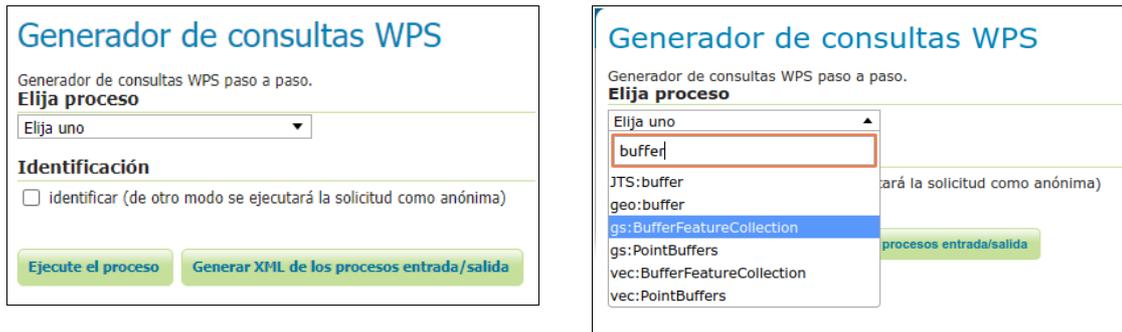


Figura 07: *Generador de consultas WPS de GeoServer.*

La pantalla cambia según el proceso y la entrada seleccionados. Para cada proceso, se muestra un formulario basado en los parámetros requeridos y opcionales (si corresponde) asociados con ese proceso (Figura 08).



Figura 08: *Generador de consultas WPS - Proceso gs:Bounds.*

A través del botón “*Ejecute el proceso*” se accede a la ejecución del proceso. La respuesta se muestra en una ventana (Figura 09) o se obtiene un archivo de salida.

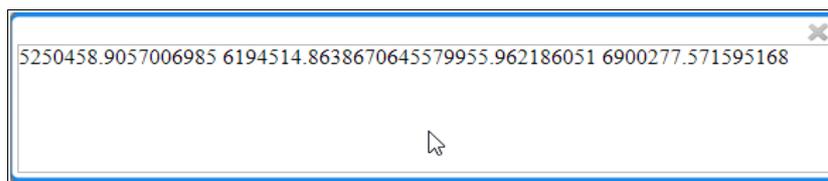


Figura 09: Generador de consultas WPS – Proceso *gs:Bounds* – Ventana Resultado.

Presionando el botón “Generar XML de los procesos entrada/salida” se obtiene el proceso como una solicitud POST (Figura 10).

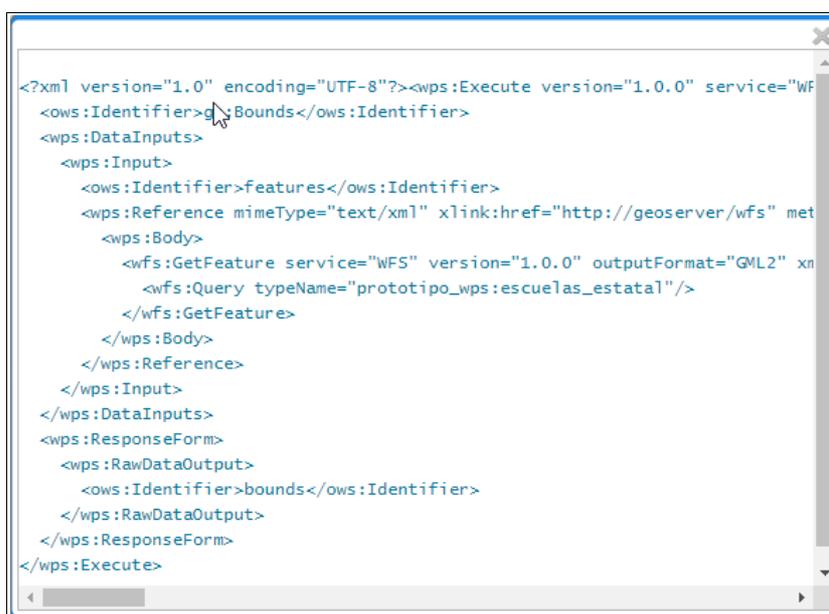


Figura 10: Proceso *gs:Bounds* - Solicitud POST WPS.

GeoServer presenta los procesos en varias categorías según el tema. Estas categorías se agrupan por prefijo:

- *geo*: procesos de geometría
- *ras*: procesos raster
- *vec*: procesos vectoriales
- *gs*: procesos específicos de GeoServer

Las versiones anteriores de GeoServer agrupaban los procesos no por tema, sino por la biblioteca interna responsable de la implementación. Los prefijos "*JTS*" y "*gt*" se pueden habilitar para preservar la compatibilidad con versiones anteriores,

o se pueden deshabilitar de manera segura debido a que su funcionalidad se clasifica correctamente en las categorías "vec" y "geo".

Los procesos de geometría (prefijo "geo") se construyen utilizando JTS Topology Suite (JTS). JTS es una biblioteca Java de funciones para procesar geometrías en dos dimensiones. JTS cumple con la especificación de características simples para SQL publicada por el OGC, similar a PostGIS. JTS incluye funciones espaciales comunes como área, zona de influencia, intersección y simplificación.

El WPS de GeoServer incluye algunos procesos creados especialmente para usar con GeoServer (prefijo "gs"). Suelen ser funciones específicas de GeoServer, como límites y reproyección. Utilizan una conexión interna al GeoServer, cómo ser a servicios WFS y WCS, que no forma parte de la especificación WPS, para leer y escribir datos.

Al igual que con los procesos "geo", los nombres y definiciones de estos procesos están sujetos a cambios. Para obtener una lista completa de los procesos específicos de GeoServer, se debe consultar el documento de capacidades (GetCapabilities) del WPS de GeoServer o navegar a través del *Generador de Consultas WPS*.

Encadenamiento de procesos

Uno de los beneficios de WPS es su capacidad nativa para encadenar procesos. Al igual que las funciones pueden llamar a otras funciones, un proceso WPS puede usar como entrada la salida de otro proceso. Por lo tanto, muchas funciones complejas se pueden combinar en una sola solicitud, para lo cual es necesario generar un XML que especifique dicho encadenamiento.

4.2.5 Sistema de Referencia de Coordenadas (SRC)

El prototipo, de acuerdo a los resultados esperados, ha sido implementado para trabajar con datos en coordenadas planas.

Con respecto a las coordenadas planas y teniendo en cuenta la ubicación geográfica de la provincia de Santa Fe, según la definición de sistemas de coordenadas y proyecciones oficiales (EPSG) de la Dirección de Geodesia del IGN³², el Sistema proyectivo Gauss-Krüger Faja 5 se corresponde con la definición de los parámetros cartográficos necesarios para asociar a la información geoespacial la proyección oficial Gauss-Krüger Faja 5 (ver Figura 11), cuyos parámetros son:

- Latitud de Origen: -90° 0' 0.00"
- Longitud del meridiano central: -60° 0' 0.00"
- Falso Norte: 0.00 m
- Falso Este: 5500000.00 m
- Factor de escala del meridiano central: 1
- Elipsoide de referencia: WGS84 (válido para POSGAR 94 y POSGAR 07. Para CAMPO INCHAUSPE 69 utilizar el elipsoide internacional de Hayford 1924)

Los códigos EPSG de esta proyección, asociada a los distintos marcos de referencia oficiales son:

- Gauss-Krüger Faja 5, POSGAR 07: 5347
- Gauss-Krüger Faja 5, POSGAR 94: 22185
- Gauss-Krüger Faja 5, CAMPO INCHAUSPE 69: 22195

La provincia de Santa Fe ha manifestado oficialmente adoptar POSGAR 07, por lo cual el prototipo trabajará para las coordenadas planas con el EPSG:5347.

Cabe destacar que los códigos EPSG relativos a SRC se han convertido en un estándar a utilizar con respecto a la IG. Dicho término deriva de la sigla "European Petroleum Survey Group", organización relacionada con la industria petrolera y del gas en el continente europeo. La cual ha desarrollado a lo largo del tiempo un repositorio de parámetros geodésicos sobre sistemas y marcos de referencia,

32 http://ramsac.ign.gob.ar/posgar07_pg_web/documentos/Informe_sobre_codigos_oficiales_EPSG.pdf

elipsoides y proyecciones cartográficas utilizados en todo el mundo. En la actualidad, los códigos EPSG son ampliamente utilizados en los SIG para trabajar con IG en formatos digitales.

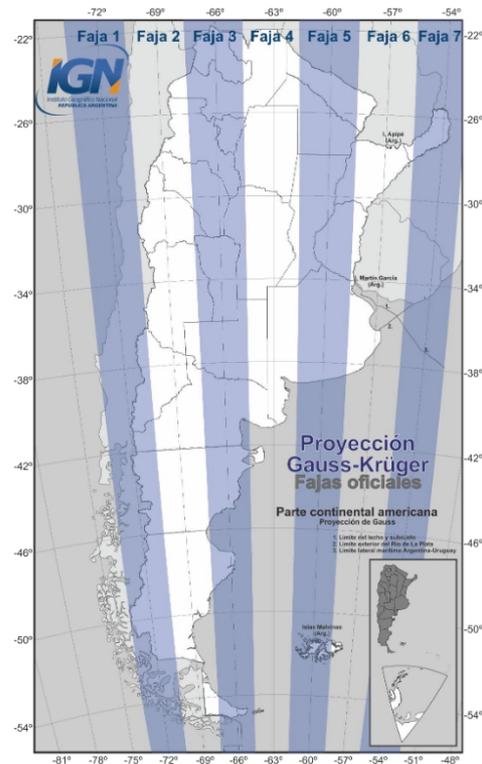


Figura 11: Proyección Gauss-Krüger.

Fuente: IGN 2017.

4.2.6 Ambiente de trabajo

La instalación del software se realizó sobre una PC con Sistema Operativo Ubuntu 18.04 LTS. Se instalaron las versiones más actuales y cercanas a la infraestructura tecnológica utilizada por la IDESF.

Herramientas, todas ellas de código abierto (Open Source), que se utilizaron en este trabajo:

- Apache Tomcat: versión Apache Tomcat/9.0.16 (Ubuntu) con JRE 8.
- GeoServer: última versión estable al momento, GeoServer 2.21.0.

- Extensión WPS para GeoServer: la versión del plugin se debe corresponder con la versión del GeoServer donde se instalará, en este caso 2.21.0.
- PostgreSQL: Versión instalada Postgresql 14.3.
- PostGIS: Versión instalada PostGIS 3.2.
- pgAdmin: Versión utilizada PGADMIN 4.29.

Clientes:

- Librería OpenLayers: versión 6.15.1.
- QGIS: última versión estable al momento, QGIS 3.22 'Białowieża' LTR.
- Plugin WPS Client de QGIS: Con este plugin se puede acceder a servicios WPS, QGIS WPS-Client 3.0.0.

GeoServer requiere que se instale un entorno Java 8 o Java 11 (JRE) en su sistema. Se instaló Java 8 para la distribución del sistema operativo Ubuntu 18.04. Entre las instalaciones existentes de GeoServer 2.21.0 se optó por la que se realiza a través de un Archivo web (.war), para ejecutar GeoServer como un servlet independiente en el servidor de aplicaciones Apache Tomcat 9, previamente instalado. Luego, se instaló la Extensión WPS de GeoServer correspondiente.

Se instaló PostgreSQL y la extensión espacial PostGIS para la creación de la base de datos espacial, la cual contiene la IG. Se utilizó el software pgAdmin como herramienta de gestión de la base de datos.

Como clientes de los servicios WPS se instaló la librería OpenLayers y el plugin WPS-Client en el software QGIS previamente instalado.

4.2.7 Recolección, preparación y publicación de la información

Luego de haber descargado las capas vectoriales, se procedió a preparar la información usando el software QGIS.

Primeramente, se tomó la capa que contiene los barrios populares de Argentina registrados en el Registro Nacional de Barrios Populares, y se generó una nueva capa seleccionando los barrios pertenecientes a las ciudades de Santa Fe (59 La Capital - Santa Fe) y Rosario (110 Rosario - Rosario), que son los de interés para el prototipo.

Luego, se editaron las capas para eliminar atributos innecesarios o irrelevantes y se re proyectaron ambas capas al Sistema de Referencia de Coordenadas (SRC) EPSG:5347 (POSGAR07 / Argentina 5)³³.

A través de PGAdmin4 se creó la Base de Datos “prototipo_wps” y se le habilitó la extensión PostGIS para soporte espacial. Se creó en la misma el esquema de trabajo “sfe_mas_conectada”. Utilizando QGIS se creó la conexión a la BD y se cargaron, en el esquema creado, las capas con SRC EPSG:5347.

Después de creada la BD, se creó un espacio de trabajo en GeoServer denominado “prototipo_wps” (URI asociado al espacio de trabajo: http://localhost:8080/prototipo_wps) y se activaron para el mismo los servicios WMS, WFS y WPS. Luego, en dicho espacio de trabajo, se creó el almacén de datos “sfe_mas_conectada” de tipo “PostGIS Database” asociada al esquema de igual nombre. De esta manera, en Geoserver aparecen listadas las capas cargadas en BD, listas para publicar.

Se publicaron ambas capas en GeoServer asignándole estilos de simbología (SLD), creados previamente en el espacio de trabajo “prototipo_wps”.

4.3 Procesos para la implementación del prototipo WPS

Teniendo en cuenta las entradas, salidas, procesamiento, modo de solicitar la ejecución de los mismos y el modo de manejar la información de salida, y en base a la finalidad del prototipo, se determinaron los procesos WPS de GeoServer a utilizar en la implementación del mismo. Se realizaron pruebas piloto sobre el ambiente de trabajo.

³³ https://ramsac.ign.gob.ar/posgar07_pg_web/documentos/Instructivo_QGis_para_aplicar_parametros.pdf

El propósito del prototipo puede ser enunciado de una manera más simple de este modo: dadas dos capas, una de puntos y otra de polígonos, obtener los puntos que caen dentro del área de influencia de 'x' metros de los polígonos.

A través de la operación *GetCapabilities* se seleccionaron procesos candidatos para la resolución del requerimiento y luego con la operación *DescribeProcess* se determinaron los procesos que definitivamente se usarían. Se revisó el contenido de los XML devueltos por dichas operaciones y se usó el *Generador de consultas WPS* como soporte. Los procesos a seleccionar eran aquellos que tienen la funcionalidad de generar buffer de polígonos y seleccionar/extraer puntos que caen dentro de polígonos.

A través de la siguiente petición *GetCapabilities*:

http://localhost:8080/geoserver/prototipo_wps/ows?service=WPS&version=1.0.0&request=GetCapabilities

se obtuvo el listado de todos los procesos que brinda el servicio WPS implementado por GeoServer (ver Anexo 2).

En la Tabla 1 se muestran los procesos candidatos y sus respectivas funcionalidades, extraído del XML devuelto por la petición WPS HTTP GET anterior:

Proceso WPS	<wps:ProcessOfferings>
Generación de Buffers	
geo:buffer	<wps:Process wps:processVersion="1.0.0"> <ows:Identifier>geo:buffer</ows:Identifier> <ows:Title>Buffer</ows:Title> <ows:Abstract>Returns a polygonal geometry representing the input geometry enlarged by a given distance around its exterior.</ows:Abstract> </wps:Process>
gs:BufferFeatureCollection	<wps:Process wps:processVersion="1.0.0"> <ows:Identifier>gs:BufferFeatureCollection</ows:Identifier> <ows:Title>Buffer</ows:Title> <ows:Abstract>Buffers features by a distance value supplied either as a parameter or by a feature attribute. Calculates buffers based on Cartesian distances.</ows:Abstract> </wps:Process>
Selección	
gs:InclusionFeatureCollection	<wps:Process wps:processVersion="1.0.0"> <ows:Identifier>gs:InclusionFeatureCollection</ows:Identifier> <ows:Title>Inclusion of Feature Collections</ows:Title> <ows:Abstract>Returns a feature collection consisting of the features from the first collection which are spatially contained in at least one feature of the second collection.</ows:Abstract>

Proceso WPS	<wps:ProcessOfferings>
	</wps:Process>
Otros de interés	
gs:CollectGeometries	<wps:Process wps:processVersion="1.0.0"> <ows:Identifier>gs:CollectGeometries</ows:Identifier> <ows:Title>Collect Geometries</ows:Title> <ows:Abstract>Collects the default geometries of the input features and combines them into a single geometry collection</ows:Abstract> </wps:Process>
gs:Import	<wps:Process wps:processVersion="1.0.0"> <ows:Identifier>gs:Import</ows:Identifier> <ows:Title>Import to Catalog</ows:Title> <ows:Abstract>Imports a feature collection into the catalog</ows:Abstract> </wps:Process>

Tabla 1: Procesos candidatos para el prototipo.

Las siguientes peticiones *DescribeProcess* devuelven archivos XML donde se especifican entradas, salidas y formatos permitidos para cada proceso en particular (ver Anexo 3):

geo:buffer

[http://localhost:8080/geoserver/prototipo_wps/ows?
service=WPS&version=1.0.0&request=DescribeProcess&identifier=geo:buffer](http://localhost:8080/geoserver/prototipo_wps/ows?service=WPS&version=1.0.0&request=DescribeProcess&identifier=geo:buffer)

gs:BufferFeatureCollection

[http://localhost:8080/geoserver/prototipo_wps/ows?
service=WPS&version=1.0.0&request=DescribeProcess&identifier=gs:BufferFeatureCollection](http://localhost:8080/geoserver/prototipo_wps/ows?service=WPS&version=1.0.0&request=DescribeProcess&identifier=gs:BufferFeatureCollection)

gs:InclusionFeatureCollection

[http://localhost:8080/geoserver/prototipo_wps/ows?
service=WPS&version=1.0.0&request=DescribeProcess&identifier=gs:InclusionFeatureCollection](http://localhost:8080/geoserver/prototipo_wps/ows?service=WPS&version=1.0.0&request=DescribeProcess&identifier=gs:InclusionFeatureCollection)

gs:CollectGeometries

[http://localhost:8080/geoserver/prototipo_wps/ows?
service=WPS&version=1.0.0&request=DescribeProcess&identifier=gs:CollectGeometries](http://localhost:8080/geoserver/prototipo_wps/ows?service=WPS&version=1.0.0&request=DescribeProcess&identifier=gs:CollectGeometries)

gs:Import

[http://localhost:8080/geoserver/prototipo_wps/ows?
service=WPS&version=1.0.0&request=DescribeProcess&identifier=gs:Import](http://localhost:8080/geoserver/prototipo_wps/ows?service=WPS&version=1.0.0&request=DescribeProcess&identifier=gs:Import)

Luego de analizar las respuestas de la operación *DescribeProcess* de cada proceso analizado y realizar pruebas de los mismos a través del *Generador de consultas WPS*, se descartaron los procesos *geo:buffer* y *gs:CollectGeometries*. El primero de ellos se dejó de lado porque sólo genera el área de influencia de una sola geometría. En cuanto al proceso *gs:CollectGeometries* se lo había tomado como candidato, por si era necesario utilizarlo para evitar elementos repetidos al realizar la selección de las escuelas que se encuentran dentro del área de influencia de los barrios populares, pero a través de las pruebas realizadas se concluyó que no hacía falta su uso.

Quedando así seleccionados los siguientes procesos que cumplían con lo requerido para la implementación del prototipo:

- *gs:BufferFeatureCollection*
- *gs:InclusionFeatureCollection*
- *gs:import*

El proceso *gs:BufferFeatureCollection* permite como entrada una capa de GeoServer (sea de puntos, líneas o polígonos), en este caso la capa de polígonos correspondiente a los barrios populares, la distancia que puede ser un valor arbitrario o bien un campo de la tabla asociada a la capa seleccionada. La capa se toma como un servicio WFS. Al procesar nos permite obtener una capa de polígonos correspondiente a las áreas de influencia. Para probar la operación se utilizó la capa “prototipo_wps:barriospopulares_stafeyrosario”, como muestra la Figura 12.

Generador de consultas WPS

Generador de consultas WPS paso a paso.

Elija proceso

gs:BufferFeatureCollection

Buffers features by a distance value supplied either as a parameter or by a feature attribute. Calculates buffers based on Cartesian distances. (Proces de descripción WPS)

Entradas para el proceso

features* - SimpleFeatureCollection
Input feature collection

VECTOR_LAYER ▾ prototipo_wps:barriospopulares_stafeyrosario ▾

distance* - Double
Fixed value to use for the buffer distance

1000

attributeName - String
Attribute containing the buffer distance value

Salidas del proceso

result* - SimpleFeatureCollection
Buffered feature collection

Generate text/xml; subtype=wfs-collection/1.0 ▾

Identificación

identificar (de otro modo se ejecutará la solicitud como anónima)

Ejecute el proceso **Generar XML de los procesos entrada/salida**

Figura 12: Consulta WPS gs:BufferFeatureCollection.

XML del proceso:

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:wps="http://www.opengis.net/wps/2.0" xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:wfs="http://www.opengis.net/wfs/1.1">
  <ows:Identifier>gs:BufferFeatureCollection</ows:Identifier>
  <wps>DataInputs>
    <wps:Input>
      <ows:Identifier>features</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2" xmlns:prototipo_wps="http://prototipo_wps:barriospopulares_stafeyrosario">
            <wfs:Query typeName="prototipo_wps:barriospopulares_stafeyrosario"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>distance</ows:Identifier>
      <wps>Data>
        <wps:LiteralData>1000</wps:LiteralData>
      </wps>Data>
    </wps:Input>
  </wps>DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

Figura 13: XML del proceso gs:BufferFeatureCollection.

El proceso *gs:InclusionFeatureCollection*, al igual que el proceso anterior, permite como entradas capas WFS de GeoServer. Al procesar nos permite obtener una capa, que contiene elementos de la primera capa ingresada que están contenidas en al menos un elemento de la segunda capa de entrada. Para la prueba se tomaron como entradas las capas “prototipo_wps:escuelas_estatal” y “prototipo_wps:barriospopulares_stafeyrosario”, en ese orden de ingreso (ver Figura 14).

The screenshot shows the 'Generador de consultas WPS' interface. It is titled 'Generador de consultas WPS' and has a subtitle 'Generador de consultas WPS paso a paso.' The main section is 'Elija proceso', where 'gs:InclusionFeatureCollection' is selected in a dropdown menu. Below this, a description reads: 'Returns a feature collection consisting of the features from the first collection which are spatially contained in at least one feature of the second collection. (Proces de descripción WPS)'. The 'Entradas para el proceso' section contains two input fields: 'first*' - SimpleFeatureCollection (First feature collection) with 'VECTOR_LAYER' and 'prototipo_wps:escuelas_estatal', and 'second*' - SimpleFeatureCollection (Second feature collection) with 'VECTOR_LAYER' and 'prototipo_wps:barriospopulares_stafeyrosario'. The 'Salidas del proceso' section shows 'result*' - SimpleFeatureCollection (Output feature collection) with a checked 'Generate' option and a dropdown set to 'text/xml; subtype=wfs-collection/1.0'. The 'Identificación' section has an unchecked 'identificar' checkbox. At the bottom, there are two buttons: 'Ejecute el proceso' and 'Generar XML de los procesos entrada/salida'.

Figura 14: Consulta WPS *gs:InclusionFeatureCollection*.

XML del proceso:

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:wps="http://www.opengis.net/wps/2.0" xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:wcs="http://www.opengis.net/wcs/1.1" xmlns:wfs="http://www.opengis.net/wfs/1.1">
  <ows:Identifier>gs:InclusionFeatureCollection</ows:Identifier>
  <wps>DataInputs>
    <wps:Input>
      <ows:Identifier>first</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2" xmlns:prototipo_wps="http://prototipo.wfs.org/1.0.0">
            <wfs:Query typeName="prototipo_wps:escuelas_estatal"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>second</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
        <wps:Body>
          <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2" xmlns:prototipo_wps="http://prototipo.wfs.org/1.0.0">
            <wfs:Query typeName="prototipo_wps:barriospopulares_stafeyrosario"/>
          </wfs:GetFeature>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
  </wps>DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

Figura 15: XML del proceso *gs:InclusionFeatureCollection*.

El proceso *gs:import*, es muy interesante, permite importar tanto datos vectoriales como raster y publicarlos como una capa en GeoServer. Se debe especificar el espacio de trabajo, el almacén de datos, nombre de la nueva capa, SRC de la misma, estilo de la simbología a aplicar.

Como prueba se generó una copia de la capa “prototipo_wps:escuelas_estatal” en el mismo espacio de trabajo (ver Figura 16). Cabe aclarar, que al ejecutar el proceso daba error si en el parámetro “Input Raster” no había nada seleccionado, por lo que se eligió arbitrariamente una de las opciones del listado desplegable, y se logró la incorporación a GeoServer de la capa nueva, que se denominó “prototipo_wps:escuelas_estatal_backup”. Observando, que en el esquema “sfe_mas_conectada” de la BD de trabajo se creó automáticamente la tabla correspondiente para esta capa, copia de la original.

Generador de consultas WPS

Generador de consultas WPS paso a paso.

Elija proceso

gs:Import

Imports a feature collection into the catalog (Proces de descripción WPS)

Entradas para el proceso

features - SimpleFeatureCollection
Input feature collection

VECTOR_LAYER ▾ prototipo_wps:escuelas_estatal ▾

coverage - GridCoverage2D
Input raster

RASTER_LAYER ▾ nunc:Arc_Sample ▾

workspace - String
Target workspace (default is the system default)

prototipo_wps

store - String
Target store (default is the workspace default)

sfe_mas_conectada

name - String
Name of the new featuretype/coverage (default is the name of the features in the collection)

escuelas_estatal_backup

srs - CoordinateReferenceSystem
Target coordinate reference system (default is based on source when possible)

EPSG:5347 EPSG:POSGAR 2007 / Argentina 5...

srsHandling - ProjectionPolicy
Desired SRS handling (default is FORCE_DECLARED, others are REPROJECT_TO_DECLARED or NONE)

FORCE_DECLARED ▾

styleName - String
Name of the style to be associated with the layer (default is a standard geometry-specific style)

escuelas_estatal

Salidas del proceso

layerName* - String
Name of the new featuretype, with workspace

Generate

Identificación

identificar (de otro modo se ejecutará la solicitud como anónima)

Figura 16: Consulta WPS gs:Import.

XML del proceso:

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS" xmlns:xsi="http://www.
<ows:Identifier>gs:Import</ows:Identifier>
<wps>DataInputs>
  <wps:Input>
    <ows:Identifier>features</ows:Identifier>
    <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
      <wps:Body>
        <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2" xmlns:prototipo_wps="http://
        <wfs:Query typeName="prototipo_wps:escuelas_estatal"/>
        </wfs:GetFeature>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>coverage</ows:Identifier>
    <wps:Reference mimeType="image/tiff" xlink:href="http://geoserver/wcs" method="POST">
      <wps:Body>
        <wcs:GetCoverage service="WCS" version="1.1.1">
          <ows:Identifier>nurc:Arc_Sample</ows:Identifier>
          <wcs:DomainSubset>
            <ows:BoundingBox crs="http://www.opengis.net/gml/srs/epsg.xml#4326">
              <ows:LowerCorner>-180.0 -90.0</ows:LowerCorner>
              <ows:UpperCorner>180.0 90.0</ows:UpperCorner>
            </ows:BoundingBox>
          </wcs:DomainSubset>
          <wcs:Output format="image/tiff"/>
        </wcs:GetCoverage>
      </wps:Body>
    </wps:Reference>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>workspace</ows:Identifier>
    <wps>Data>
      <wps:LiteralData>prototipo_wps</wps:LiteralData>
    </wps>Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>store</ows:Identifier>
    <wps>Data>
      <wps:LiteralData>sfe_mas_conectada</wps:LiteralData>
    </wps>Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>name</ows:Identifier>
    <wps>Data>
      <wps:LiteralData>escuelas_estatal_backup</wps:LiteralData>
    </wps>Data>
  </wps:Input>
  <wps:Input>
    <ows:Identifier>srs</ows:Identifier>
    <wps>Data>
```

Figura 17: XML del proceso gs:Import.

Previsualización de la capa desde GeoServer a través del servicio WMS (formato OpenLayers):

http://localhost:8080/geoserver/prototipo_wps/wms?service=WMS&version=1.1.0&request=GetMap&layers=prototipo_wps:escuelas_estatal_backup&bbox=5250458.5,6194514.5,5579956.0,6900278.0&width=358&height=768&srs=EPSG:5347&styles=&format=application/openlayers

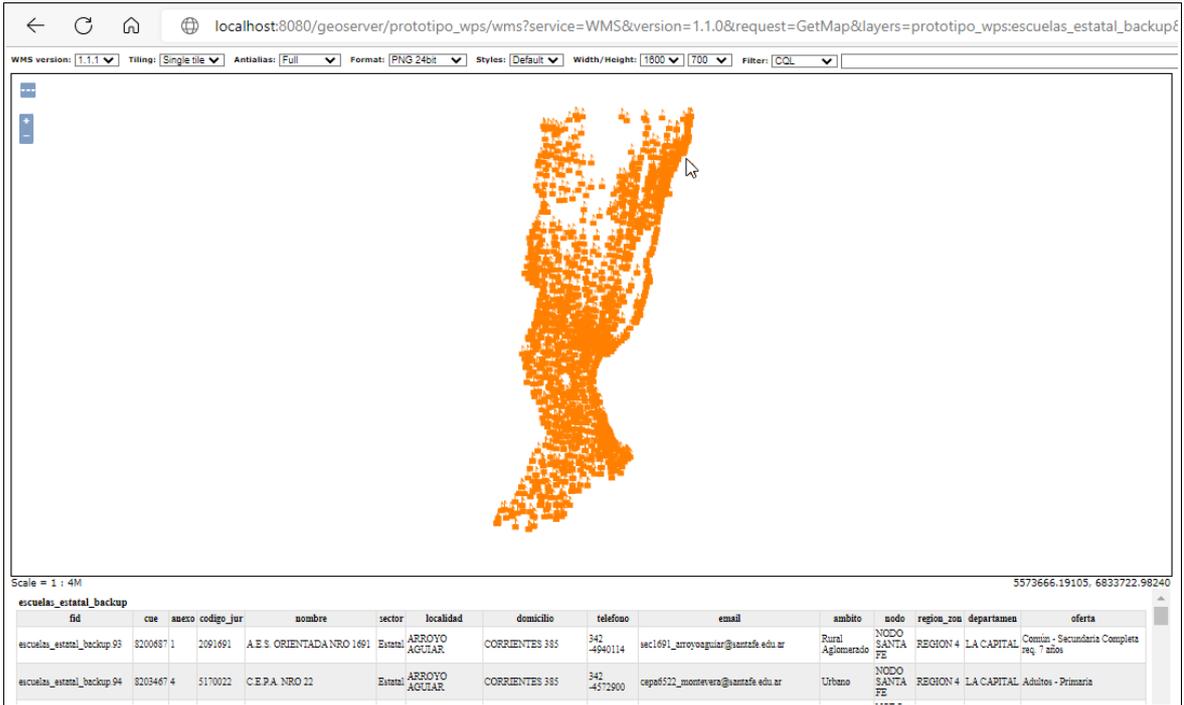


Figura 18: Previsualización de la capa.

Una funcionalidad extra para el requerimiento, era agregar al resultado el nombre del barrio más cercano a cada escuela y la distancia al mismo, para priorizar por distancia el acceso a cobertura WiFi para los establecimientos escolares. Es decir, calcular la distancia del punto (escuela) al polígono más cercano (barrio).

Con respecto a este cálculo, dentro de los procesos WPS disponibles de GeoServer, como se aprecia en la Tabla 2, ninguno permite realizar la operación entre dos capas, por lo cual no se contempló este cálculo para el prototipo.

Proceso WPS para calcular distancias	<wps:ProcessOfferings>
geo:distance	<wps:Process wps:processVersion="1.0.0"> <ows:Identifier>geo:distance</ows:Identifier> <ows:Title>Distance</ows:Title> <ows:Abstract>Returns the minimum distance between two geometries. Measurement is given in the input units, so geographic coordinates are not recommended.</ows:Abstract> </wps:Process>
gs:Nearest	<wps:Process wps:processVersion="1.0.0"> <ows:Identifier>gs:Nearest</ows:Identifier> <ows:Title>Nearest Feature</ows:Title> <ows:Abstract>Returns the feature in a given feature collection that has the smallest distance to a given point.</ows:Abstract> </wps:Process>
gs:Snap	<wps:Process wps:processVersion="1.0.0"> <ows:Identifier>gs:Snap</ows:Identifier> <ows:Title>Snap</ows:Title> <ows:Abstract>Returns the feature in a feature collection nearest to a given point. Attributes for distance and bearing are added.</ows:Abstract> </wps:Process>

Tabla 2: Procesos para calcular distancias

5. RESULTADOS

5.1 Prototipo WPS en el marco de la IDESF

El resultado final del proyecto es el prototipo WPS, en el marco de la IDESF, implementado para obtener las escuelas estatales provinciales que se encuentran dentro del área de influencia de los barrios populares, de las ciudades de Santa Fe y Rosario, para lograr que todas ellas tengan conexión WiFi.

Con los procesos WPS elegidos en el apartado 6.3 se implementó el prototipo. A través del encadenamiento de procesos, se logró generar un proceso WPS más complejo para llegar al resultado.

Con la ayuda del *Generador de consultas WPS* se generó el XML necesario para la ejecución de los procesos en cadena, donde cada uno recibe como entrada el resultado de la ejecución del proceso anterior. Esta tarea hubiese sido más sencilla si la generación de subprocesos en el *Generador de consultas WPS* funcionara. Desde unas versiones anteriores dejó de funcionar³⁴ y según lo investigado al respecto hay pocas razones para arreglarlo, debido a que la herramienta para generar consultas es una demo y se usa para pruebas. Por lo tanto, se fue generando el XML por partes y luego se fue ensamblando hasta obtener el definitivo.

El orden en el que se ejecutan los procesos es muy importante. Primeramente, a través del geoproceso *gs:BufferFeatureCollection* seleccionando como entrada la capa de barrios populares (“prototipo_wps:barriospopulares_stafeyrosario”) se obtuvo como resultado los polígonos correspondientes a las áreas de influencia de dicha capa. Luego, a través del proceso WPS *gs:InclusionFeatureCollection*, se tomó como entradas la capa obtenida en el paso anterior y las escuelas estatales (“prototipo_wps:escuelas_estatal”), obteniendo así aquellas escuelas estatales que se encuentran dentro de las áreas de influencia. Finalmente, el geoproceso *gs:import* incorpora y publica en GeoServer la capa resultante “escuelas_resultado” en el espacio de trabajo “prototipo_wps”.

34 <https://osgeo-org.atlassian.net/browse/GEOS-7967>

En el XML del prototipo (ver Anexo 4), que contiene el encadenamiento de procesos necesarios para llegar al resultado y agregarlo al catálogo de GeoServer, se aprecia lo siguiente:

- Las capas se toman a través del servicio WFS desde el espacio de trabajo *prototipo_wps* de GeoServer:

```
<wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:prototipo_wps="http://localhost:8080/geoserver/prototipo_wps">
  <wfs:Query srsName="EPSG:5347" typeName="prototipo_wps:escuelas_estatal"/>
</wfs:GetFeature>
```

```
<wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:prototipo_wps="http://localhost:8080/geoserver/prototipo_wps">
  <wfs:Query srsName="EPSG:5347"
typeName="prototipo_wps:barriospopulares_stafeyrosario"/>
</wfs:GetFeature>
```

- El proceso WPS *gs:BufferFeatureCollection*

```
<ows:Identifier>vec:BufferFeatureCollection</ows:Identifier>
```

toma como entrada la capa "barriospopulares_stafeyrosario"

y devuelve la salida como una wfs-collection basada en XML:

```
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
  <ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
```

- El proceso WPS *gs:InclusionFeatureCollection*

```
<ows:Identifier>vec:InclusionFeatureCollection</ows:Identifier>
```

toma como entrada la capa "escuelas_estatal" y la salida del proceso *gs:BufferFeatureCollection*, y devuelve la salida como una wfs-collection basada en XML:

```
<wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
  <ows:Identifier>result</ows:Identifier>
</wps:RawDataOutput>
```

Cuya salida es una de las entradas al proceso WPS *gs:Import*

```
<ows:Identifier>gs:Import</ows:Identifier>
```

junto a estas otras entradas requeridas, que indican los parámetros necesarios para publicar la capa en GeoServer:

```
<wps:Input>
  <ows:Identifier>workspace</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>prototipo_wps</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>store</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>sfe_mas_conectada</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>name</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>escuelas_ai_resultado</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>srs</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>EPSG:5347</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>srsHandling</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>FORCE_DECLARED</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>styleName</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>escuelas_resultado</wps:LiteralData>
  </wps:Data>
</wps:Input>
```

5.2 Clientes WPS

Este apartado trata la compatibilidad de las herramientas, ya adoptadas por la IDESF, OpenLayers (librería cliente Web) y QGIS (software SIG de escritorio) como clientes WPS del servidor GeoServer.

5.2.1 Integración con OpenLayers como cliente ligero WPS

Haciendo uso del XML del encadenamiento de procesos, se integró el prototipo WPS en OpenLayers. Se agregó este requerimiento en una aplicación web OpenLayers mínima para testear los resultados de la solicitud HTTP POST con XML.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>IDESF Prototipo WPS OL</title>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content=
    "width=device-width, initial-scale=1, shrink-to-fit=no">
  <!-- OL stylesheet -->
  <link rel="stylesheet" href="../openlayers6/css/ol.css" type="text/css">
  <link rel="stylesheet" href="https://raw.githack.com/walkermatt/ol-layerswitcher/master/src/ol-
layerswitcher.css"/>
</head>
<body>
  ...
  <!-- OpenLayers JS dependency debug build and proj4js for projection -->
  <script type="text/javascript" src="../openlayersv6/build/ol.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/proj4js/2.6.1/proj4.js"></script>
  <script src="https://raw.githack.com/walkermatt/ol-layerswitcher/master/dist/ol-
layerswitcher.js"></script>

  <!-- Begin wps-ol Javascript -->
  <script type="text/javascript">
```

El SRC utilizado es el EPSG:5347:

```
proj4.defs('EPSG:5347', '+proj=tmerc +lat_0=-90 +lon_0=-60 +k=1 +x_0=5500000 +y_0=0
+ellps=WGS84 +units=m +no_defs');
ol.proj.proj4.register(proj4);
var proj5347 = ol.proj.get('EPSG:5347');
proj5347.setExtent([5187300, 6146880, 5688720, 6935760]);
```

En una variable de JavaScript se guarda el XML del prototipo WPS, explicado en el apartado anterior (ver Anexo 5).

El envío de la solicitud HTTP POST con el XML al servidor, se hace a través del objeto JavaScript XMLHttpRequest (estándar de la W3C³⁵). Una vez que la transacción se ha completado, el objeto contendrá información útil tal como el cuerpo de la respuesta y el estado HTTP status del resultado.

```
url = host + '/wps';
var req = new XMLHttpRequest();
req.open("POST", url, true);
req.setRequestHeader('Content-type', 'text/xml');
req.onreadystatechange = function() {
  if (req.readyState != 4) return;
  if (req.status != 200 && req.status != 304) {
    alert('HTTP error ' + req.status);
    return;
  }
  ...
  if (req.readyState == 4) return;
};
req.send(postData);
```

Al abrir la aplicación por primera vez se visualiza la capa base que contiene los departamentos de la provincia de Santa Fe (Figura 19), accedida desde el servicio WMS de la IDESF³⁶.

35 <https://www.w3.org/TR/XMLHttpRequest/>

36 <https://aswe.santafe.gov.ar/idesf/wms>

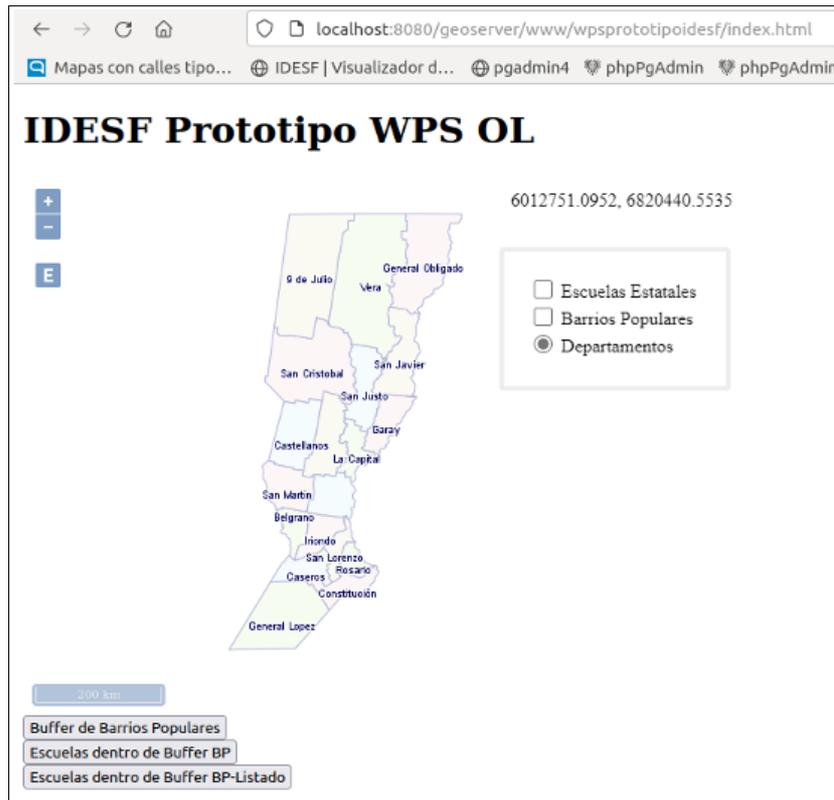


Figura 19: Prototipo WPS en OpenLayers – Capa base.

Las capas de partida están disponibles para ser visualizadas, a través del servicio WMS de GeoServer³⁷ (Figura 20).

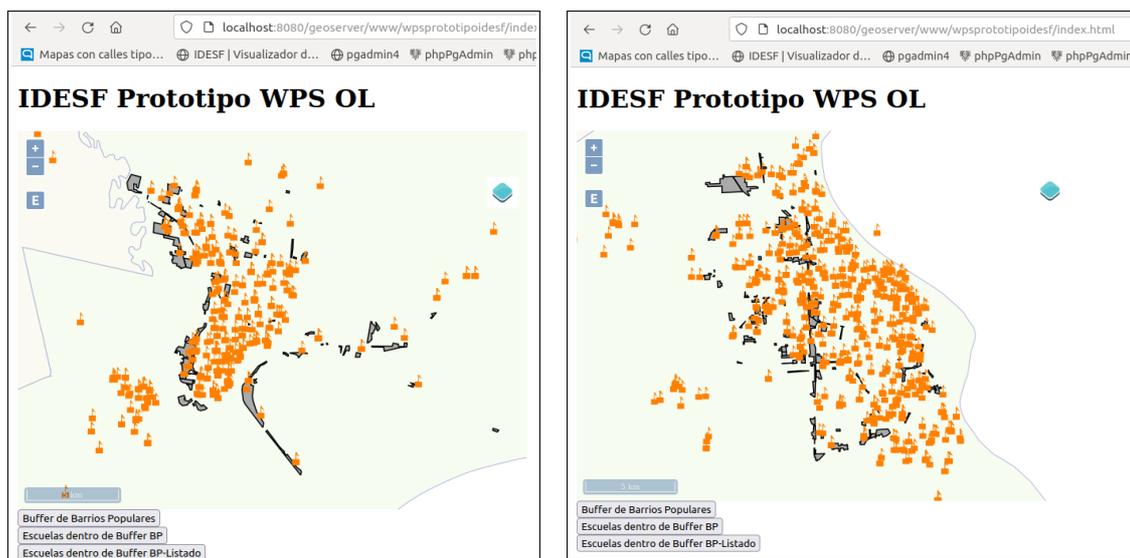


Figura 20: Prototipo WPS en OpenLayers – Capas de partida.

37 http://localhost:8080/geoserver/prototipo_wps/wms

Al hacer clic en el botón “Escuelas dentro de Buffer BP”, se enviará la solicitud WPS que está en formato XML y se publicará la capa resultante en GeoServer. La cual es agregada al mapa a través del servicio WMS de GeoServer (Figura 21).

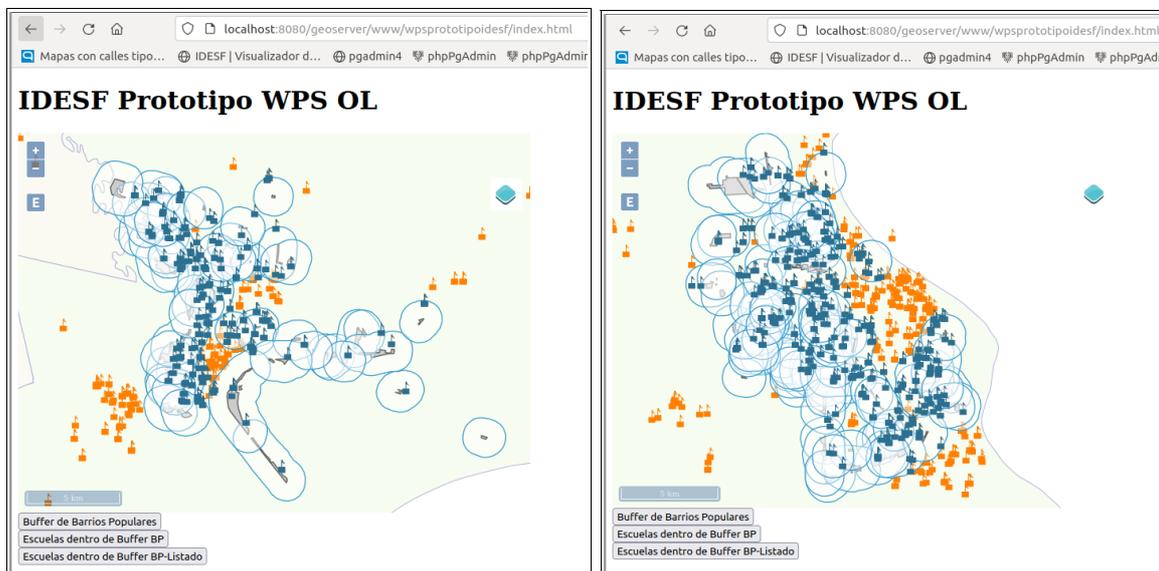


Figura 21: Prototipo WPS en OpenLayers – Escuelas dentro de Buffer BP.

Entonces, utilizando OpenLayers como cliente ligero WPS a través del prototipo de interés se obtienen y se visualizan, en color azul, las escuelas que caen dentro del área de influencia de 1000m de los barrios populares de las ciudades de Santa Fe y Rosario, como se ve en la Figura 21.

En este caso la capa resultante se incorpora y publica en GeoServer a través del proceso *gs:import* y luego se agrega al mapa desde GeoServer. Pero también podría brindarse el resultado simplemente como un listado, usando como formato de salida “text/csv” (`<wps:RawDataOutput mimeType="text/csv">`). Al presionar el botón “Escuelas dentro de Buffer BP-Listado” se obtiene el resultado, como se muestra en la Figura 22.

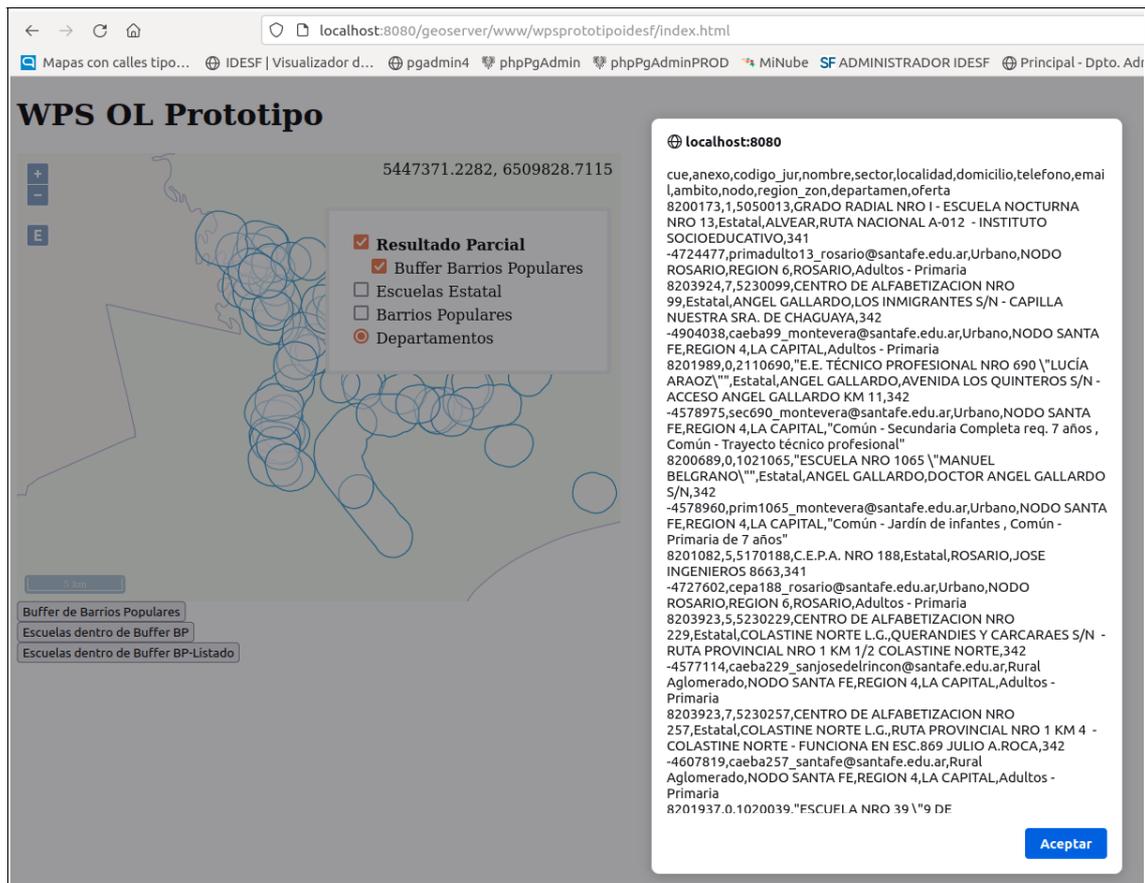


Figura 22: Prototipo WPS en OpenLayers – Escuelas dentro de Buffer BP-Listado.

En cuanto a la capa correspondiente al área de influencia de los barrios populares no es necesaria conservarla, por tratarse de una capa intermedia que se usa para obtener el resultado final, pero sí es interesante visualizarla en superposición con las otras capas. Al seleccionar el botón “*Buffer de Barrios Populares*” se ejecuta el proceso para generar los buffers, la respuesta de la petición (en formato `<wps:RawDataOutput mimeType="application/json">`) se procesa en la aplicación OpenLayers y se agrega al mapa (Figura 23).

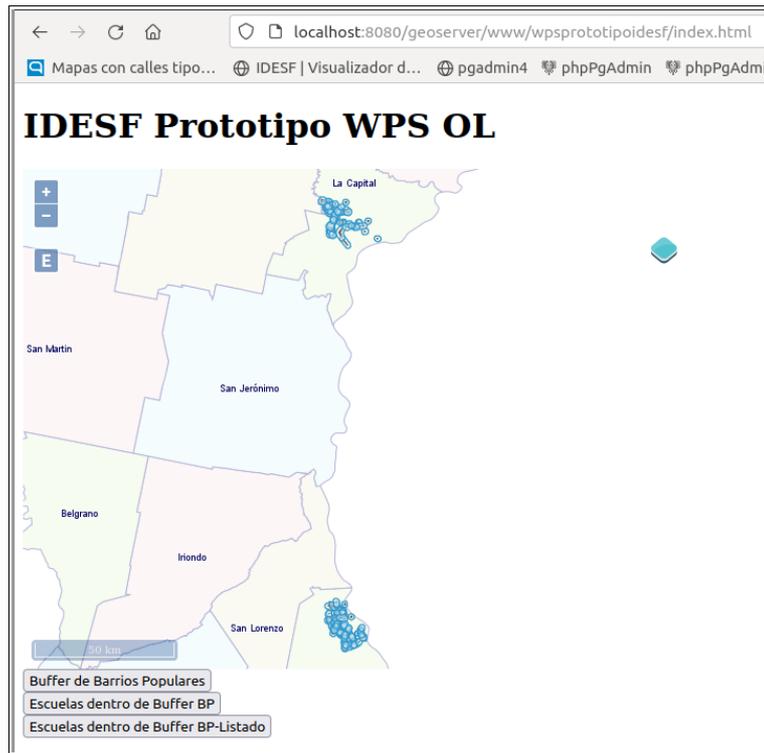


Figura 23: Prototipo WPS en OpenLayers – Buffer de Barrios Populares.

5.2.2 QGIS como cliente pesado WPS

El complemento *WPS Client* de QGIS, permite conectar este software SIG con servidores WPS para ejecutar procesos.

Se probó la última versión disponible del complemento (3.0.0) en QGIS 3.22. Al ejecutar el plugin nos permite conectar al servidor WPS, listando los procesos que dicho servidor brinda:

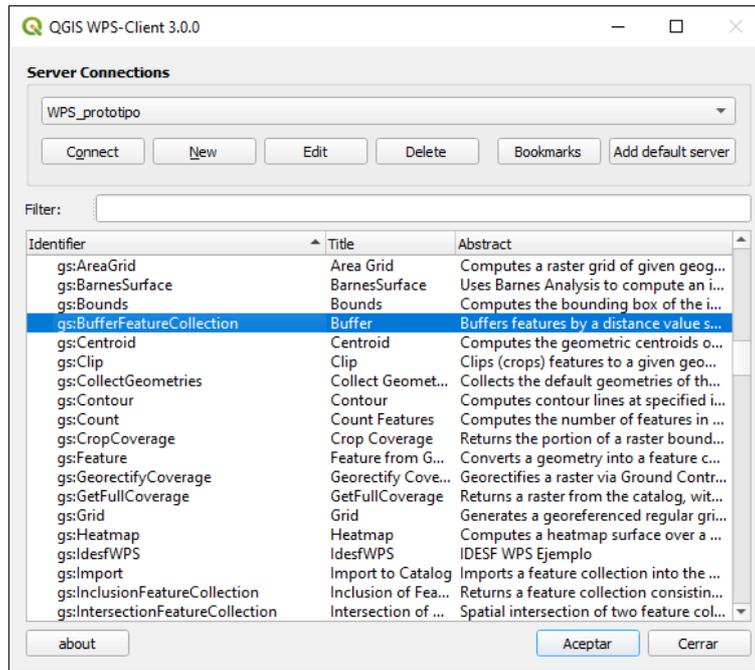


Figura 24: QGIS – Plugin WPS-Client.

Al seleccionar el proceso se presenta la interfaz donde suministrar los parámetros necesarios para realizar el proceso:

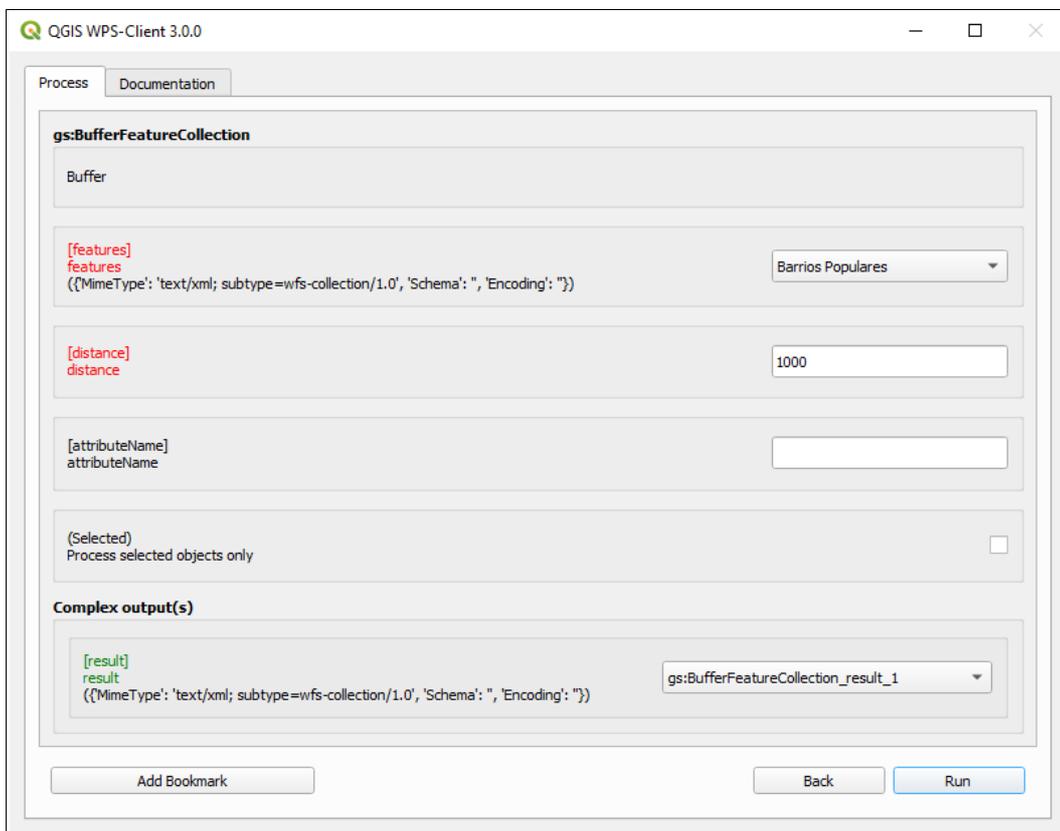


Figura 25: QGIS – Plugin WPS-Client – Proceso gs:BufferFeatureCollection.

Al querer ejecutar el proceso WPS “*gs:BufferFeatureCollection*”, el cual se pudo procesar sin ningún problema desde el *Generador de consultas WPS* y desde OpenLayers, utilizando este plugin de QGIS se detectaron inconvenientes.

Se observa que la capa a utilizar como entrada del proceso debe estar cargada y visible en QGIS para ser listada y poder seleccionarla. Otra observación, como se aprecia en la Figura 25, es que no presenta la opción de elegir el formato de salida del proceso. Presenta como formato predeterminado: (`{'MimeType': 'text/xml; subtype=wfs-collection/1.0', 'Schema': "", 'Encoding': }`).

Al ejecutar el proceso se obtiene un error referido a la codificación de caracteres, como se muestra en la Figura 26:

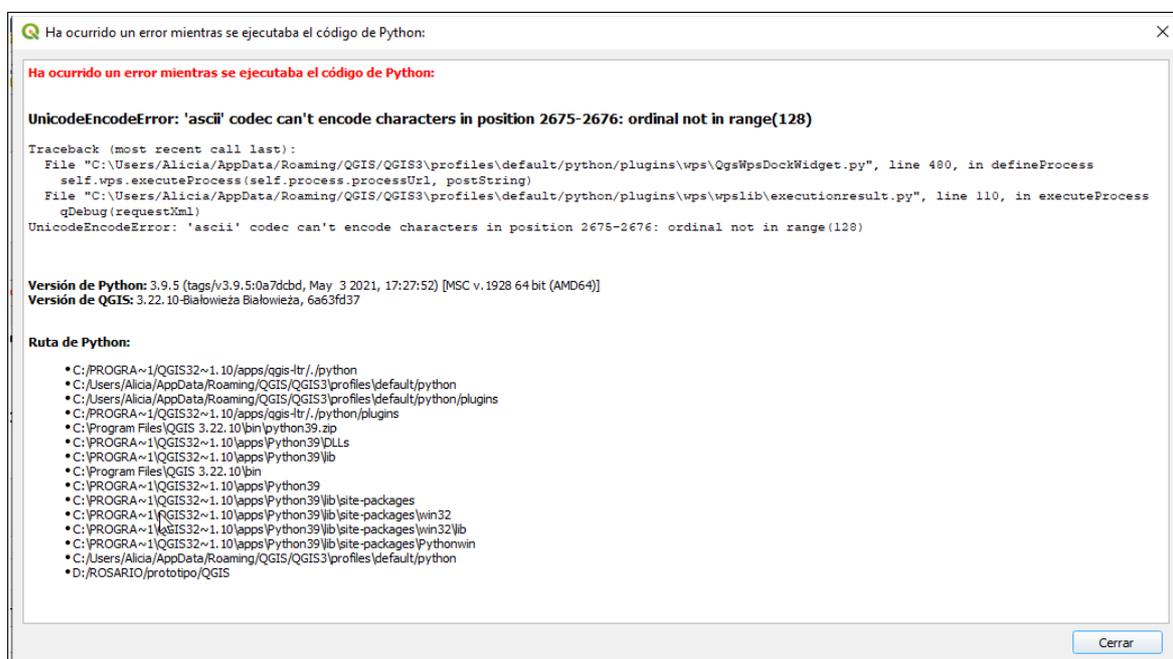


Figura 26: QGIS – Plugin WPS-Client – Mensaje de error.

Como prueba, se editó la capa de barrios populares sacando las columnas que presentaban caracteres especiales (como acentos y ñ) y se volvió a ejecutar el proceso. En este caso se obtuvo el siguiente mensaje de advertencia de python al devolver el resultado del proceso:

Advertencia de python

WARNING

```
warning:C:\Users/Alicia/AppData/Roaming/QGIS/QGIS3\profiles\default/python/plugins\wps\wpslib\execution
request.py:166: DeprecationWarning: QgsVectorFileWriter.writeAsVectorFormat() is deprecated
error, msg = QgsVectorFileWriter.writeAsVectorFormat(vLayer, tmpFile, encoding,
vLayer.dataProvider().crs(), "GML", processSelected, dso, lco)
traceback: File
"C:\Users/Alicia/AppData/Roaming/QGIS/QGIS3\profiles\default/python/plugins\wps\QgsWpsDockWidget.py"
, line 372, in defineProcess
gmldata = createTmpGML(self.tools.getVLayer(comboBox.currentText()),
File
"C:\Users/Alicia/AppData/Roaming/QGIS/QGIS3\profiles\default/python/plugins\wps\wpslib\executionrequest.
py", line 166, in createTmpGML
error, msg = QgsVectorFileWriter.writeAsVectorFormat(vLayer, tmpFile, encoding,
vLayer.dataProvider().crs(), "GML", processSelected, dso, lco)
```

Se puede concluir que el plugin no estaría funcionando satisfactoriamente, al menos para la última versión LTR de QGIS. Cabe aclarar, que no se presentaron problemas al probar procesos que devuelven un texto o un número.

5.3 Incorporación de nuevos procesos WPS

Se investigó la factibilidad de crear e incorporar nuevos procesos WPS a los ya integrados en GeoServer, en su documentación explica cómo implementar un proceso WPS nuevo. Es una clase Java que proporciona un método *execute*, este método acepta parámetros que corresponden a los parámetros de WPS y devuelve un valor que se convierte en la salida del proceso. La clase también proporciona metadatos para especificar los nombres y descripciones del proceso, y sus parámetros. La librería GeoTools utilizada por GeoServer proporciona un framework (estructura) necesario para crear clases de procesos con un mínimo de código redundante.

En este apartado se describe la creación de un proceso WPS personalizado en Java, su implementación y ejecución en GeoServer. El proceso de ejemplo utilizado es un proceso simple "*Hola Mundo*" que acepta un único parámetro de entrada y devuelve una única salida de texto.

Creación del proyecto Java

Para crear un proceso WPS personalizado, se debe crear un nuevo proyecto Java, ya sea con un IDE o utilizando herramientas de línea de comandos.

Para crear el proyecto Java para el proceso se utilizó Maven, que generó la estructura del directorio del proyecto (wps_demo_idesf), el archivo pom.xml que describe el proyecto y algunos artefactos Java útiles.

Al archivo pom.xml se modificó agregando las dependencias de la librería de GeoTools y de GeoServer; las versiones de GeoServer y Geotools; las referencias a los repositorios de Boundless Maven y OSGeo, lo que proporcionará acceso a las librerías requeridas. (Ver Anexo 6).

Con la definición del proyecto y las dependencias seteadas, se creó el proyecto Eclipse a través del siguiente comando: `mvn eclipse:eclipse`

El proyecto quedó con la siguiente estructura:

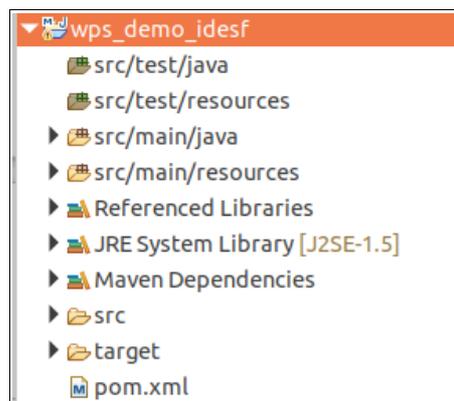


Figura 27: Estructura del proyecto.

Creación de la clase del proceso

Se creó el paquete denominado org.ejemplo.wps dentro de la estructura del directorio src/main/java. Dentro de dicho paquete se creó la clase Java IdesfWPS.java que implementará el proceso WPS personalizado.

```

package org.ejemplo.wps;

import org.geotools.process.factory.DescribeParameter;
import org.geotools.process.factory.DescribeProcess;
import org.geotools.process.factory.DescribeResult;
import org.geoserver.wps.gs.GeoServerProcess;

@DescribeProcess(title="IdesfWPS", description="IDESF WPS Ejemplo")
public class IdesfWPS implements GeoServerProcess {
    @DescribeResult(name="result", description="output result")
    public String execute(@DescribeParameter(name="name", description="name to return") String name) {
        return "Hola " + name;
    }
}

```

El método *execute* será llamado cuando GeoServer procese la solicitud WPS. El método toma un parámetro de tipo *String* y devuelve también un *String*.

Se utilizan *anotaciones* para describir el proceso en su publicación. Estos contienen descripciones del proceso, y sus entradas y salidas, que se expondrán a través de las solicitudes WPS *DescribeProcess* y *GetCapabilities*.

La anotación *DescribeProcess* proporciona la descripción del proceso para la solicitud *DescribeProcess*. La anotación *DescribeResult* proporciona la descripción de la salida del proceso. La anotación *DescribeParameter* proporciona descripciones de las entradas.

Registro del proceso en GeoServer

GeoServer utiliza Spring Framework³⁸ para gestionar la creación de instancias de componentes. Este mecanismo se utiliza para registrar el proceso WPS en GeoServer cuando inicia.

Se creó el directorio */src/main/resources* en la raíz del nuevo módulo y dentro de él se incluyó el archivo *applicationContext.xml* con el siguiente contenido:

38 <https://spring.io/projects/spring-framework/>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">
<beans>
  <bean id="IdesfWPS" class="org.ejemplo.wps.IdesfWPS"/>
</beans>
```

El proceso registrado en el contexto spring de GeoServer se asignará al espacio de nombres de proceso “gs”. El nombre completo del proceso será *gs:IdesfWPS*.

Compilación e implementación

Para generar el proceso personalizado, se ejecutó el siguiente comando desde el directorio raíz del proyecto: *mvn clean install*

Esto limpia el área de compilación, compila el código y crea un archivo JAR en el directorio *target* (Figura 28). El nombre de archivo JAR está determinado por el nombre y la versión asignados al proyecto en el archivo *pom.xml* (*wps_demo_idesf-0.0.1-SNAPSHOT.jar*).

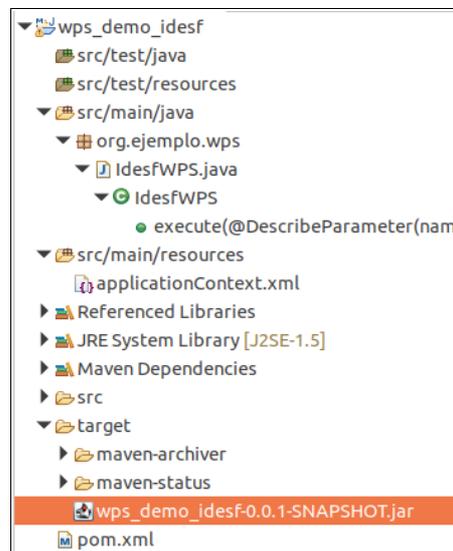


Figura 28: Archivo JAR del proceso.

Para implementar el módulo del proceso, se copió este archivo JAR en el directorio */WEB-INF/lib* de GeoServer (Figura 29) y, a continuación, se reinició GeoServer.

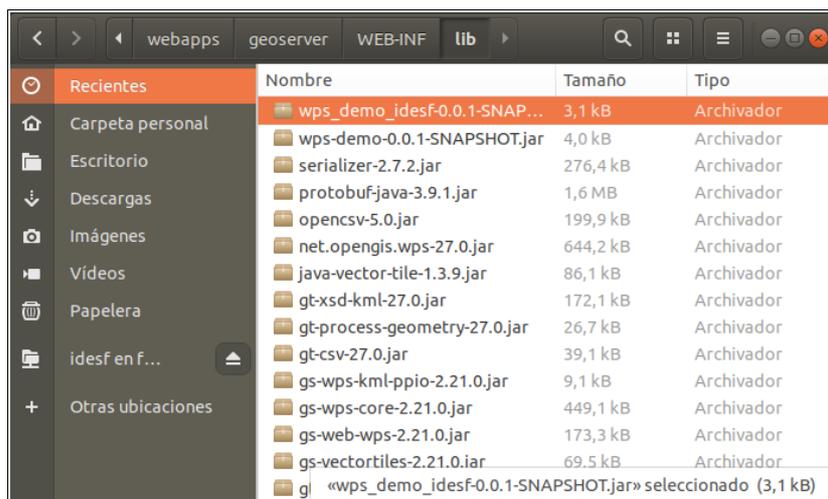


Figura 29: Archivo JAR del proceso en GeoServer.

Prueba

Para comprobar que el nuevo proceso se implementó correctamente se utilizó el *Generador de consultas WPS*. En el menú desplegable “Elija proceso” se verificó que estaba agregado el proceso personalizado llamado *gs:IdesfWPS*.

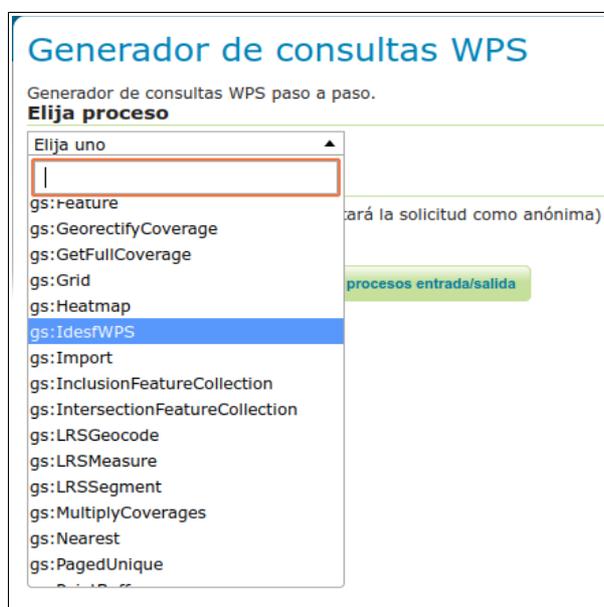


Figura 30: Generador de consultas WPS – Proceso *gs:IdesfWPS*.

Al seleccionarlo se muestra la interfaz asociada a dicho proceso, que permite ejecutarlo en función de los parámetros y salidas descritos en las capacidades del proceso.

La Figura 31 muestra el *Generador de solicitudes WPS* ejecutando el proceso *gs:IdesfWPS*.

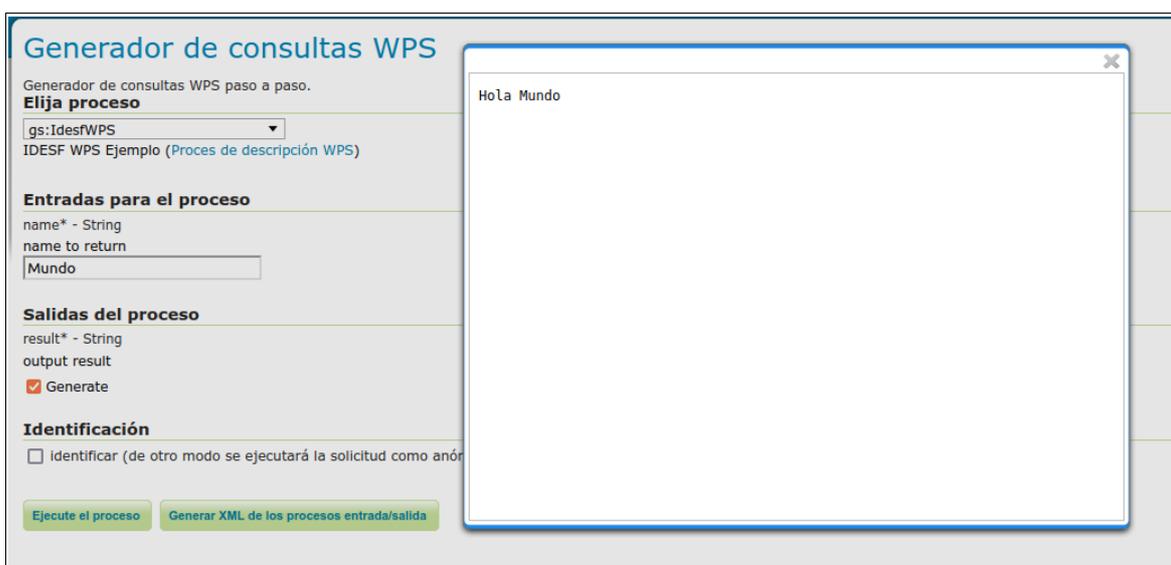


Figura 31: *Generador de consultas WPS – Proceso gs:IdesfWPS – Resultado.*

6. DISCUSIÓN, CONCLUSIONES Y TRABAJOS FUTUROS

6.1 Discusión

Para el desarrollo de este trabajo, se contemplaron las herramientas informáticas utilizadas por la IDESF para la implementación del prototipo WPS. El propósito del prototipo está basado en un requerimiento que se resolvió con QGIS. Esta solicitud disparó el interrogante si el mismo se podría resolver a través de un proceso WPS y de esta manera facilitar al usuario la obtención de los resultados.

GeoServer es el servidor utilizado para brindar los servicios OGC (WMS, WFS y WPS) necesarios para la realización de este trabajo. El servicio WPS de GeoServer, permite peticiones HTTP GET y POST, como así también modo de ejecución sincrónica como asincrónica (recomendado).

Al generar el XML del prototipo WPS, con la ayuda del *Generador de consultas WPS* de GeoServer se detectó que la generación de subprocesos no funcionaba. Se investigó al respecto y se encontró como respuesta que desde algunas versiones anteriores ya presentaba ese problema. Por lo cual, el encadenamiento de procesos se generó por partes hasta lograr el XML completo.

Se construyó un archivo HTML trabajando con las librerías de OpenLayers 6 como cliente ligero WPS consumiendo los servicios de GeoServer. Se investigó en la web acerca del desarrollo de estos códigos y se hicieron pruebas obteniendo resultados satisfactorios. Se logró un visualizador a través del cual se testearon los resultados de solicitudes HTTP POST con XML asincrónicas.

En cambio, al usar QGIS como cliente pesado WPS, a través de su complemento WPS-Client 3.0.0 en la última versión 3.22 LTR, los resultados no fueron satisfactorios. El plugin conectó el servicio WPS exitosamente, pero presentó problemas al momento de generar resultados, excepto para aquellos procesos que devuelven como resultado datos tipo texto o numérico.

En cuanto a la investigación de la factibilidad de creación e incorporación de nuevos procesos WPS a GeoServer, se siguieron procedimientos encontrados en la web³⁹. Implementando de manera satisfactoria un simple ejemplo presentado en la página oficial de GeoServer. Esto podría ser útil, por ejemplo, para generar e incorporar el proceso wps que permita *agregar a una de las capas de entrada el nombre del elemento más cercano de otra capa de entrada y la distancia al mismo*, que como se dijo no está implementado en GeoServer y hubiese sido interesante agregarlo al prototipo wps del presente trabajo. Por exceder el alcance de este trabajo, sólo se documentó la generación de un proceso básico, pero cabe mencionar que se realizaron pruebas con códigos de geoprosesos encontrados en la web, que funcionaron satisfactoriamente.

Al ejecutar el proceso *gs:import* daba errores. Uno de ellos era porque el espacio de trabajo no contaba con los permisos necesarios. Otro de los errores, que se descubrió su motivo a través de prueba-error (el mensaje de error que arrojaba no

39 <http://og.cens.am:8081/opengeo-docs/processing/wpsjava/index.html>
<https://docs.geotools.org/latest/userguide/tutorial/process.html>

era muy preciso), ocurría al no seleccionar ninguna capa raster de entrada. Aunque esta entrada no era necesaria para la ejecución del proceso, al darle un valor arbitrario el proceso se ejecutó sin problemas. Este último error se presentó sólo utilizando el *Generador de consultas WPS* de GeoServer, no así desde OpenLayers.

La construcción de la petición de manera correcta, nos permite resolver análisis espaciales complejos con una sola solicitud. Usar las capas desde el mismo GeoServer sería una ventaja en cuanto a rendimiento. Igualmente, todo esto debería ser verificado en la Infraestructura Tecnológica utilizada por la IDESF, teniendo en cuenta que ejecutar una solicitud POST encadenada de procesos muy compleja puede requerir algo de tiempo antes de devolver el resultado. Asimismo, en este caso se trabajó con el WPS implementado en GeoServer, debido a que la IDESF utiliza este software como Servidor de Mapas, lo que representa una ventaja al estar ya implementado, pero ello no quita que se evalúe otra opción Open Source más adecuada en cuanto a implementación y tiempos de respuestas.

6.2 Conclusiones

El prototipo implementado en este trabajo es una iniciativa en la incorporación del servicio WPS por parte de la IDESF. Importante a la hora de brindar asistencia de procesamiento geoespacial, tanto a usuarios como a productores de IG, en la toma de decisiones en el territorio provincial santafesino.

Mediante la investigación y las pruebas realizadas sobre las herramientas citadas en el presente trabajo se logró la satisfacción de los objetivos planteados. Se implementó el prototipo WPS utilizando los procesos brindados por GeoServer y se probó a través de clientes WPS, con resultados satisfactorios con OpenLayers. Además, se consiguió verificar que es posible generar e incorporar nuevos procesos WPS, muy importante para resolver solicitudes que no pueden llevarse a cabo con los procesos disponibles en GeoServer.

Muchas veces se provocan retrasos en la atención de las solicitudes relacionadas con IG, que por lo general requieren respuesta inmediata. Implementar el servicio WPS facilitaría a los usuarios finales la realización de geoprocesamientos de acuerdo a sus necesidades, sin que sea necesario tener conocimientos de manejo de algún software SIG de escritorio.

La IDESF al incorporar el servicio WPS ofrecería servicios de procesamiento geoespacial de forma fiable, facilitando la generación de IG y servir como un medio para compartir datos espaciales. Cumpliendo así con uno de los objetivos de la IDESF que es brindar servicios de IG, facilitando su acceso y uso por parte de los usuarios.

6.3 Trabajos Futuros

Durante el desarrollo de este trabajo han ido surgiendo varios temas relacionados al mismo, los cuales pueden considerarse como futuras líneas de investigación.

- Incorporar otros procesos WPS brindados por GeoServer, al Geoportal de la IDESF.
- Detectar y generar geoprocesos, para incorporar a GeoServer, necesarios para satisfacer necesidades de los usuarios.
- Analizar la implementación del Servicio WPS desde otros softwares Open Source.
- Investigar el nuevo estándar OGC API – Processes, que viene a reemplazar al WPS de OGC. El mismo define API centradas en recursos que aprovechan las prácticas modernas de desarrollo web. (A la fecha no está incorporado a GeoServer).

En cuanto a otros servicios de geoprocesamiento que se pueden brindar, se puede analizar la posibilidad de: convertir información tabular con datos de las coordenadas en una capa geográfica puntual; hacer recortes de capas a través de zonas de interés; reproyectar capas, facilitar la impresión de mapas como así

también la generación de simples listados (ejemplo, las escuelas de una localidad).

En cuanto a la creación y generación de nuevos geoprocesos WPS que puedan incorporarse al servidor, deberían realizarse de manera tal que puedan ser compartidos con otras IDE. Tal vez a través de IDERA podría concretarse un repositorio de geoprocesos WPS, para evitar la duplicación de esfuerzos y/o potenciar los servicios brindados por las IDE.

7. BIBLIOGRAFÍA

B. Mearns, *Expert GeoServer: Build and secure advanced interfaces and interactive maps*. Packt Publishing Ltd., Birmingham, 2018.

<https://books.google.com.ar/books?id=AAInDwAAQBAJ&lpg=PP1&hl=es&pg=PP1#v=onepage&q&f=false>

D. Oliveros, J. Bosque, *El papel del estándar WPS en la arquitectura orientada a servicios en SIG*, XV Congreso Nacional de Tecnologías de la Información Geográfica, Madrid, AGE-CSIC, 19-21 de Septiembre de 2012. Disponible en <https://1library.co/document/z1d7popv-papel-estandar-wps-arquitectura-orientada-servicios-sig.html>, consultado en Mayo-2022.

GeoServer

<http://geoserver.org/>, consultado en Octubre-2022.

GeoServer 2.21.x Developer Manual, *Implementing a WPS Process*. Disponible en

<https://docs.geoserver.org/2.21.x/en/developer/programming-guide/wps-services/implementing.html>, consultado en Mayo 2022.

GeoSolutions, GeoServer Training, *Web Processing Service and Rendering Transformations*.

Disponible en <https://docs.geoserver.geo-solutions.it/edu/en/wps/index.html>, consultado en Agosto-2022.

Gobierno de la Provincia de Santa Fe, *Infraestructura de Datos Espaciales de la Provincia de Santa Fe (IDESF)*.

<https://www.santafe.gob.ar/idesf>, consultado en Septiembre-2022.

IDECOR, *¿Conoces los GeoPackages? Una evolución interesante para guardar tus archivos geográficos*. Disponible en <https://www.idecor.gob.ar/conoces-los-geopackages-una-evolucion-interesante-para-guardar-tus-archivos-geograficos/>, consultado en Septiembre-2022.

IDEE, *Estándar de Interfaz. Web Processing Service (WPS) Versión 1.0.0*, 2021. Disponible en https://www.idee.es/resources/documentos/RD_wps_v1_0.pdf

IGN, *División Política, Superficie y Población*. Disponible en

<https://www.ign.gob.ar/NuestrasActividades/Geografia/DatosArgentina/DivisionPolitica>, consultado en Septiembre-2022.

INDEC, *Anuario Estadístico de la República Argentina*, 2017. Disponible en

https://www.indec.gob.ar/ftp/cuadros/publicaciones/anuario_estadistico_2017.pdf

INDEC, *Composición y distribución*. Disponible en

<https://www.indec.gob.ar/ftp/cuadros/poblacion/n020210.xls>, consultado en Septiembre-2022.

Infraestructura de Datos Espaciales de la República Argentina (IDERA).

<https://www.idera.gob.ar/>, consultado en Septiembre-2022.

M. A. Bernabé-Poveda, C. M. López-Vázquez, *Fundamentos de las Infraestructuras de Datos Espaciales (IDE)*. UPM-Press, Madrid, 2012.

M. Iniesto, A. Núñez, *Introducción a las Infraestructuras de Datos Espaciales*. 2014.
<https://www.santafe.gob.ar/idesf/geoportal/recursos/documentos/libros/LibroIntroduccionalasIDE2015.pdf>

M. J. Iniesto Alba, A. Núñez Andrés, *Infraestructuras de Datos Espaciales*. O. A. Centro Nacional de Información Geográfica, Madrid, 2021.
<http://www.ign.es/web/ign/portal/libros-digitales/infraestructuras-datos-espaciales>

O. A. López Hernández, *Desarrollo de Aplicación y Servicio Web Geoespacial para la difusión de información relativa al COVID-19 en Andalucía*. Tesis de Máster, Universidad de Jaén, 2021.
<https://hdl.handle.net/10953.1/15191>

OGC, e-Learning 2.0.0 documentation, *WPS – Introduction*. Disponible en <http://opengeospatial.github.io/e-learning/wps/text/basic-main.html>, consultado en Mayo-2022.

Open Geospatial Consortium (OGC). *Implementations by Specification*. Disponible en <https://www.ogc.org/resource/products/byspec>, consultado en Febrero-2022.

Open Geospatial Consortium (OGC). *Web Processing Service*. Disponible en <https://www.ogc.org/standards/wps>, consultado en Julio-2022.

P. J. Zader, *Implementación de geoprocesos como servicios web*. Tesis de Magíster, Universidad Nacional de Córdoba, 2016.
<https://www.famaf.unc.edu.ar/documents/1351/43-Zader.pdf>

PROJ4JS
<http://proj4js.org/>, consultado en Septiembre 2022.

QGIS, *Un Sistema de Información Geográfica libre y de Código Abierto*. Disponible en <https://www.qgis.org/>, consultado en Agosto-2022.

Wikipedia, *Anexo: Departamentos de la provincia de Santa Fe*. Disponible en https://es.wikipedia.org/wiki/Anexo:Departamentos_de_la_provincia_de_Santa_Fe, consultado en Septiembre-2022.

Wikipedia, *GeoPackage*. Disponible en <https://en.wikipedia.org/wiki/GeoPackage>, consultado en Septiembre-2022.

Wikipedia, *Maven*. Disponible en <https://es.wikipedia.org/wiki/Maven>, consultado en Septiembre-2022.

Wikipedia, *PostGIS*. Disponible en <https://es.wikipedia.org/wiki/PostGIS>, consultado en Mayo-2022.

ANEXO 1. Requerimiento resuelto con QGIS

Requerimiento: obtener las escuelas del área de influencia de 1000m de los Barrios Populares para asegurar que todas tengan acceso a WiFi.

Capas necesarias:

- Escuelas estatales de la provincia de Santa Fe.
- Barrios populares de las ciudades de Santa Fe y Rosario.

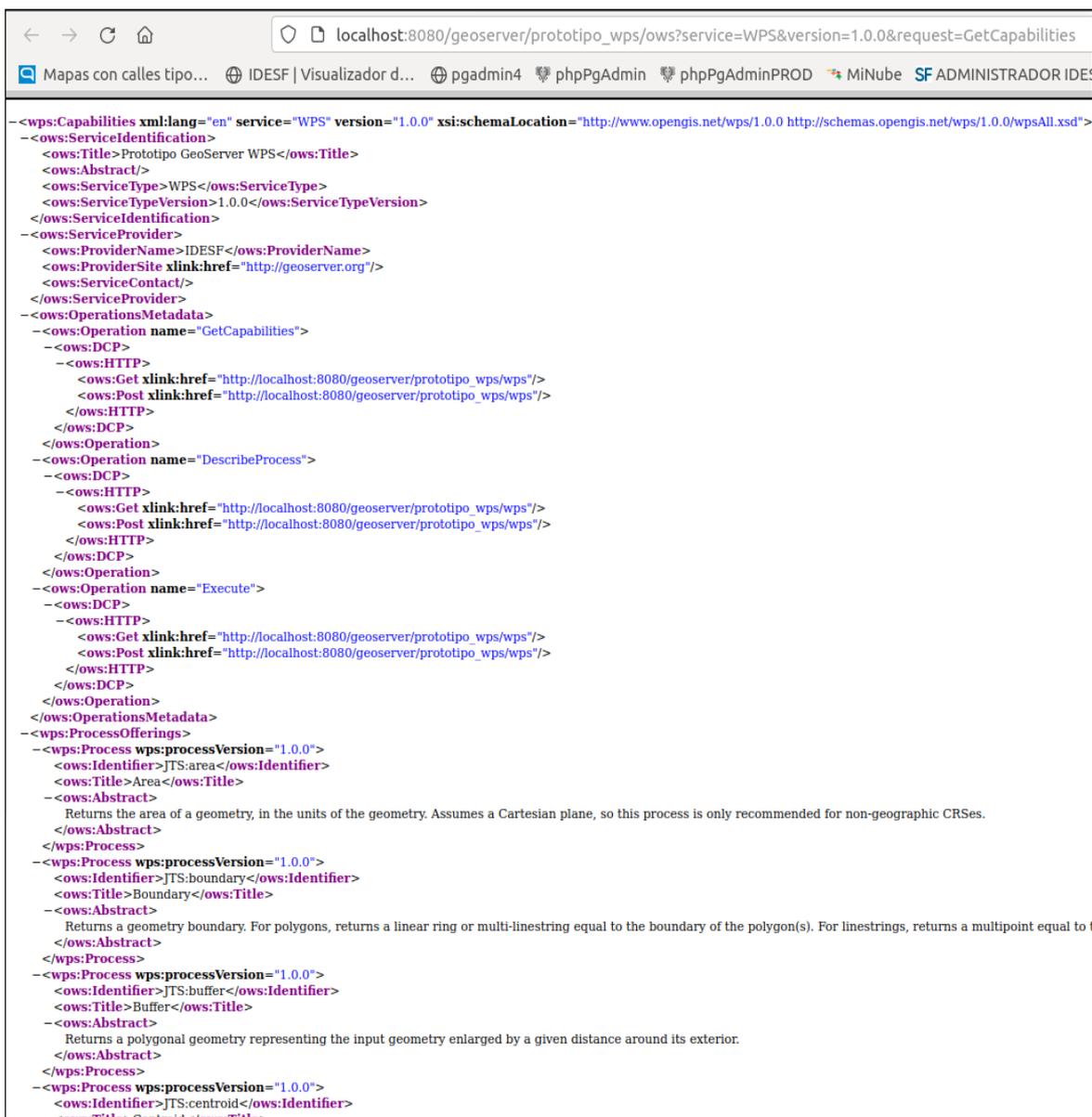
Pasos realizados para la resolución del requerimiento utilizando QGIS:

- 1- Una vez descargadas y preparadas, se añadieron a QGIS las capas necesarias.
- 2- Utilizando la herramienta de geoprocso “*Buffer*” (*Vectorial/Herramientas de geoprocso/Buffer*), tomando como entrada la capa de barrios populares, se generó la capa de las áreas de influencia de 1000m de los barrios.
- 3- A través de la herramienta “*Seleccionar por localización*” (*Herramientas de Investigación/Seleccionar por localización*), utilizando la capa de escuelas y la obtenida en el paso anterior, se seleccionaron las escuelas estatales que caen dentro de las áreas de influencia de los barrios populares.
- 4- Se exportaron las escuelas seleccionadas en el paso anterior (*Exportar/Guardar objetos seleccionados como...*) obteniendo la capa resultado.

Por último, utilizando el plugin *NNJoin* previamente instalado (*Vectorial/NNJoin*) se logra enriquecer la capa resultado agregando a la misma el nombre del barrio más cercano para cada escuela y la distancia al mismo, utilizando como entrada la capa resultado del paso 4 y la capa original de barrios populares.

ANEXO 2. WPS – Petición *GetCapabilities*

http://localhost:8080/geoserver/prototipo_wps/ows?service=WPS&version=1.0.0&request=GetCapabilities



```
--<wps:Capabilities xmlns:lang="en" service="WPS" version="1.0.0" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
--<ows:ServiceIdentification>
  <ows:Title>Prototipo GeoServer WPS</ows:Title>
  <ows:Abstract/>
  <ows:ServiceType>WPS</ows:ServiceType>
  <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
</ows:ServiceIdentification>
--<ows:ServiceProvider>
  <ows:ProviderName>IDESF</ows:ProviderName>
  <ows:ProviderSite xlink:href="http://geoserver.org"/>
  <ows:ServiceContact/>
</ows:ServiceProvider>
--<ows:OperationsMetadata>
--<ows:Operation name="GetCapabilities">
  --<ows:DCP>
    --<ows:HTTP>
      <ows:Get xlink:href="http://localhost:8080/geoserver/prototipo_wps/wps"/>
      <ows:Post xlink:href="http://localhost:8080/geoserver/prototipo_wps/wps"/>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
--<ows:Operation name="DescribeProcess">
  --<ows:DCP>
    --<ows:HTTP>
      <ows:Get xlink:href="http://localhost:8080/geoserver/prototipo_wps/wps"/>
      <ows:Post xlink:href="http://localhost:8080/geoserver/prototipo_wps/wps"/>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
--<ows:Operation name="Execute">
  --<ows:DCP>
    --<ows:HTTP>
      <ows:Get xlink:href="http://localhost:8080/geoserver/prototipo_wps/wps"/>
      <ows:Post xlink:href="http://localhost:8080/geoserver/prototipo_wps/wps"/>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>
</ows:OperationsMetadata>
--<wps:Process wps:processVersion="1.0.0">
  <ows:Identifier>JTS:area</ows:Identifier>
  <ows:Title>Area</ows:Title>
  --<ows:Abstract>
    Returns the area of a geometry, in the units of the geometry. Assumes a Cartesian plane, so this process is only recommended for non-geographic CRSes.
  </ows:Abstract>
</wps:Process>
--<wps:Process wps:processVersion="1.0.0">
  <ows:Identifier>JTS:boundary</ows:Identifier>
  <ows:Title>Boundary</ows:Title>
  --<ows:Abstract>
    Returns a geometry boundary. For polygons, returns a linear ring or multi-linestring equal to the boundary of the polygon(s). For linestrings, returns a multipoint equal to
  </ows:Abstract>
</wps:Process>
--<wps:Process wps:processVersion="1.0.0">
  <ows:Identifier>JTS:buffer</ows:Identifier>
  <ows:Title>Buffer</ows:Title>
  --<ows:Abstract>
    Returns a polygonal geometry representing the input geometry enlarged by a given distance around its exterior.
  </ows:Abstract>
</wps:Process>
--<wps:Process wps:processVersion="1.0.0">
  <ows:Identifier>JTS:centroid</ows:Identifier>
  <ows:Title>Centroid</ows:Title>
```

ANEXO 3. WPS – Petición DescribeProcess

gs:buffer

http://localhost:8080/geoserver/prototipo_wps/ows?

[service=WPS&version=1.0.0&request=DescribeProcess&identifier=geo:buffer](http://localhost:8080/geoserver/prototipo_wps/ows?service=WPS&version=1.0.0&request=DescribeProcess&identifier=geo:buffer)

```
-<wps:ProcessDescriptions xml:lang="en" service="WPS" version="1.0.0" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  -<ProcessDescription wps:processVersion="1.0.0" statusSupported="true" storeSupported="true">
    <ows:Identifier>geo:buffer</ows:Identifier>
    <ows:Title>Buffer</ows:Title>
    -<ows:Abstract>
      Returns a polygonal geometry representing the input geometry enlarged by a given distance around its exterior.
    </ows:Abstract>
    -<DataInputs>
      -<Input maxOccurs="1" minOccurs="1">
        <ows:Identifier>geom</ows:Identifier>
        <ows:Title>geom</ows:Title>
        <ows:Abstract>Input geometry</ows:Abstract>
        -<ComplexData>
          -<Default>
            -<Format>
              <MimeType>text/xml; subtype=gml/3.1.1</MimeType>
            </Format>
          </Default>
          -<Supported>
            -<Format>
              <MimeType>text/xml; subtype=gml/3.1.1</MimeType>
            </Format>
            -<Format>
              <MimeType>text/xml; subtype=gml/2.1.2</MimeType>
            </Format>
            -<Format>
              <MimeType>application/wkt</MimeType>
            </Format>
            -<Format>
              <MimeType>application/gml-3.1.1</MimeType>
            </Format>
            -<Format>
              <MimeType>application/gml-2.1.2</MimeType>
            </Format>
            -<Format>
              <MimeType>application/json</MimeType>
            </Format>
          </Supported>
        </ComplexData>
      </Input>
      -<Input maxOccurs="1" minOccurs="1">
        <ows:Identifier>distance</ows:Identifier>
        <ows:Title>distance</ows:Title>
        -<ows:Abstract>
          Distance to buffer the input geometry, in the units of the geometry
        </ows:Abstract>
        -<LiteralData>
          <ows:DataType>xs:double</ows:DataType>
          <ows:AnyValue/>
        </LiteralData>
      </Input>
      +<Input maxOccurs="1" minOccurs="0"></Input>
      +<Input maxOccurs="1" minOccurs="0"></Input>
    </DataInputs>
    -<ProcessOutputs>
      -<Output>
        <ows:Identifier>result</ows:Identifier>
        <ows:Title>result</ows:Title>
        -<ComplexOutput>
          -<Default>
            +<Format></Format>
          </Default>
          -<Supported>
            -<Format>
              <MimeType>text/xml; subtype=gml/3.1.1</MimeType>
            </Format>
            -<Format>
              <MimeType>text/xml; subtype=gml/2.1.2</MimeType>
            </Format>
            -<Format>
              <MimeType>application/wkt</MimeType>
            </Format>
            -<Format>
              <MimeType>application/gml-3.1.1</MimeType>
            </Format>
            -<Format>
              <MimeType>application/gml-2.1.2</MimeType>
            </Format>
            -<Format>
              <MimeType>application/json</MimeType>
            </Format>
          </Supported>
        </ComplexOutput>
      </Output>
    </ProcessOutputs>
  </ProcessDescription>
</wps:ProcessDescriptions>
```

gs:BufferFeatureCollection

http://localhost:8080/geoserver/prototipo_wps/ows?service=WPS&version=1.0.0&request=DescribeProcess&identifier=gs:BufferFeatureCollection

```
-<wps:ProcessDescriptions xml:lang="en" service="WPS" version="1.0.0" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
-<ProcessDescription wps:processVersion="1.0.0" statusSupported="true" storeSupported="true">
  <ows:Identifier>gs:BufferFeatureCollection</ows:Identifier>
  <ows:Title>Buffer</ows:Title>
-<ows:Abstract>
  Buffers features by a distance value supplied either as a parameter or by a feature attribute. Calculates buffers based on Cartesian distances.
</ows:Abstract>
-<DataInputs>
  -<Input maxOccurs="1" minOccurs="1">
    <ows:Identifier>features</ows:Identifier>
    <ows:Title>features</ows:Title>
    <ows:Abstract>Input feature collection</ows:Abstract>
  -<ComplexData>
    +<Default></Default>
  -<Supported>
    -<Format>
      <MimeType>text/xml; subtype=wfs-collection/1.0</MimeType>
    </Format>
    -<Format>
      <MimeType>text/xml; subtype=wfs-collection/1.1</MimeType>
    </Format>
    -<Format>
      <MimeType>application/wfs-collection-1.0</MimeType>
    </Format>
    -<Format>
      <MimeType>application/wfs-collection-1.1</MimeType>
    </Format>
    -<Format>
      <MimeType>application/zip</MimeType>
      <Encoding>base64</Encoding>
    </Format>
    -<Format>
      <MimeType>text/csv</MimeType>
    </Format>
    -<Format>
      <MimeType>application/json</MimeType>
    </Format>
    -<Format>
      <MimeType>application/vnd.google-earth.kml+xml</MimeType>
    </Format>
  </Supported>
</ComplexData>
</Input>
  -<Input maxOccurs="1" minOccurs="1">
    <ows:Identifier>distance</ows:Identifier>
    <ows:Title>distance</ows:Title>
    <ows:Abstract>Fixed value to use for the buffer distance</ows:Abstract>
    +<LiteralData></LiteralData>
  </Input>
  +<Input maxOccurs="1" minOccurs="0"></Input>
</DataInputs>
-<ProcessOutputs>
  -<Output>
    <ows:Identifier>result</ows:Identifier>
    <ows:Title>result</ows:Title>
  -<ComplexOutput>
    +<Default></Default>
  -<Supported>
    -<Format>
      <MimeType>text/xml; subtype=wfs-collection/1.0</MimeType>
    </Format>
    -<Format>
      <MimeType>text/xml; subtype=wfs-collection/1.1</MimeType>
    </Format>
    -<Format>
      <MimeType>application/wfs-collection-1.0</MimeType>
    </Format>
    -<Format>
      <MimeType>application/wfs-collection-1.1</MimeType>
    </Format>
    -<Format>
      <MimeType>application/zip</MimeType>
    </Format>
    -<Format>
      <MimeType>text/csv</MimeType>
    </Format>
    -<Format>
      <MimeType>application/json</MimeType>
    </Format>
    -<Format>
      <MimeType>application/vnd.google-earth.kml+xml</MimeType>
    </Format>
  </Supported>
</ComplexOutput>
</Output>
</ProcessOutputs>
</ProcessDescription>
</wps:ProcessDescriptions>
```

gs:InclusionFeatureCollection

http://localhost:8080/geoserver/prototipo_wps/ows?service=WPS&version=1.0.0&request=DescribeProcess&identifier=gs:InclusionFeatureCollection

```
-<wps:ProcessDescriptions xml:lang="en" service="WPS" version="1.0.0" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wps-process-description.xsd">
  -<ProcessDescription wps:processVersion="1.0.0" statusSupported="true" storeSupported="true">
    <ows:Identifier>gs:InclusionFeatureCollection</ows:Identifier>
    <ows:Title>Inclusion of Feature Collections</ows:Title>
    -<ows:Abstract>
      Returns a feature collection consisting of the features from the first collection which are spatially contained in at least one feature of the second collection.
    </ows:Abstract>
    -<DataInputs>
      -<Input maxOccurs="1" minOccurs="1">
        <ows:Identifier>first</ows:Identifier>
        <ows:Title>first</ows:Title>
        <ows:Abstract>First feature collection</ows:Abstract>
        -<ComplexData>
          +<Default></Default>
          +<Supported></Supported>
          -<Format>
            <MimeType>text/xml; subtype=wfs-collection/1.0</MimeType>
          </Format>
          -<Format>
            <MimeType>text/xml; subtype=wfs-collection/1.1</MimeType>
          </Format>
          -<Format>
            <MimeType>application/wfs-collection-1.0</MimeType>
          </Format>
          -<Format>
            <MimeType>application/wfs-collection-1.1</MimeType>
          </Format>
          -<Format>
            <MimeType>application/zip</MimeType>
            <Encoding>base64</Encoding>
          </Format>
          -<Format>
            <MimeType>text/csv</MimeType>
          </Format>
          -<Format>
            <MimeType>application/json</MimeType>
          </Format>
          -<Format>
            <MimeType>application/vnd.google-earth.kml+xml</MimeType>
          </Format>
          </Supported>
        </ComplexData>
      </Input>
      -<Input maxOccurs="1" minOccurs="1">
        <ows:Identifier>second</ows:Identifier>
        <ows:Title>second</ows:Title>
        <ows:Abstract>Second feature collection</ows:Abstract>
        -<ComplexData>
          +<Default></Default>
          +<Supported></Supported>
        </ComplexData>
      </Input>
    </DataInputs>
    -<ProcessOutputs>
      -<Output>
        <ows:Identifier>result</ows:Identifier>
        <ows:Title>result</ows:Title>
        -<ComplexOutput>
          +<Default></Default>
          +<Supported></Supported>
          -<Format>
            <MimeType>text/xml; subtype=wfs-collection/1.0</MimeType>
          </Format>
          -<Format>
            <MimeType>text/xml; subtype=wfs-collection/1.1</MimeType>
          </Format>
          -<Format>
            <MimeType>application/wfs-collection-1.0</MimeType>
          </Format>
          -<Format>
            <MimeType>application/wfs-collection-1.1</MimeType>
          </Format>
          -<Format>
            <MimeType>application/zip</MimeType>
          </Format>
          -<Format>
            <MimeType>text/csv</MimeType>
          </Format>
          -<Format>
            <MimeType>application/json</MimeType>
          </Format>
          -<Format>
            <MimeType>application/vnd.google-earth.kml+xml</MimeType>
          </Format>
          </Supported>
        </ComplexOutput>
      </Output>
    </ProcessOutputs>
  </ProcessDescription>
</wps:ProcessDescriptions>
```

gs:CollectGeometries

http://localhost:8080/geoserver/prototipo_wps/ows?service=WPS&version=1.0.0&request=DescribeProcess&identifier=gs:CollectGeometries

```
-<wps:ProcessDescriptions xml:lang="en" service="WPS" version="1.0.0" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  -<ProcessDescription wps:processVersion="1.0.0" statusSupported="true" storeSupported="true">
    <ows:Identifier>gs:CollectGeometries</ows:Identifier>
    <ows:Title>Collect Geometries</ows:Title>
    -<ows:Abstract>
      Collects the default geometries of the input features and combines them into a single geometry collection
    </ows:Abstract>
    -<DataInputs>
      -<Input maxOccurs="1" minOccurs="1">
        <ows:Identifier>features</ows:Identifier>
        <ows:Title>features</ows:Title>
        <ows:Abstract>Input feature collection</ows:Abstract>
        -<ComplexData>
          -<Default>
            -<Format>
              <MimeType>text/xml; subtype=wfs-collection/1.0</MimeType>
            </Format>
          </Default>
          -<Supported>
            -<Format>
              <MimeType>text/xml; subtype=wfs-collection/1.0</MimeType>
            </Format>
            -<Format>
              <MimeType>text/xml; subtype=wfs-collection/1.1</MimeType>
            </Format>
            -<Format>
              <MimeType>application/wfs-collection-1.0</MimeType>
            </Format>
            -<Format>
              <MimeType>application/wfs-collection-1.1</MimeType>
            </Format>
            -<Format>
              <MimeType>application/zip</MimeType>
              <Encoding>base64</Encoding>
            </Format>
            -<Format>
              <MimeType>application/json</MimeType>
            </Format>
            -<Format>
              <MimeType>application/vnd.google-earth.kml+xml</MimeType>
            </Format>
          </Supported>
        </ComplexData>
      </Input>
    </DataInputs>
    -<ProcessOutputs>
      -<Output>
        <ows:Identifier>result</ows:Identifier>
        <ows:Title>result</ows:Title>
        -<ComplexOutput>
          -<Default>
            -<Format>
              <MimeType>text/xml; subtype=gml/3.1.1</MimeType>
            </Format>
          </Default>
          -<Supported>
            -<Format>
              <MimeType>text/xml; subtype=gml/3.1.1</MimeType>
            </Format>
            -<Format>
              <MimeType>text/xml; subtype=gml/2.1.2</MimeType>
            </Format>
            -<Format>
              <MimeType>application/wkt</MimeType>
            </Format>
            -<Format>
              <MimeType>application/gml-3.1.1</MimeType>
            </Format>
            -<Format>
              <MimeType>application/gml-2.1.2</MimeType>
            </Format>
            -<Format>
              <MimeType>application/json</MimeType>
            </Format>
          </Supported>
        </ComplexOutput>
      </Output>
    </ProcessOutputs>
  </ProcessDescription>
</wps:ProcessDescriptions>
```

gs:Import

http://localhost:8080/geoserver/prototipo_wps/ows?service=WPS&version=1.0.0&request=DescribeProcess&identifier=gs:Import

```
-<wps:ProcessDescriptions xml:lang="en" service="WPS" version="1.0.0" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wps-process.xsd" statusSupported="true" storeSupported="true">
  -<ProcessDescription wps:processVersion="1.0.0" statusSupported="true" storeSupported="true">
    <ows:Identifier>gs:Import</ows:Identifier>
    <ows:Title>Import to Catalog</ows:Title>
    <ows:Abstract>Imports a feature collection into the catalog</ows:Abstract>
    -<DataInputs>
      -<Input maxOccurs="1" minOccurs="0">
        <ows:Identifier>features</ows:Identifier>
        <ows:Title>features</ows:Title>
        <ows:Abstract>Input feature collection</ows:Abstract>
        -<ComplexData>
          +<Default></Default>
          -<Supported>
            -<Format>
              <MimeType>text/xml; subtype=wfs-collection/1.0</MimeType>
            </Format>
            -<Format>
              <MimeType>text/xml; subtype=wfs-collection/1.1</MimeType>
            </Format>
            -<Format>
              <MimeType>application/wfs-collection-1.0</MimeType>
            </Format>
            -<Format>
              <MimeType>application/wfs-collection-1.1</MimeType>
            </Format>
            -<Format>
              <MimeType>application/zip</MimeType>
              <Encoding>base64</Encoding>
            </Format>
            -<Format>
              <MimeType>text/csv</MimeType>
            </Format>
            -<Format>
              <MimeType>application/json</MimeType>
            </Format>
            -<Format>
              <MimeType>application/vnd.google-earth.kml+xml</MimeType>
            </Format>
          </Supported>
        </ComplexData>
      </Input>
      +<Input maxOccurs="1" minOccurs="0"></Input>
      -<Input maxOccurs="1" minOccurs="0">
        <ows:Identifier>workspace</ows:Identifier>
        <ows:Title>workspace</ows:Title>
        <ows:Abstract>Target workspace (default is the system default)</ows:Abstract>
        +<LiteralData></LiteralData>
      </Input>
      -<Input maxOccurs="1" minOccurs="0">
        <ows:Identifier>store</ows:Identifier>
        <ows:Title>store</ows:Title>
        <ows:Abstract>Target store (default is the workspace default)</ows:Abstract>
        +<LiteralData></LiteralData>
      </Input>
      -<Input maxOccurs="1" minOccurs="0">
        <ows:Identifier>name</ows:Identifier>
        <ows:Title>name</ows:Title>
        -<ows:Abstract>
          Name of the new featuretype/coverage (default is the name of the features in the collection)
        </ows:Abstract>
        +<LiteralData></LiteralData>
      </Input>
      -<Input maxOccurs="1" minOccurs="0">
        <ows:Identifier>srs</ows:Identifier>
        <ows:Title>srs</ows:Title>
        -<ows:Abstract>
          Target coordinate reference system (default is based on source when possible)
        </ows:Abstract>
        +<LiteralData></LiteralData>
      </Input>
      -<Input maxOccurs="1" minOccurs="0">
        <ows:Identifier>srsHandling</ows:Identifier>
        <ows:Title>srsHandling</ows:Title>
        -<ows:Abstract>
          Desired SRS handling (default is FORCE_DECLARED, others are REPROJECT_TO_DECLARED or NONE)
        </ows:Abstract>
        +<LiteralData></LiteralData>
      </Input>
      -<Input maxOccurs="1" minOccurs="0">
        <ows:Identifier>styleName</ows:Identifier>
        <ows:Title>styleName</ows:Title>
        -<ows:Abstract>
          Name of the style to be associated with the layer (default is a standard geometry-specific style)
        </ows:Abstract>
        +<LiteralData></LiteralData>
      </Input>
    </DataInputs>
    -<ProcessOutputs>
      -<Output>
        <ows:Identifier>layerName</ows:Identifier>
        <ows:Title>layerName</ows:Title>
        <LiteralOutput/>
      </Output>
    </ProcessOutputs>
  </ProcessDescription>
</wps:ProcessDescriptions>
```

ANEXO 4. XML del encadenamiento de procesos del prototipo WPS

```
<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0" service="WPS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opengis.net/wps/1.0.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:wcs="http://www.opengis.net/wcs/1.1.1"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
  <ows:Identifier>gs:Import</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>features</ows:Identifier>
      <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wps" method="POST">
        <wps:Body>
          <wps:Execute version="1.0.0" service="WPS">
            <ows:Identifier>gs:InclusionFeatureCollection</ows:Identifier>
            <wps:DataInputs>
              <wps:Input>
                <ows:Identifier>first</ows:Identifier>
                <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
                  <wps:Body>
                    <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:prototipo_wps="http://localhost:8080/geoserver/prototipo_wps">
                      <wfs:Query srsName="EPSG:5347" typeName="prototipo_wps:escuelas_estatal"/>
                    </wfs:GetFeature>
                  </wps:Body>
                </wps:Reference>
              </wps:Input>
              <wps:Input>
                <ows:Identifier>second</ows:Identifier>
                <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wps" method="POST">
                  <wps:Body>
                    <wps:Execute version="1.0.0" service="WPS">
                      <ows:Identifier>gs:BufferFeatureCollection</ows:Identifier>
                      <wps:DataInputs>
                        <wps:Input>
                          <ows:Identifier>features</ows:Identifier>
                          <wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wfs" method="POST">
                            <wps:Body>
                              <wfs:GetFeature service="WFS" version="1.0.0" outputFormat="GML2"
xmlns:prototipo_wps="http://localhost:8080/geoserver/prototipo_wps">
                                <wfs:Query srsName="EPSG:5347" typeName="prototipo_wps:barriospopulares_stafeyrosario"/>
                              </wfs:GetFeature>
                            </wps:Body>
                          </wps:Reference>
                        </wps:Input>
                        <wps:Input>
                          <ows:Identifier>distance</ows:Identifier>
                          <wps>Data>
                            <wps:LiteralData>1000</wps:LiteralData>
                          </wps>Data>
                        </wps:Input>
                      </wps:DataInputs>
                      <wps:ResponseForm>
                        <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
                          <ows:Identifier>result</ows:Identifier>
                        </wps:RawDataOutput>
                      </wps:ResponseForm>
                    </wps:Execute>
                  </wps:Body>
                </wps:Reference>
              </wps:Input>
            </wps:DataInputs>
            <wps:ResponseForm>
              <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
                <ows:Identifier>result</ows:Identifier>
              </wps:RawDataOutput>
            </wps:ResponseForm>
          </wps:Execute>
        </wps:Body>
      </wps:Reference>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:RawDataOutput mimeType="text/xml; subtype=wfs-collection/1.0">
      <ows:Identifier>result</ows:Identifier>
    </wps:RawDataOutput>
  </wps:ResponseForm>
</wps:Execute>
```

```

</wps:ResponseForm>
</wps:Execute>
</wps:Body>
</wps:Reference>
</wps:Input>
<wps:Input>
  <ows:Identifier>workspace</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>prototipo_wps</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>store</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>sfe_mas_conectada</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>name</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>escuelas_ai_resultado</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>srs</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>EPSG:5347</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>srsHandling</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>FORCE_DECLARED</wps:LiteralData>
  </wps:Data>
</wps:Input>
<wps:Input>
  <ows:Identifier>styleName</ows:Identifier>
  <wps:Data>
    <wps:LiteralData>escuelas_resultado</wps:LiteralData>
  </wps:Data>
</wps:Input>
</wps>DataInputs>
<wps:ResponseForm>
  <wps:RawDataOutput>
    <ows:Identifier>layerName</ows:Identifier>
  </wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>

```

ANEXO 5. Variable de JavaScript donde se guarda el XML del prototipo WPS

```
postData = `<?xml version="1.0" encoding="UTF-8"?><wps:Execute version="1.0.0"
service="WPS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.opengis.net/wps/1.0.0" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:gml="http://www.opengis.net/gml" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:wcs="http://www.opengis.net/wcs/1.1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsAll.xsd">
<ows:Identifer>gs:Import</ows:Identifer><wps:DataInputs><wps:Input><ows:Identifer>features</
ows:Identifer><wps:Reference mimeType="text/xml" xlink:href="http://geoserver/wps"
method="POST"><wps:Body><wps:Execute version="1.0.0"
service="WPS"><ows:Identifer>gs:InclusionFeatureCollection</ows:Identifer><wps:DataInputs><
wps:Input><ows:Identifer>first</ows:Identifer><wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wfs" method="POST"><wps:Body><wfs:GetFeature service="WFS"
version="1.0.0" outputFormat="GML2"
xmlns:prototipo_wps="http://localhost:8080/geoserver/prototipo_wps"><wfs:Query
srsName="EPSG:5347"
typeName="prototipo_wps:escuelas_estatal"/></wfs:GetFeature></wps:Body></wps:Reference><
/wps:Input><wps:Input><ows:Identifer>second</ows:Identifer><wps:Reference
mimeType="text/xml" xlink:href="http://geoserver/wps"
method="POST"><wps:Body><wps:Execute version="1.0.0"
service="WPS"><ows:Identifer>gs:BufferFeatureCollection</ows:Identifer><wps:DataInputs><wps:
Input><ows:Identifer>features</ows:Identifer><wps:Reference mimeType="text/xml"
xlink:href="http://geoserver/wfs" method="POST"><wps:Body><wfs:GetFeature service="WFS"
version="1.0.0" outputFormat="GML2"
xmlns:prototipo_wps="http://localhost:8080/geoserver/prototipo_wps"><wfs:Query
srsName="EPSG:5347"
typeName="prototipo_wps:barriospopulares_stafeyrosario"/></wfs:GetFeature></wps:Body></wps:
Reference></wps:Input><wps:Input><ows:Identifer>distance</ows:Identifer><wps:Data><wps:Li
teralData>1000</wps:LiteralData></wps:Data></wps:Input></wps:DataInputs><wps:ResponseFor
m><wps:RawDataOutput mimeType="text/xml; subtype=wfs-
collection/1.0"><ows:Identifer>result</ows:Identifer></wps:RawDataOutput></wps:ResponseFor
m></wps:Execute></wps:Body></wps:Reference></wps:Input></wps:DataInputs><wps:ResponseFor
m><wps:RawDataOutput mimeType="text/xml; subtype=wfs-
collection/1.0"><ows:Identifer>result</ows:Identifer></wps:RawDataOutput></wps:ResponseFor
m></wps:Execute></wps:Body></wps:Reference></wps:Input><wps:Input><ows:Identifer>works
pace</ows:Identifer><wps:Data><wps:LiteralData>prototipo_wps</wps:LiteralData></wps:Data>
</wps:Input><wps:Input><ows:Identifer>store</ows:Identifer><wps:Data><wps:LiteralData>sfe_
mas_conectada</wps:LiteralData></wps:Data></wps:Input><wps:Input><ows:Identifer>name</o
ws:Identifer><wps:Data><wps:LiteralData>escuelas_ai_resultado</wps:LiteralData></wps:Data>
</wps:Input><wps:Input><ows:Identifer>srs</ows:Identifer><wps:Data><wps:LiteralData>EPSG:
5347</wps:LiteralData></wps:Data></wps:Input><wps:Input><ows:Identifer>srsHandling</ows:Id
entifer><wps:Data><wps:LiteralData>FORCE_DECLARED</wps:LiteralData></wps:Data></wps:Id
input><wps:Input><ows:Identifer>styleName</ows:Identifer><wps:Data><wps:LiteralData>escuel
as_resultado</wps:LiteralData></wps:Data></wps:Input></wps:DataInputs><wps:ResponseForm
><wps:RawDataOutput><ows:Identifer>layerName</ows:Identifer></wps:RawDataOutput></wps:
ResponseForm></wps:Execute`;
```

ANEXO 6. Archivo pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.ejemplo.wps</groupId>
  <artifactId>wps_demo_idesf</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>wps_demo_idesf</name>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <geoserver.version>2.21.0</geoserver.version>
    <geotools.version>27.0</geotools.version>
  </properties>
  <repositories>
    <repository>
      <id>boundless</id>
      <name>Boundless Maven Repository</name>
      <url>https://mvnrepository.com/repos/boundless</url>
    </repository>
    <repository>
      <id>osgeo</id>
      <name>OSGeo Release Repository</name>
      <url>https://repo.osgeo.org/repository/release/</url>
    </repository>
  </repositories>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.geotools</groupId>
      <artifactId>gt-process</artifactId>
      <version>${geotools.version}</version>
    </dependency>
    <dependency>
      <groupId>org.geoserver.extension</groupId>
      <artifactId>gs-wps-core</artifactId>
      <version>${geoserver.version}</version>
    </dependency>
    <dependency>
      <groupId>org.geoserver</groupId>
      <artifactId>gs-main</artifactId>
      <version>${geoserver.version}</version>
      <classifier>tests</classifier>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```