



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY



FACULTAD DE
INGENIERÍA

Giso

Plataforma de Geocodificación Asistida y Análisis Geoespacial

Informe de Proyecto de Grado presentado por

Valentina Sediey, Valentín Segura

en cumplimiento parcial de los requerimientos para la graduación de la carrera
de Ingeniería en Computación de Facultad de Ingeniería de la
Universidad de la República

Supervisores

Raquel Sosa
Bruno Rienzi
Laura González

Colaboradora

Janet Cortés

Montevideo, 12 de diciembre de 2024



Giso por Valentina Sediey, Valentín Segura tiene licencia [CC Atribución - No Comercial 4.0](https://creativecommons.org/licenses/by-nc/4.0/).

Resumen

En el contexto actual, las organizaciones enfrentan el desafío de procesar grandes volúmenes de datos empresariales, incluidos aquellos con información georreferenciable, como las direcciones de personas de interés. Sin embargo, procesar estos datos mediante geocodificación masiva y análisis espaciales requiere conocimientos avanzados en *Sistemas de Información Geográfica (GIS)* y programación, lo que limita su aprovechamiento para la toma de decisiones.

El objetivo de este proyecto es proponer una solución e implementar un prototipo que permita a organizaciones, independientemente del sector al que pertenecen, realizar procesos de geocodificación y análisis geoespaciales sin necesidad de contar con conocimientos avanzados en GIS o programación.

En el marco de este proyecto, se realizaron pruebas de concepto de geocodificadores específicos para evaluar su viabilidad de uso. Además, se llevó a cabo un análisis de los visualizadores de datos geográficos disponibles en Uruguay y de las tecnologías relacionadas, complementado con un estudio de los aspectos más relevantes de la información geográfica. Este análisis permitió identificar las mejores prácticas y características de cada tecnología para enriquecer la solución, optimizando las tareas de análisis, mejorando la experiencia del analista y promoviendo su independencia de expertos en GIS para realizar análisis geoespaciales.

La plataforma desarrollada está compuesta por un frontend para administradores, otro para analistas, un backend, y un servidor de servicios espaciales junto con una base de datos geográfica. El frontend de analistas, implementado en *React*, incluye un módulo para la carga y visualización de datos empresariales con direcciones, así como un visualizador de mapas que permite mostrar datos, aplicar filtros, y ejecutar funciones geoespaciales avanzadas, como el cálculo de distancias y la cuantificación de elementos. El backend, desarrollado en *Java*, se encarga de procesar volúmenes de datos y de realizar procesos de geocodificación utilizando servicios externos como *IDeuy* y *Nominatim*. La base de datos geográfica, basada en *PostgreSQL* con la extensión *PostGIS*, almacena tanto datos espaciales como información del negocio. Para mejorar la precisión en la ubicación de direcciones, además de la geocodificación, se incorporó la posibilidad de ubicar puntos manualmente en el mapa.

El prototipo fue aplicado en el dominio de la salud, utilizando datos simulados de emergencias hospitalarias, además de datos reales de accidentes de tránsito en Uruguay, lo que muestra la viabilidad técnica y adecuación funcional

de la solución propuesta.

Palabras clave: Sistema de Información Geográfica, Análisis geoespacial, Datos espaciales, Geocodificación, GeoServer, Leaflet, PostGIS, Java, React.

Índice general

| | |
|--|-----------|
| 1. Introducción | 1 |
| 1.1. Motivación | 1 |
| 1.2. Objetivos | 1 |
| 1.3. Aportes del proyecto | 2 |
| 1.4. Organización del documento | 3 |
| 2. Revisión de antecedentes | 5 |
| 2.1. Datos espaciales | 5 |
| 2.2. Geocodificación | 7 |
| 2.3. Sistemas de Información Geográfica (GIS) | 8 |
| 2.4. Arquitecturas GIS | 9 |
| 2.4.1. Base de datos geográfica | 9 |
| 2.4.2. GeoServer | 9 |
| 2.4.3. Visualizador | 10 |
| 2.5. Web Service Geográficos | 10 |
| 2.6. Computación en la nube | 11 |
| 2.7. Trabajos relacionados | 11 |
| 2.7.1. Herramientas para la visualización de datos espaciales en Uruguay: plataformas actuales | 12 |
| 2.7.2. Herramientas para el análisis geoespacial: revisión de publicaciones científicas | 16 |
| 2.7.3. Análisis de plataformas de geocodificación | 17 |
| 3. Análisis | 19 |
| 3.1. Contexto de trabajo | 19 |
| 3.2. Actores | 20 |
| 3.3. Requerimientos | 20 |
| 3.3.1. Funcionalidades de Analistas | 21 |
| 3.3.2. Funcionalidades de Administradores | 22 |
| 3.3.3. Requerimientos no funcionales | 22 |
| 3.4. Casos de uso críticos | 23 |
| 3.5. Análisis de Geocoders | 24 |
| 3.6. Conclusiones del análisis | 28 |

| | |
|---|-----------|
| 4. Propuesta de solución | 29 |
| 4.1. Diseño | 29 |
| 4.1.1. Modelo conceptual | 29 |
| 4.1.2. Geocodificación de las direcciones | 34 |
| 4.1.3. Formato de los datos importados | 36 |
| 4.2. Arquitectura | 37 |
| 4.2.1. Arquitectura de los frontends | 37 |
| 4.2.2. Arquitectura del backend | 37 |
| 4.2.3. Sistemas e integraciones externas | 38 |
| 4.2.4. Procesamiento con cola de mensajes | 40 |
| 5. Implementación | 43 |
| 5.1. Tecnologías utilizadas | 43 |
| 5.2. Decisiones tecnológicas | 46 |
| 5.2.1. Carga de capas | 46 |
| 5.2.2. Funciones geoespaciales | 47 |
| 5.3. Despliegue en la nube | 47 |
| 5.4. Problemas encontrados y soluciones implementadas | 49 |
| 5.4.1. GeoServer | 49 |
| 5.4.2. Cola de mensajes | 50 |
| 5.5. Pruebas | 50 |
| 5.5.1. Pruebas de performance de carga de datos | 51 |
| 5.5.2. Pruebas de performance de geocodificadores | 53 |
| 6. Caso de estudio | 55 |
| 6.1. Contexto del caso de estudio | 55 |
| 6.2. Datos del negocio | 56 |
| 6.3. Ejecución en la aplicación | 57 |
| 6.3.1. Geocodificación de los puntos | 63 |
| 6.4. Análisis de resultados | 64 |
| 7. Gestión del proyecto | 65 |
| 7.1. Planificación | 65 |
| 7.2. Desviaciones | 66 |
| 7.3. Herramientas | 67 |
| 8. Conclusiones y trabajo futuro | 69 |
| 8.1. Conclusiones | 69 |
| 8.2. Trabajo futuro | 70 |
| Referencias | 73 |

Capítulo 1

Introducción

1.1. Motivación

En la actualidad, las organizaciones manejan grandes volúmenes de datos que son fundamentales para la toma de decisiones y el desarrollo de estrategias. Entre estos datos empresariales, la información georreferenciable, como las direcciones de clientes, proveedores o personas de interés, resulta especialmente valiosa para optimizar la gestión eficiente de recursos.

Los servicios de geocodificación, conocidos como *Geocoders*, asignan coordenadas geográficas a direcciones a través del proceso de geocodificación. Sin embargo, estos servicios suelen operar de forma individual por cada dirección, lo que representa una desventaja cuando se necesita transformar grandes volúmenes de datos. Para gestionar este tipo de tareas masivas, es necesario que un programador desarrolle un proceso automatizado capaz de geocodificar múltiples direcciones de manera eficiente.

Para aprovechar los datos georreferenciados, se llevan a cabo análisis espaciales que implican el manejo de capas geográficas y la ejecución de operaciones geométricas complejas. Estas tareas requieren el uso de sistemas especializados como los GIS, los cuales demandan conocimientos avanzados. Esta necesidad de especialización limita la capacidad de los analistas para realizar estos estudios de manera independiente, sin el apoyo de expertos en la materia.

Con el fin de superar estas barreras, surge la necesidad de diseñar y desarrollar un sistema integral que permita a las organizaciones procesar grandes volúmenes de datos, combinando la geocodificación con análisis geoespaciales avanzados de manera intuitiva y accesible, sin requerir conocimientos especializados en GIS ni programación.

1.2. Objetivos

Como objetivo se plantea diseñar, proponer una solución y desarrollar un prototipo de plataforma que integre datos provenientes de aplicaciones empre-

sariales, geocodifique dicha información para generar datos espaciales y ofrezca servicios de análisis geoespacial.

Para alcanzar este objetivo general, se definen los siguientes objetivos específicos:

- Realizar un estudio inicial sobre geocodificación, las tecnologías disponibles para ello y trabajos relacionados.
- Investigar y analizar las principales técnicas y herramientas de análisis espacial.
- Analizar la problemática e identificar los requerimientos de la plataforma.
- Diseñar la arquitectura de la plataforma, que soporte la geocodificación y el análisis espacial.
- Validar la factibilidad técnica de la plataforma implementando un prototipo.
- Comprobar la adecuación funcional aplicando un caso de estudio concreto.

1.3. Aportes del proyecto

Los principales aportes que se lograron con este proyecto son:

- Análisis de conceptos espaciales, investigación de visualizadores de datos geográficos en Uruguay, y revisión de trabajos relacionados en distintos dominios, con especial enfoque en el ámbito de la salud.
- Pruebas de concepto de uso de *Geocoders*, su funcionamiento y precisión para la integración a la plataforma.
- Definición de requerimientos necesarios para construir una plataforma adaptable a distintos tipos de empresas y dominios.
- Diseño de una arquitectura distribuida, incluyendo una base de datos geográfica y un servidor de mapas para la gestión de los datos espaciales, con autenticación mediante contraseña e integración con autenticación *ID Uruguay*¹, que proporciona un sistema seguro para la gestión de accesos y usuarios dentro de la plataforma. Integración con sistemas externos de geocodificación, con una arquitectura flexible que permite la integración de futuros geocodificadores, sin afectar la estructura existente.
- Implementación de un prototipo web de la plataforma, capaz de cargar datos empresariales, geocodificarlos, visualizarlos en un mapa y realizar análisis espaciales. El prototipo fue desplegado en la plataforma *Elastic*

¹<https://mi.iduruguay.gub.uy/>

*Cloud*² de Antel y cuenta con documentación detallada sobre su despliegue, configuración y uso de la plataforma. Se encuentra disponible en el repositorio de GitLab³.

- Evaluación del prototipo en dos escenarios distintos: uno mediante un caso de uso con datos simulados de pacientes que asisten a una emergencia médica, y otro con datos reales de accidentes de tránsito en Uruguay, obtenidos del *Catálogo de Datos Abiertos*⁴. Estas evaluaciones permitieron verificar la adecuación funcional de la plataforma en contextos aplicados.

Cabe destacar que, como resultado de este proyecto, junto a los tutores se presentó un artículo que fue seleccionado para ser expuesto en la conferencia *IEEE URUCON2024*, que se llevará a cabo en noviembre del presente año (Sosa y cols., 2024).

1.4. Organización del documento

El resto del documento se organiza en los siguientes capítulos:

- Capítulo 2: Revisión de antecedentes. Contiene las investigaciones realizadas sobre los principales conceptos que abarca el proyecto.
- Capítulo 3: Análisis. Presenta el detalle del análisis de requisitos y casos de uso establecidos.
- Capítulo 4: Propuesta de solución. Brinda información sobre el diseño y la arquitectura elegidos para implementar la plataforma.
- Capítulo 5: Implementación. Brinda el detalle de las tecnologías utilizadas, tareas realizadas para el desarrollo y las pruebas a las que fue sometida la plataforma.
- Capítulo 6: Caso de estudio. Especifica el uso de la aplicación en el dominio de salud y los análisis espaciales realizados para dichos datos.
- Capítulo 7: Gestión del proyecto. Describe el transcurso de las etapas del proyecto y las herramientas utilizadas.
- Capítulo 8: Conclusiones y trabajo futuro. Contiene la descripción del resultado obtenido y presenta mejoras que se podrían implementar en un futuro a la plataforma.

Adicionalmente, se entregan junto a este documento en el repositorio de GitLab⁵, el manual de usuario para analistas, el manual de usuario para administradores, el manual de instalación, la documentación detallada de la arquitectura del sistema y la colección de *Postman* de la *Interfaz de Programación de Aplicaciones (API)* del backend.

²https://minubeantel.uy/index.php?NAME_PATH=Elastic_Cloud

³<https://gitlab.fing.edu.uy/groups/open-lins/giso>

⁴https://catalogodatos.gub.uy/dataset/unasev-fallecidos_siniestros_transito

⁵<https://gitlab.fing.edu.uy/groups/open-lins/giso/giso-documentation>

Capítulo 2

Revisión de antecedentes

En este capítulo, se describen los temas relevantes del estudio inicial del dominio, realizado con el fin de alcanzar el objetivo del proyecto. Se analizan estudios previos, definiciones teóricas y enfoques utilizados en proyectos similares.

2.1. Datos espaciales

Los datos espaciales son información sobre la ubicación y las características de fenómenos geográficos. Estos datos se utilizan en una variedad de dominios, como la planificación urbana, la gestión de recursos naturales, la ingeniería y la salud pública, entre otros. Los mismos pueden ser representados por datos vectoriales o raster (Bernabé Poveda y López Vázquez, 2012; Shekhar, Xiong, y Zhou, 2017).

Los datos vectoriales describen entidades atómicas del mundo geográfico, que son representadas por geometrías compuestas por uno o más vértices interconectados.

Para ubicar espacialmente estas entidades se utilizan los ejes X, Y y opcionalmente el Z. Existen tres geometrías que permiten una representación de la realidad espacial (Longley, Maguire, Rhind, Goodchild, y Goodchild, 2010):

- **Punto:** es la entidad geométrica más simple y se representa por un único vértice. Los puntos se utilizan para identificar ubicaciones específicas, como la posición de un edificio o una persona.
- **Línea:** se representa por una secuencia de dos o más vértices conectados, formando una figura lineal que puede ser recta o curva. Las líneas se utilizan para representar objetos como rutas, ríos o fronteras.
- **Polígono:** se forma por una serie de vértices que están conectados para crear un área cerrada. Los polígonos se utilizan para representar áreas definidas como zonas urbanas, departamentos o países.

Para definir las ubicaciones exactas de estas entidades en la superficie terrestre, se utilizan *Sistemas de Referencia de Coordenadas*, que consisten en coordenadas (conjunto de uno o más números) que representan las posiciones en la Tierra. Existen dos tipos de sistemas de referencia de coordenadas, como se muestra en la Figura 2.1 (QGis, 2024):

- **Sistemas de Coordenadas Geográficas:** en este sistema se utilizan las ubicaciones directamente sobre la Tierra, representándolas con valores de latitud, longitud y opcionalmente altitud. El sistema más popularmente utilizado es el *World Geodetic System 1984 (WGS84)*.
- **Sistemas de Coordenadas Proyectadas:** en este sistema, las coordenadas son proyectadas desde la superficie curva de la Tierra a un plano bidimensional. Hay tres familias de proyecciones: cilíndricas, cónicas y planas.

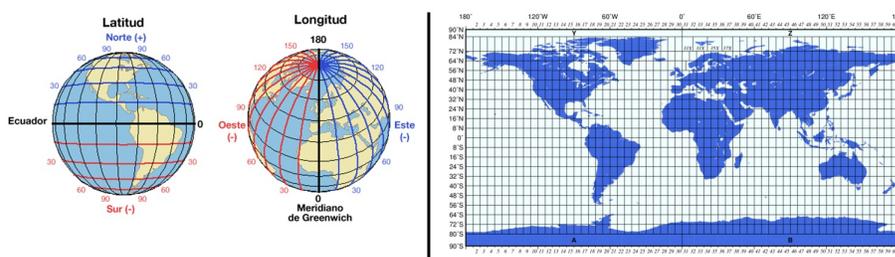


Figura 2.1: Sistema de Coordenadas Geográficas (izq.), Sistemas de Coordenadas Proyectadas (der.) (Chaer, Nadia, 2015).

Existe un identificador único, *Spatial Reference System Identifier (SRID)*, que se utiliza para identificar de manera inequívoca independientemente del sistema de coordenadas utilizado. Estos identificadores se encuentran en registros públicos como el *EPSG Geodetic Parameter Dataset*¹, donde se pueden visualizar las especificaciones y definiciones de los diferentes sistemas de coordenadas para ser utilizados en GIS.

En Uruguay, se emplea el sistema de referencia *EPSG:4326* para coordenadas geográficas y *EPSG:32721* para coordenadas proyectadas. El *EPSG:32721* corresponde al sistema *Universal Transverse Mercator (UTM)* en la zona 21 Sur, que cubre la mayor parte del país. Adicionalmente, el *EPSG:32722* también se utiliza en Uruguay, abarcando una porción del territorio en la zona 22 Sur (ver Figura 2.2).

Otro sistema de referencia relevante es el *EPSG:31981*, que se basa en el datum SIRGAS 2000 y también utiliza el sistema UTM en la zona 21 Sur.

¹<https://epsg.io/>

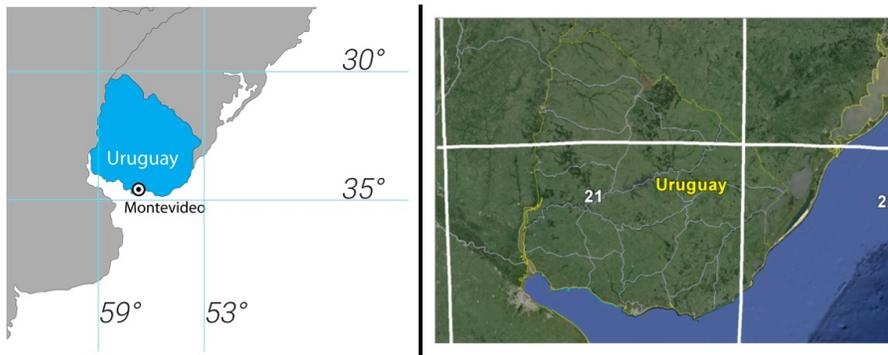


Figura 2.2: Uruguay EPSG:4326 (izq.), Uruguay EPSG:32721 y EPSG:32722 (der.) (Chaer, Nadia, 2015; Instituto Geográfico Militar, 2022).

2.2. Geocodificación

La geocodificación es el proceso por el cual una ubicación que está descrita por una dirección, nombre del lugar o punto de interés, se transforma en un dato vectorial ubicado en la Tierra.

Para llevar a cabo esta transformación, los geocodificadores emplean un proceso que se descompone en varias etapas (Goldberg, Wilson, y Knoblock, 2007):

- **Normalización:** en esta etapa, el texto ingresado se analiza y descompone en unidades más pequeñas llamadas *tokens*. Este análisis incluye la corrección de errores comunes y la eliminación de puntuación. La normalización puede variar en complejidad, desde una simple descomposición hasta métodos probabilísticos avanzados que utilizan técnicas de aprendizaje automático.
- **Estandarización:** este paso consiste en eliminar información superflua y reemplazar abreviaciones comunes. También se pueden aplicar técnicas avanzadas de aprendizaje automático en esta etapa para mejorar la precisión y consistencia de los datos.
- **Búsqueda de coincidencias:** Los componentes estandarizados se comparan con un conjunto de datos de ubicaciones conocidas. Este proceso selecciona el candidato más adecuado y asigna las coordenadas geográficas correspondientes al resultado.
- **Ajustes adicionales:** cuando no se encuentran coincidencias o se identifican múltiples coincidencias, se realizan ajustes adicionales. Estos ajustes pueden incluir el uso de conjuntos de datos complementarios o la simplificación de los datos ingresados para mejorar la precisión de la búsqueda.

En 2012, la *Agencia para el Desarrollo del Gobierno de Gestión Electrónica*

y la Sociedad de la Información y el Conocimiento² (AGESIC) en Uruguay desarrolló el *Modelo de Direcciones Geográficas del Uruguay*. Este documento proporciona una especificación técnica detallada de un modelo de direcciones geográficas aplicable a nivel nacional, definiendo un conjunto mínimo de componentes obligatorios para normalizar direcciones dentro del proceso de geocodificación. Además, define criterios estandarizados para el intercambio y representación de datos, facilitando la interoperabilidad entre organismos. Antes de esta estandarización, cada institución utilizaba su propio sistema, lo que daba lugar a múltiples referencias para una misma dirección según la entidad. El modelo fue actualizado en 2024 por la *Infraestructura de Datos Espaciales de Uruguay*³ (IDEuY).

Los principales componentes definidos por el Modelo de Direcciones incluyen: Departamento, Localidad, Tipo de vialidad (como calle o ruta), y Punto de acceso (número de puerta, kilómetro, manzana/solar, etc.). Estos elementos constituyen los *tokens* esenciales que posibilitan una representación estandarizada y precisa de las direcciones en el país ([Infraestructura de Datos Espaciales, 2024a](#)).

En 2022, Uruguay, basándose en las especificaciones técnicas del Modelo de Direcciones, implementó el *Sistema Único de Direcciones (SuDir)*, una base de datos única y de alta calidad gestionada de manera descentralizada. Este sistema ofrece un formato interoperable y un identificador único a nivel nacional para cada dirección, asegurando su disponibilidad para uso público. La base de datos SuDir y los servicios interinstitucionales que facilitan la gestión de esta información son accesibles a todos los organismos, promoviendo así un modelo de colaboración en la gestión de datos geográficos ([Infraestructura de Datos Espaciales, 2022](#)).

2.3. Sistemas de Información Geográfica (GIS)

Los *Sistemas de Información Geográfica* son herramientas diseñadas para crear, gestionar, analizar, visualizar y manejar datos referenciados mediante coordenadas espaciales o geográficas. Estos sistemas permiten integrar distintos componentes que facilitan la comprensión de patrones, relaciones y contextos geográficos, permitiendo mejorar la administración y toma de decisiones ([ESRI, 2024](#)).

Dentro de las funcionalidades principales que realiza un GIS se encuentran:

- **Gestión de Datos:** permite almacenar datos geográficos e integrarlos con otros datos del dominio, haciéndolos disponibles para su uso por otros subsistemas.
- **Mapeo y visualización:** posibilita la visualización de los datos geográficos dentro de distintos tipos de mapas.

²<https://www.gub.uy/agencia-gobierno-electronico-sociedad-informacion-conocimiento/>

³<https://www.gub.uy/infraestructura-datos-espaciales/>

- **Análisis Geoespacial:** proporciona herramientas para utilizar datos espaciales en la resolución de problemas del mundo real, facilitando la realización de análisis y estadísticas. Este proceso es fundamental en las ciencias sociales y políticas, y tradicionalmente se ha empleado en el estudio de problemas en ecología, la geología y la epidemiología (Dr J de Smith, Prof F Goodchild, Prof A Longley, y Associates, 2021).

2.4. Arquitecturas GIS

Las arquitecturas GIS están diseñadas para manejar grandes volúmenes de datos espaciales y no espaciales. Su estructura varía según los requerimientos específicos de las aplicaciones, pero generalmente incluye elementos como bases de datos geográficas y servidores de mapas.

2.4.1. Base de datos geográfica

Una base de datos es un sistema organizado para almacenar, gestionar y recuperar datos de manera eficiente, permitiendo operaciones como insertar, actualizar, eliminar y consultar.

Las bases de datos geográficas, diseñadas específicamente para manejar datos espaciales, van más allá de las capacidades tradicionales al incorporar funciones geográficas. Estas funciones permiten realizar operaciones y análisis espaciales, como cálculos de longitud (*length*), distancia (*distance*), intersección (*intersection*), contención (*contains*), entre otros.

Existe un estándar definido por el *Open Geospatial Consortium*⁴ (OGC) llamado “*Simple Features*”, que define un modelo de datos y un conjunto de operaciones para manejar geometrías espaciales en bases de datos (Herring, John R., 2011).

2.4.2. GeoServer

*GeoServer*⁵ es un servidor web de código abierto que facilita la publicación de datos espaciales para su uso en aplicaciones web. Se destaca por su capacidad para obtener datos de diversas fuentes y su compatibilidad con estándares abiertos, lo que lo hace altamente interoperable. Entre los estándares del OGC se incluyen *Web Map Service (WMS)*, *Web Coverage Service (WCS)* y *Web Feature Service (WFS)*.

En su funcionamiento, *GeoServer* organiza los datos en capas, cada una de las cuales representa un conjunto de objetos geográficos (*features*). Esta estructura de capas permite una gestión eficiente y flexible de los datos espaciales, facilitando su visualización y análisis.

⁴<https://www.ogc.org/>

⁵<https://geoserver.org/>

Además, *GeoServer* soporta consultas espaciales mediante el uso del lenguaje de consulta *Common Query Language (CQL)*, lo que permite realizar funciones como filtrado y análisis directamente sobre los datos geográficos.

2.4.3. Visualizador

El visualizador o *Map Viewer* es el componente que permite la visualización e interacción con los datos espaciales en formato de mapas dentro de una interfaz de usuario, generalmente en una aplicación cliente. Es el encargado de renderizar la información geoespacial sobre un mapa base.

En la Figura 2.3 se puede observar cómo el visualizador interactúa con la base de datos geográfica y el servidor de mapas (como *GeoServer*) en el contexto de una arquitectura GIS.

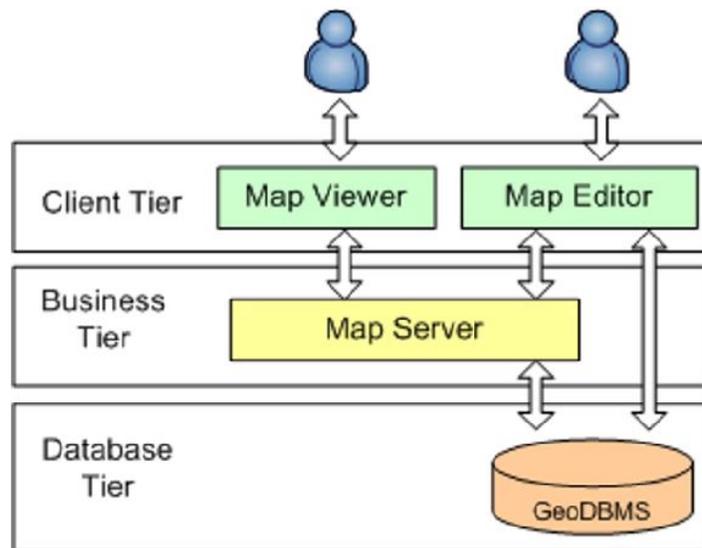


Figura 2.3: Arquitectura GIS (González, Foti, Sosa, y Rienzi, 2010).

2.5. Web Service Geográficos

Un *Web Service* es una interfaz que facilita la comunicación e interoperabilidad entre diferentes aplicaciones o sistemas a través de Internet, utilizando diversos protocolos y arquitecturas. Entre ellos, destacan los servicios web basados en la arquitectura *Representational State Transfer (REST)*, que comúnmente intercambian datos en formato *JavaScript Object Notation (JSON)*.

En el caso de *Web Service Geográfico*, es un tipo de *Web Service* diseñado específicamente para gestionar y proporcionar datos espaciales. Originalmente,

se enfocaba en la generación de mapas dinámicos a partir de datos espaciales almacenados en un servidor.

Dentro de los *Web Services Geográficos*, se encuentra el WFS, un estándar internacional definido por el OGC, que especifica diversas operaciones para interactuar con entidades geográficas a través de una interfaz web (Shekhar y cols., 2017).

Algunas de las operaciones que se destacan son:

- **GetCapabilities:** retorna una descripción del servicio y las colecciones de entidades geográficas.
- **DescribeFeatureType:** retorna las propiedades (atributos) de cada entidad.
- **GetFeature:** retorna una colección específica de entidades espaciales. Los datos devueltos comprenden la geometría y los valores de las propiedades.

2.6. Computación en la nube

En la actualidad, muchos sistemas GIS despliegan todos los componentes de su arquitectura en la nube. La computación en la nube ofrece escalabilidad, flexibilidad, reducción de costos y accesibilidad, facilitando la gestión y análisis de grandes volúmenes de datos espaciales desde cualquier lugar con acceso a Internet.

Entre los diversos servicios en la nube, la *Plataforma como Servicio (PaaS)* se destaca al proporcionar un entorno basado en la nube para que los clientes desplieguen y gestionen sus aplicaciones de software sin preocuparse por la infraestructura subyacente, como sistemas operativos y servidores. Además, ofrece herramientas para controlar las aplicaciones desplegadas y configurar el entorno de alojamiento (Mell y Grance, 2011).

2.7. Trabajos relacionados

Las herramientas GIS son fundamentales en muchos aspectos de la vida moderna. Facilitan la toma de decisiones basadas en datos espaciales precisos, abarcando desde la planificación urbana hasta la gestión de recursos naturales.

En el ámbito de la salud, las herramientas GIS tienen múltiples aplicaciones. Permiten analizar enfermedades para comprender los factores ambientales que influyen en su aparición y estudiar la propagación de epidemias. Actualmente, también se utilizan en la gestión sanitaria, optimizando recursos como la ubicación de salas de emergencia, visualizando centros de población y tiempos de viaje a hospitales. Esto ayuda a definir áreas de cobertura sanitaria y contribuye a la asignación eficiente de recursos (E. Delmelle, Cahill Delmelle, Casas, y Barto, 2011).

En este capítulo, es importante destacar las plataformas GIS disponibles actualmente en Uruguay, así como las investigaciones sobre plataformas similares llevadas a cabo en otras partes del mundo.

2.7.1. Herramientas para la visualización de datos espaciales en Uruguay: plataformas actuales

En Uruguay, existen diversas plataformas web que permiten el acceso a visualizadores de datos espaciales. Cada una de ellas presenta datos específicos proporcionados por la entidad u organización que gestiona la plataforma. Aunque en algunos casos es posible descargar esos datos, no se permite agregar información adicional a los existentes, salvo la opción de añadir puntos de interés o capas espaciales. Además, algunas de estas plataformas incluyen geocodificación, permitiendo ingresar una dirección para ubicarla en el mapa; sin embargo, esta funcionalidad se limita a direcciones individuales y no admite cargas masivas de datos.

La IDEuy, cuyo objetivo es coordinar la producción y facilitar la disponibilidad, el acceso y el uso de productos, servicios e información geográfica del territorio nacional, recopila en su web los visualizadores disponibles en Uruguay ([Registro Nacional de Leyes y Decretos, 2013](#)). Para el estudio realizado, se analizaron algunos de estos visualizadores ([Infraestructura de Datos Espaciales, 2024b](#)).

GeoPortal del Ministerio de Transporte y Obras Públicas

Este visualizador⁶, perteneciente al *Ministerio de Transporte y Obras Públicas (MTOP)*, permite visualizar datos específicos sobre transporte en el mapa, así como acceder a los metadatos de las capas y descargar la información. Los usuarios pueden realizar mediciones espaciales, como distancias y áreas, y cargar capas desde archivos o mediante servicios web geoespaciales (ver Figura 2.4).

La información de cada capa se superpone a varias capas base, incluyendo una específica del MTOP, además de capas de *Google* y *OpenStreetMap*. La información geográfica es proporcionada principalmente por *GeoServer* a través de solicitudes WMS, mientras que algunos elementos se renderizan mediante *MapServer*.

La aplicación utiliza *OpenLayers* para la visualización de mapas interactivos y *GeoMoose*, un framework GIS en *JavaScript*, para la visualización de datos geográficos. *Proj4js* se encarga de la conversión de coordenadas entre distintos sistemas de referencia geoespaciales. Además, se emplean scripts en *PHP* para gestionar ciertas solicitudes y ofrecer información dinámica en el visualizador.

⁶<https://geportal.mtop.gub.uy/visualizador/>

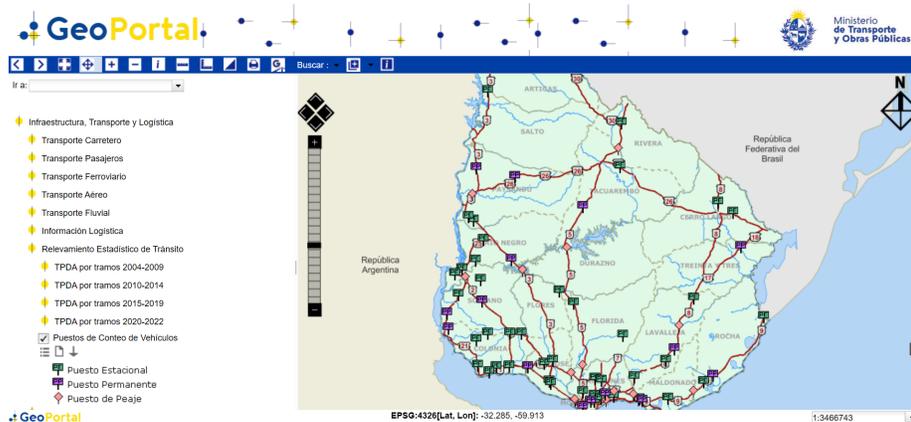


Figura 2.4: GeoPortal, Ministerio de Transporte y Obras Públicas.

Visualizador de mapas del Ministerio de Ambiente de Uruguay

El visualizador⁷ de mapas del *Ministerio de Ambiente de Uruguay* proporciona acceso a información ambiental detallada del país. Además de ofrecer funcionalidades similares a las del GeoPortal del MTOP, permite a los usuarios geolocalizar ubicaciones mediante coordenadas geográficas y localidades, así como transformar coordenadas entre diferentes formatos (ver Figura 2.5).

Las capas base utilizadas en este visualizador son las de *OpenStreetMap* y *Bing Maps*. La información geográfica se suministra a través de *GeoServer*, que realiza las solicitudes de tipo WMS para la visualización de capas de datos ambientales.

Para la visualización de mapas interactivos, el visualizador usa *OpenLayers*, una biblioteca *JavaScript* de código abierto. Además, se emplean scripts *PHP* para gestionar diversas solicitudes del visualizador.

La funcionalidad de geocodificación está basada en *Nominatim*, el servicio de geocodificación de *OpenStreetMap*.

⁷<https://www.ambiente.gub.uy/visualizador/index.php?vis=sig>

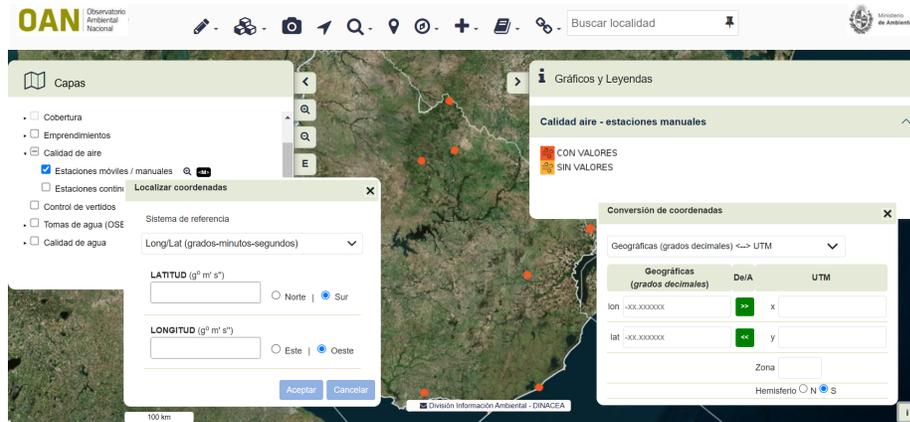


Figura 2.5: Visualizador de Mapas, Ministerio de Ambiente.

MonteviMap de la Intendencia de Montevideo

En esta plataforma⁸, el visualizador permite geocodificar un lugar no solo por sus coordenadas, sino también mediante diferentes valores. Aunque no permite agregar nuevas capas, sí ofrece la opción de añadir puntos de interés al seleccionar una ubicación en el mapa (ver Figura 2.6).

A diferencia de los visualizadores anteriores que utilizan capas base adicionales, este visualizador emplea únicamente una capa base propia.

La tecnología utilizada incluye *PMapper* como interfaz principal, facilitando la interacción cliente-servidor al traducir las solicitudes del cliente al servidor de mapas. En este caso, el servidor de mapas es *MapServer*, complementado con *MapScript* en *PHP* para la manipulación de mapas. Además, la plataforma utiliza *JavaScript* y *Proj4js* para la transformación de coordenadas.

⁸<https://intgis.montevideo.gub.uy/pmapper/map.phtml>

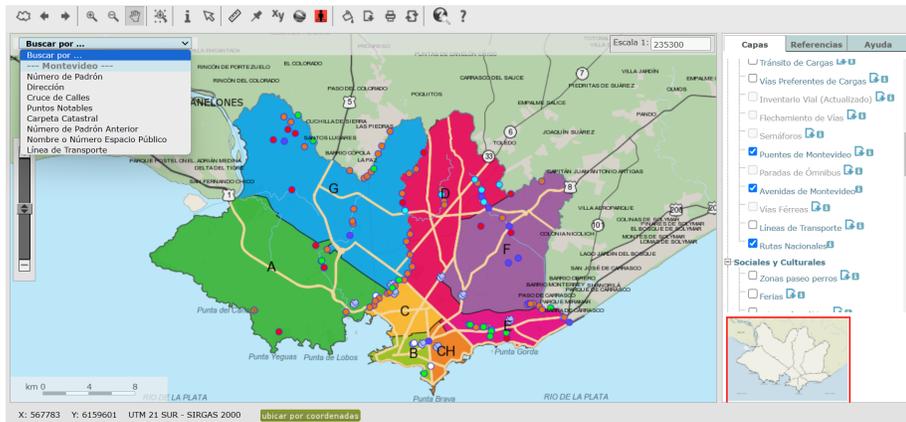


Figura 2.6: MonteviMap, Intendencia de Montevideo.

Visualizador IDEuy

El visualizador⁹ de la IDEuy organiza información específica en varios proyectos temáticos, tales como datos sobre especies exóticas invasoras, el sector vitivinícola y datos del *Instituto Nacional de Estadística* (ver Figura 2.7).

Este sistema comparte varias funcionalidades con otros visualizadores, destacándose por la capacidad de aplicar filtros personalizados a las capas de datos. Es importante señalar que, al aplicar un filtro, el sistema devuelve el conjunto de datos correspondiente, aunque estos datos filtrados no se reflejan visualmente en el mapa.

Las capas base del visualizador incluyen mapas de *OpenStreetMap* y ortoimágenes propias. Estos datos se distribuyen a través de *GeoServer* utilizando *Web Map Tile Service (WMTS)* para las imágenes de mapa, y *WMS* y *WFS* para los datos geospaciales.

El visualizador emplea varias tecnologías como *JavaScript*, *Turf.js*, *Proj4js* y *OpenLayers*. *Turf.js* facilita los análisis geospaciales avanzados, como la manipulación de geometrías y el cálculo de áreas y distancias. La geocodificación se realiza a través del servicio propio de la IDEuy.

⁹<https://visualizador.ide.uy>

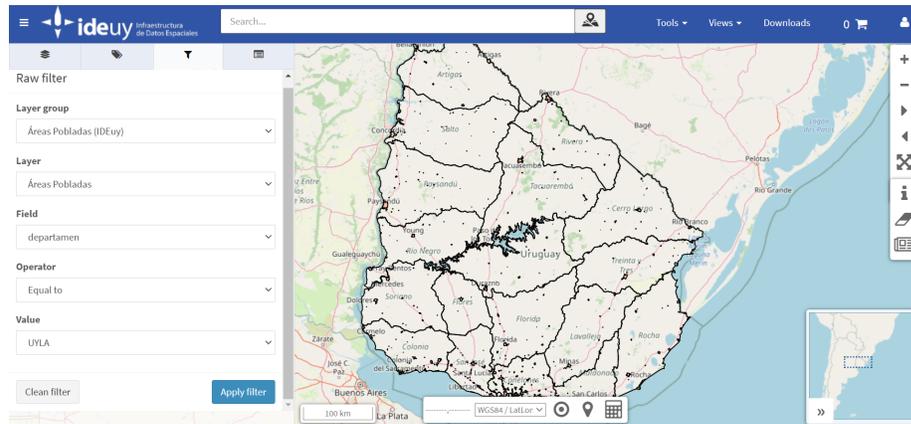


Figura 2.7: Visualizador de la IDEuy.

2.7.2. Herramientas para el análisis geoespacial: revisión de publicaciones científicas

Existen numerosas publicaciones científicas que abordan el análisis geoespacial, centrándose en dominios específicos. Resulta interesante destacar algunos de estos estudios, ya que sus aportes sobre la arquitectura y la gestión de los datos espaciales ofrecen una valiosa perspectiva sobre las prácticas y metodologías empleadas en este campo.

Web-based GIS for spatial pattern detection: application to malaria incidence in Vietnam

En este proyecto, se desarrolló un GIS web interactivo que permite a los profesionales de la salud en Vietnam analizar y visualizar patrones de malaria mediante mapas y estadísticas. El sistema utiliza tecnologías web y GIS para que los usuarios puedan definir áreas de interés, seleccionar niveles de significancia y analizar patrones de enfermedades. Además, facilita la visualización de la densidad de casos a través de un mapa de calor (Bui y Pham, 2016).

A Web-based geospatial toolkit for the monitoring of Dengue Fever

Este proyecto consistió en el desarrollo de una herramienta web geoespacial llamada OnTAPP (Online Toolkit for the Analysis of Point Patterns), diseñada para monitorear brotes de dengue en tiempo y espacio en Colombia. El sistema incluye funcionalidades como la carga de datos, la descarga de resultados del análisis geoespacial, la visualización de patrones de incidencia en un mapa, la generación de mapas de calor para los brotes, entre otras (E. M. Delmelle, Zhu, Tang, y Casas, 2014).

2.7.3. Análisis de plataformas de geocodificación

En el año 2023, se llevó a cabo una investigación sobre diversas plataformas de geocodificación por parte de algunos estudiantes de la materia *Taller de Sistemas de Información Geográfica Empresarial, de la Facultad de Ingeniería de la Udelar*. En su estudio, compararon las características de cada plataforma y realizaron pruebas de geocodificación con direcciones en Uruguay. Esta investigación sirvió como base para el análisis y la selección de las plataformas empleadas en este proyecto. Los detalles completos de la investigación se presentan en la Sección [3.5](#).

Capítulo 3

Análisis

En este capítulo, se presenta un análisis detallado de los requerimientos del proyecto, identificados a través de reuniones con los *stakeholders*. Se describen los actores principales y sus roles, así como los requerimientos funcionales y no funcionales que el sistema debe cumplir. Además, se analizan los casos de uso críticos que la solución debe soportar. Este análisis es esencial para asegurar que todas las necesidades y expectativas sean correctamente atendidas, proporcionando una base precisa para las fases posteriores del proyecto.

3.1. Contexto de trabajo

Como se vio anteriormente, existe la necesidad de que los analistas de datos empresariales procesen grandes volúmenes de información georreferenciable, que se encuentran en formato CSV. En estos archivos, cada fila representa un sujeto de interés (como un cliente, paciente, compra, etc.) vinculado a un evento, mientras que las columnas contienen diversos atributos como nombre, dirección, fecha y otros detalles relevantes para el análisis.

Los analistas requieren una herramienta que les permita cargar estos archivos CSV dentro de espacios de trabajo (*workspaces*), que actúan como proyectos dentro de la aplicación, permitiendo gestionar diferentes conjuntos de datos. Una vez que los datos están cargados en estos espacios de trabajo, las direcciones se geocodifican automáticamente. En los casos en que la geocodificación no sea posible, el sistema ofrece asistencia manual para ubicar los puntos en un mapa.

También es de interés para los analistas poder visualizar todos los puntos en un mapa y cargar distintas capas geográficas, que hoy en día existen como archivos que contienen polígonos o puntos. Estas capas representan elementos como departamentos, barrios, hospitales y centros de distribución. Además, resulta esencial poder cargar puntos de interés, que son coordenadas de ubicaciones relevantes para el dominio en estudio, lo que permitirá llevar a cabo análisis espaciales más completos.

En la Figura 3.1 se muestra un diagrama que representa como se relacionan

los elementos nombrados.

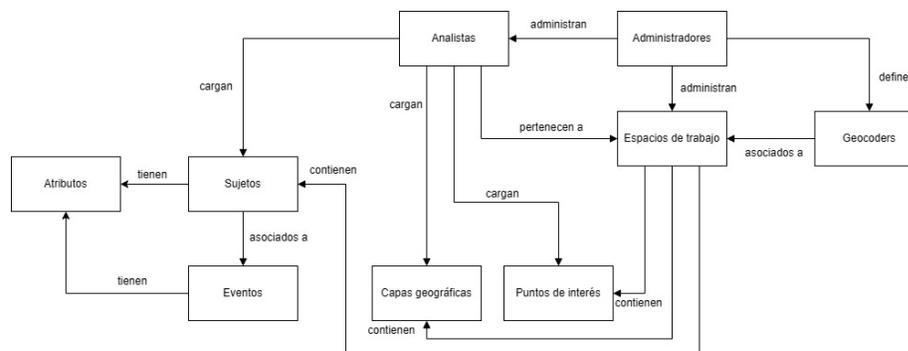


Figura 3.1: Diagrama del contexto de trabajo.

3.2. Actores

Se identificaron dos tipos de actores necesarios para el uso de la aplicación:

- **Analista:** es un usuario que puede tener o no conocimientos en programación o en GIS. Su responsabilidad principal es gestionar la carga de datos que se utilizarán en el sistema. Además, debe poder visualizar estos datos en un mapa, aplicar filtros, y consultar gráficos y estadísticas relacionadas.
- **Administrador:** es el usuario encargado de gestionar la aplicación. Sus responsabilidades incluyen la creación y eliminación de usuarios analistas, la administración de los *workspaces* y la configuración de ajustes globales para estos espacios de trabajo.

3.3. Requerimientos

En esta sección se detallan los requerimientos del sistema, clasificados según las funcionalidades necesarias para cada uno de los actores identificados. Además, se especifican los requerimientos no funcionales que garantizan el rendimiento, la seguridad y la adaptabilidad del sistema. Estos requerimientos aseguran que tanto los analistas como los administradores puedan realizar sus tareas de manera eficaz y eficiente, cumpliendo con las expectativas y necesidades del proyecto.

3.3.1. Funcionalidades de Analistas

1. Autenticación de usuarios:

- El sistema deberá proporcionar un mecanismo para la autenticación de los analistas, sea por usuario y contraseña o por usuario *ID Uruguay*.
- Una vez autenticado, el sistema desplegará los *workspaces* disponibles para el usuario y el mismo seleccionará al que desee ingresar.

2. Gestión de atributos:

- El sistema debe tener la posibilidad de crear, editar y eliminar los atributos asociados a los sujetos y los eventos. Para cada atributo, se deberá poder especificar su nombre, su tipo y su forma de categorizarse.

3. Gestión de puntos de interés:

- El analista debe tener la posibilidad de subir un archivo CSV con el nombre y las coordenadas de los puntos de interés.

4. Gestión de datos:

- El analista deberá tener la capacidad de cargar un archivo CSV que contenga información sobre los sujetos de interés y sus eventos asociados.
- En caso de que no se haga la geocodificación automática, se dará la opción al usuario de geolocalizar manualmente, desplegando un mapa para facilitar la ubicación.
- Posibilidad de filtrar eventos o sujetos de interés en base a si fueron geocodificados automáticamente o no.
- Posibilidad de eliminar sujetos de interés del sistema.

5. Análisis de datos:

- Visualización de puntos que representan sujetos de interés en un mapa.
- Posibilidad de aplicar múltiples filtros combinados utilizando condiciones lógicas predefinidas (como “y” u “o”) basadas en los atributos del sujeto de interés o evento, o mediante funciones geográficas (por ejemplo, filtrando sujetos contenidos en un área específica dentro de una capa de polígonos). El conjunto de puntos resultante debe poder visualizarse en el mapa con la opción de asignarle un color distintivo.

- Capacidad de cargar capas geográficas adicionales, como puntos o polígonos.
- Aplicación de funciones geoespaciales, como por ejemplo visualización de la cantidad de sujetos de interés dentro de un polígono seleccionado.

6. Dashboard de métricas:

- Presentación de gráficos de barra y torta según el atributo seleccionado por el analista.
- Inclusión de métricas numéricas, como promedios o totales de los sujetos de interés o eventos.

3.3.2. Funcionalidades de Administradores

1. Autenticación de usuarios:

- El sistema deberá proporcionar un mecanismo para la autenticación de los administradores con usuario y contraseña.

2. Gestión ABM de usuarios:

- Posibilidad de administrar (alta, baja y modificación) de usuarios y gestionar los roles (lectura, modificación y/o eliminación) de los analistas en el sistema.

3.3.3. Requerimientos no funcionales

■ Seguridad:

- Se deben utilizar servidores nacionales con el fin de promover que los datos se rijan por la normativa nacional.
- Las interacciones deben realizarse utilizando métodos *HTTPS* para preservar la seguridad de los datos.

■ Adaptabilidad:

- El sistema debe ser flexible en la cantidad y tipo de atributos asociados a los sujetos de interés y los eventos, para satisfacer las necesidades específicas de distintos dominios de aplicación (por ejemplo, hospitales, supermercados, etc.).

■ Configurabilidad:

- El sistema debe permitir definir el o los *Geocoders* que se quiere utilizar para cada *workspace*.

- **Implementación:**

- El sistema debe utilizar tecnologías *open source*.
- El sistema debe ser accesible vía web.

- **Interoperabilidad:**

- El sistema debe utilizar estándares del OGC, para consumir datos espaciales de servicios externos.

3.4. Casos de uso críticos

Durante la etapa de obtención de requerimientos, se identificaron varios casos de uso críticos que deben ser soportados para cumplir con las necesidades del proyecto. En la Figura 3.2 se presentan los casos de uso críticos identificados.

La elección de estos casos de uso se fundamenta en su relevancia para el funcionamiento del sistema. Comenzando con la creación de un *workspace*, que es el elemento central donde se cargan y organizan todos los datos; sin un *workspace*, ninguna operación podría llevarse a cabo, ya que todos los datos y atributos dependen de él. El inicio de sesión también es fundamental, ya que garantiza que solo los usuarios autorizados tengan acceso al *workspace*, protegiendo así la integridad de los datos. Por otra parte, la creación de atributos es un paso previo necesario para cargar los datos desde archivos CSV; sin atributos definidos, el sistema no podría interpretar ni procesar adecuadamente la información geográfica y sus características. Luego, la carga de datos implica integrar grandes volúmenes de información y conectarse con sistemas geocodificadores, asegurando que los sujetos sean ubicados geográficamente de manera precisa, lo cual es vital para cualquier análisis posterior. Finalmente, la visualización de datos en el mapa es donde se produce la interacción clave con el sistema, permitiendo a los analistas ver y analizar la información geográficamente. En esta etapa, también se produce la interacción con el servidor de servicios espaciales.

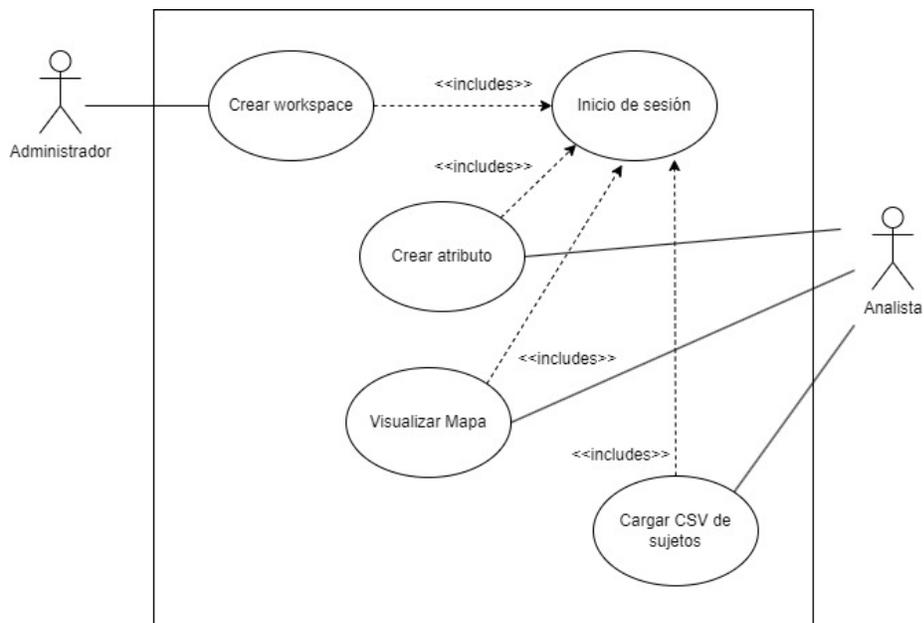


Figura 3.2: Casos de uso críticos.

3.5. Análisis de Geocoders

Uno de los requerimientos de la solución es la capacidad de definir el *Geocoder* a ser utilizado. Este requisito motivó un análisis detallado de diversas plataformas de geocodificación, con el objetivo de identificar aquellas que ofrecieran la mejor combinación de precisión, flexibilidad en la configuración, y compatibilidad con los datos de Uruguay.

Como parte del análisis realizado, se consideraron los resultados de la investigación llevada a cabo por estudiantes de grado¹ en 2023, quienes evaluaron diversos geocodificadores. Esta investigación previa sirvió como base para complementar la evaluación, lo que permitió identificar tres geocodificadores, *IDeuy*, *Nominatim* y *Pelias*, como candidatos para realizar pruebas más específicas.

Las pruebas comparativas se enfocaron en diversas características relacionadas con la infraestructura, configuración y funcionamiento de cada geocodificador. Primero, se verificó la posibilidad de instalación local, evaluando si el software podía ser implementado en servidores propios. Luego, se analizó la facilidad de configuración e instalación, considerando la cantidad de tiempo necesarias para completar el proceso y la disponibilidad de documentación o tutoriales de apoyo. Otro aspecto relevante fue la capacidad de importar bases de datos regionales, donde se estudió la compatibilidad de cada geocodificador con

¹Estudiantes del curso Taller de Sistemas de Información Geográficos Empresariales

bases de datos específicas de Uruguay. Finalmente, se evaluó la precisión del geocodificador, tomando un conjunto de direcciones y comparando las coordenadas obtenidas con su ubicación real.

En la Figura 3.3 se presenta un resumen de los resultados obtenidos en estas pruebas, lo que permitió identificar los geocodificadores más adecuados para las necesidades del proyecto.

| Nombre | Posibilidad de instalación local | Facilidad de configuración e instalación | Capacidad de importar bases de datos regionales | Precisión del geocodificador |
|------------------|----------------------------------|---|---|--|
| IDEuy | No es posible. | No aplica. | Utiliza datos propios de Uruguay. | Alta precisión en direcciones con manzana y solar del interior del país. La mayoría de las direcciones urbanas y rurales son ubicadas correctamente. |
| Nominatim | Es posible. | Requiere entre 1-2 horas y experiencia técnica. Manual para Ubuntu disponible. | Datos regionales de OpenStreetMap. Carga regional en 1 hora. | Alta precisión en zonas urbanas. Problemas con direcciones basadas en kilómetros en rutas: algunas rutas se ubican, pero no en el kilómetro exacto. |
| Pelias | Es posible. | Aproximadamente 2 horas, pero se necesita conocimiento de Docker para la configuración. | Cuenta con la posibilidad de instalar datos de todo el mundo. La carga de los datos regionales demora aproximadamente una hora. | Reconoce ubicaciones a nivel departamental, pero tiene dificultades para ubicar correctamente las direcciones luego de descomponerlas. |

Figura 3.3: Tabla comparativa de geocodificadores analizados.

Para ilustrar cómo se realizó la evaluación de la precisión, se presentan dos ejemplos representativos de las direcciones utilizadas en las pruebas. El primero corresponde a una ubicación en el interior del Uruguay, compuesta por una ruta y un kilómetro específico. El segundo es una dirección urbana en Montevideo, que incluye calle y número de puerta.

En las siguientes figuras, se presenta una comparación visual entre las ubicaciones reales (ver Figuras 3.4 y 3.6) obtenidas mediante *Google Maps*, y las coordenadas generadas por cada geocodificador (ver Figuras 3.5 y 3.7). Se utilizó *Google Maps* debido a la funcionalidad de *Street View*, que permite verificar visualmente el punto exacto en una ruta o calle, asegurando así que las direcciones correspondan a las ubicaciones reales esperadas.

Tanto el servicio de la *IDEuy* como *Nominatim* ofrecen visualizadores de mapa integrados, lo que permite la verificación directa de las coordenadas obtenidas. En el caso de *Pelias*, al no contar con un mapa integrado, las coordenadas devueltas fueron ubicadas manualmente en *Google Maps* para realizar la comparación.

| Dirección | Coordenadas | Punto en el Mapa |
|-------------------------------------|------------------------|---|
| Ruta 5 KM 36500, Canelones, Uruguay | -34.605270, -56.260414 |  |

Figura 3.4: Dirección del interior del Uruguay seleccionada.

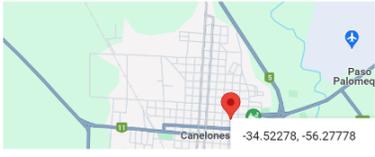
| Geocoder | Resultado | Coordenadas | Punto en el Mapa |
|-----------|------------|----------------------|---|
| IDEuy | Ubicado | -34.59387, -56.2616 |  |
| Nominatim | No ubicado | - | - |
| Pelias | Ubicado | -34.52278, -56.27778 |  |

Figura 3.5: Comparación de geocodificadores para dirección en el interior del país.

Como se puede observar en el ejemplo 3.5, ninguno de los geocodificadores logró ubicar con exactitud la dirección. Sin embargo, el servicio de la *IDEuy* fue el que más se aproximó a la ubicación real.

| Dirección | Coordenadas | Punto en el Mapa |
|--|------------------------|--|
| Julio Herrera y Reissig 738, Montevideo, Montevideo | -34.915110, -56.165686 |  |

Figura 3.6: Dirección de Montevideo seleccionada.

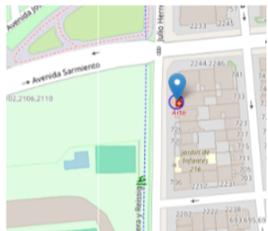
| Geocoder | Resultado | Coordenadas | Punto en el Mapa |
|-----------|-----------|--------------------------|--|
| IDEuy | Ubicado | -34.91531, -56.16564 |  |
| Nominatim | Ubicado | -34.9153057, -56.1656424 |  |
| Pelias | Ubicado | -34.83346, -56.16735 |  |

Figura 3.7: Comparación de geocodificadores para dirección urbana de Montevideo.

Como se observa en el otro ejemplo 3.7, tanto el servicio de la *IDEuy* como *Nominatim* lograron obtener un resultado exacto, mientras que *Pelias* no alcanzó la misma precisión.

Dado el análisis realizado, los resultados obtenidos y el informe utilizado como complemento, se concluyó que los servicios de la *IDEuy* y *Nominatim* son las opciones más adecuadas para este proyecto. Además, *Nominatim* ofrece la ven-

taja de poder ser instalado localmente, lo que permite mitigar riesgos operativos en caso de interrupciones en los servicios públicos de los demás geocodificadores. Esta capacidad de funcionamiento independiente refuerza la elección de estos dos geocodificadores.

3.6. Conclusiones del análisis

Tras la recopilación de requerimientos y el análisis de los visualizadores existentes, se concluyó que ninguna de las plataformas disponibles satisface completamente las necesidades específicas del proyecto. Por ello, se decidió diseñar y desarrollar una nueva plataforma que integre todas las funcionalidades requeridas, garantizando así una solución adaptada a los requisitos definidos.

Capítulo 4

Propuesta de solución

En este capítulo se presenta el diseño arquitectónico de la solución propuesta, detallando las decisiones tomadas durante su elaboración.

4.1. Diseño

En esta sección se detallan las principales decisiones de diseño tomadas para desarrollar la aplicación.

4.1.1. Modelo conceptual

Para cumplir con los requisitos de que la solución sea genérica y agnóstica al dominio, el diseño del modelo conceptual fue una parte crítica del proceso. Se evaluaron dos posibles enfoques.

El primer enfoque permite tener atributos variables, lo que ofrece una gran flexibilidad a la hora de cargar datos y asignar distintas cantidades de atributos a cada sujeto. Esta flexibilidad facilita la personalización de los datos según las necesidades específicas de cada caso. Sin embargo, la principal desventaja de este modelo es que las búsquedas dentro del sistema se vuelven más costosas en términos de rendimiento a medida que se incrementa el volumen de datos.

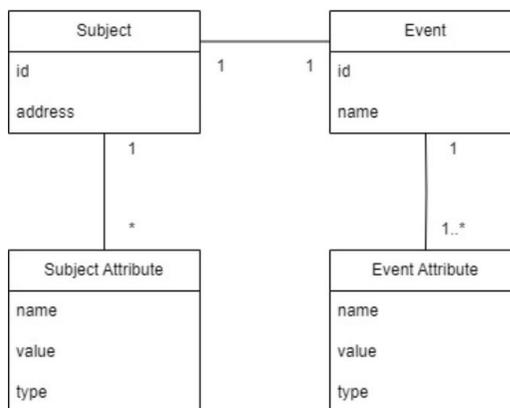


Figura 4.1: Primer enfoque del modelo conceptual.

En la Figura 4.1, el “Subject” representa al sujeto cargado y “Event” al evento asociado. Cada uno cuenta con atributos variables, representados mediante la entidad “Subject Attribute” o “Event Attribute”, donde se almacenan el nombre, el valor y el tipo del atributo. Al cargar un archivo CSV con datos, cada atributo se almacena con el nombre correspondiente al encabezado de la columna, el valor correspondiente y el tipo (por ejemplo: cadena de caracteres, número, etc).

A modo de ejemplo, en la Figura 4.2 se presentan las tablas que ilustran cómo se almacenarían dos sujetos junto con sus respectivos eventos en la base de datos para este enfoque.

Tabla: Subject

| id | address |
|----|-------------------------------|
| 1 | POINT(-34.605270, -56.260414) |
| 2 | POINT(-34.915110, -56.16586) |

Tabla: Event

| id | name | subject_id |
|----|-------------------|------------|
| 1 | Emergencia médica | 1 |
| 2 | Consulta médica | 2 |

Tabla: Subject Attribute

| subject_id | name | value | type |
|------------|-----------|---|---------|
| 1 | Edad | 30 | Integer |
| 1 | Dirección | Ruta 5 KM 36500, Canelones, Uruguay | String |
| 2 | Edad | 57 | Integer |
| 2 | Dirección | Julio Herrera y Reising 738, Montevideo, Montevideo | String |

Tabla: Event Attribute

| event_id | name | value | type |
|----------|---------------|------------|--------|
| 1 | Clasificación | Rojo | String |
| 1 | Fecha | 2024-03-30 | Date |
| 2 | Clasificación | Verde | String |
| 2 | Fecha | 2023-07-15 | Date |

Figura 4.2: Tablas de base de datos para el primer enfoque.

El segundo enfoque, por otro lado, requiere que se definan los atributos principales de un evento y un sujeto antes de la primera carga de datos. En este caso, los atributos son fijos, lo que hace que las búsquedas sean menos costosas en términos de rendimiento y garantiza que la performance del sistema no se degrade significativamente con la adición de nuevos datos. No obstante, agregar nuevos atributos en este modelo es más complejo, ya que implica realizar modificaciones estructurales en la base de datos, como un *ALTER TABLE*, lo que podría bloquear temporalmente las operaciones en esa tabla y afectar la interacción con el sistema. Además, la necesidad de definir y cargar todos los atributos desde el principio puede llevar a que se almacenen datos innecesarios para algunos sujetos, lo que a largo plazo podría afectar negativamente la performance del sistema, especialmente si estos atributos no se utilizan realmente.

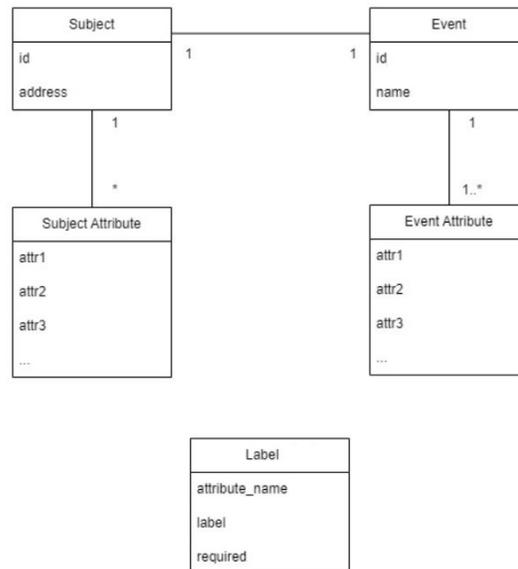


Figura 4.3: Segundo enfoque del modelo conceptual.

En este caso, en la Figura 4.3, cada sujeto y evento cuentan con la lista de los valores de los atributos, representados en las entidades de “Subject Attribute” y “Event Attribute”. Además, se tiene la entidad “Label”, para guardar el nombre del campo indicado por el usuario y si es obligatorio, ya que el identificador del usuario y la dirección son atributos obligatorios para el funcionamiento del sistema. Como ejemplo, en “Subject Attribute” se podría definir *attr1* como el campo “Nombre” de un sujeto. A partir de esto, se creará un registro en “Label” con *attribute_name* como “attr1”, *label* igual a “Nombre” y *required* con el valor que corresponda. A partir de ahí, se cargarán datos en un archivo CSV con la key “Nombre” y se almacenarán en el campo “attr1” de la entidad “Subject

Attribute”.

Nuevamente, a modo de ejemplo, en la Figura 4.4 se presentan las tablas que ilustran cómo se almacenarían dos sujetos junto con sus respectivos eventos en la base de datos para este enfoque.

| id | address |
|----|-------------------------------|
| 1 | POINT(-34.605270, -56.260414) |
| 2 | POINT(-34.915110, -56.16586) |

| id | name | subject_id |
|----|-------------------|------------|
| 1 | Emergencia médica | 1 |
| 2 | Consulta médica | 2 |

| subject_id | attr1 | attr2 |
|------------|-------|---|
| 1 | 30 | Ruta 5 KM 36500, Canelones, Uruguay |
| 2 | 57 | Julio Herrera y Reising 738, Montevideo, Montevideo |

| event_id | attr3 | attr4 |
|----------|-------|------------|
| 1 | Rojo | 2024-03-30 |
| 2 | Verde | 2023-07-15 |

| attribute_name | label | required |
|----------------|---------------|----------|
| attr1 | Edad | TRUE |
| attr2 | Dirección | TRUE |
| attr3 | Clasificación | TRUE |
| attr4 | Fecha | FALSE |

Figura 4.4: Tablas de base de datos para el segundo enfoque.

Luego de realizada la evaluación de las opciones, se optó por el primer enfoque debido a su mayor flexibilidad, ya que no es necesario cargar todos los atributos desde el inicio. Esto permite que se carguen únicamente los atributos que realmente se vayan a utilizar, evitando la necesidad de realizar modificaciones estructurales complejas como *ALTER TABLE*, las cuales no son recomendables en entornos de producción debido a los riesgos asociados. Además, este modelo ofrece una escalabilidad superior, permitiendo que el sistema crezca y se adapte a nuevos requerimientos sin afectar su estructura. Al evitar operaciones que podrían causar tiempos de inactividad, se asegura una mayor continuidad del servicio. Asimismo, el enfoque reduce el riesgo de errores que podrían comprometer la integridad de los datos o la estabilidad del sistema, haciendo que sea una opción más segura y eficiente para el proyecto.

En el diseño final del modelo conceptual de la Figura 4.5, se decidió unificar al sujeto y sus atributos en una única entidad. De este modo, cada fila del archivo CSV representa un sujeto asociado a un evento, lo que facilita la carga de datos y elimina la necesidad de manejar múltiples archivos CSV con identificadores para relacionar eventos y sujetos, o de realizar procesos adicionales para separarlos. Además, se evita que el usuario deba definir explícitamente qué atributos pertenecen al sujeto y cuáles al evento. Por ejemplo, la edad podría considerarse un atributo del sujeto; sin embargo, si existen dos eventos asociados al mismo sujeto pero con edades diferentes, la edad se convierte en un atributo

del evento. Por esta razón, se optó por unificar ambos elementos, minimizando así posibles confusiones y asegurando una mayor coherencia en la gestión de los datos.

En la entidad “Subject”, se representan los aspectos geográficos del sujeto asociado a un evento, almacenando las coordenadas en un formato de punto (POINT). Además, se incluyó la especificación de ciertos valores para los atributos clave que el sistema requiere para funcionar, como la dirección y el identificador del usuario.

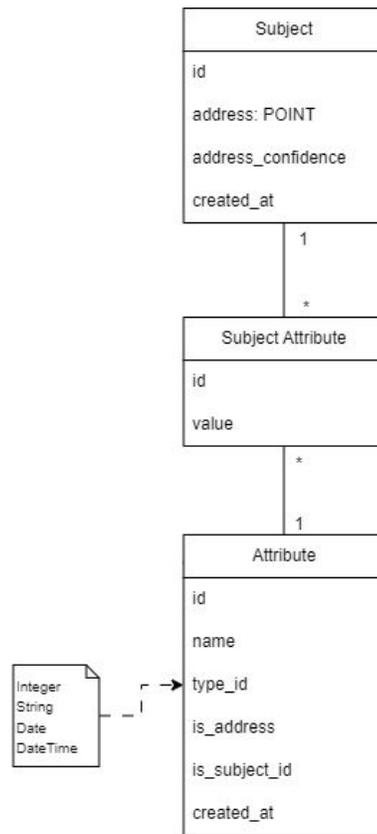


Figura 4.5: Diseño final del modelo conceptual.

A modo de ejemplo, en la Figura 4.6 se presentan las tablas de la base de datos que darán soporte al diseño final elegido.

Tabla: Subject

| id | address | address_confidence | created_at |
|----|-------------------------------|--------------------|------------|
| 1 | POINT(-34.605270, -56.260414) | Low | 2024-01-19 |
| 2 | POINT(-34.915110, -56.16586) | High | 2024-01-19 |

Tabla: Subject_Attribute

| id | value | attribute_id | subject_id |
|----|---|--------------|------------|
| 1 | 30 | 1 | 1 |
| 2 | Ruta 5 KM 36500, Canelones, Uruguay | 2 | 1 |
| 3 | Rojo | 3 | 1 |
| 4 | 2024-03-30 | 4 | 1 |
| 5 | 12345678 | 5 | 1 |
| 6 | 57 | 1 | 2 |
| 7 | Julio Herrera y Reising 738, Montevideo, Montevideo | 2 | 2 |
| 8 | Verde | 3 | 2 |
| 9 | 2023-07-15 | 4 | 2 |
| 10 | 51234567 | 5 | 2 |

Tabla: Attribute

| id | name | is_address | is_subject_id | type | created_at |
|----|---------------|------------|---------------|---------|------------|
| 1 | Edad | FALSE | FALSE | Integer | 2024-01-19 |
| 2 | Dirección | TRUE | FALSE | String | 2024-01-19 |
| 3 | Clasificación | FALSE | FALSE | String | 2024-01-19 |
| 4 | Fecha | FALSE | FALSE | Date | 2024-01-19 |
| 5 | Cédula | FALSE | TRUE | Integer | 2024-01-19 |

Figura 4.6: Tablas de base de datos para el diseño final.

4.1.2. Geocodificación de las direcciones

Cada dominio puede representar una dirección de diversas maneras, con diferentes formas normales que pueden estar representadas por uno o varios *tokens*. Para garantizar la mayor precisión posible en el sistema, se optó por predefinir varias formas normales que facilitarían la ubicación de las direcciones. Se definieron seis formas normales en total, cada una diseñada para capturar diferentes aspectos de una dirección y mejorar la capacidad de geocodificación. Cabe destacar que las primeras tres formas normales están basadas en el Modelo de Direcciones de la IDEuy, mencionado en la Sección 2.2.

- **Opción 1:** Departamento, Localidad, Calle/Ruta, Número/Km/Manzana Solar.
- **Opción 2:** Departamento, Calle/Ruta, Número/Km/Manzana Solar.
- **Opción 3:** Departamento, Dirección.
- **Opción 4:** Coordenadas geográficas.
- **Opción 5:** Puntos de interés.
- **Opción 6:** Personalizada.

En la última opción, no se puede garantizar que el geocodificador encuentre la dirección correcta, ya que los *tokens* de la dirección se crean manualmente y pueden estar en un orden que el geocodificador no interprete correctamente.

Una vez completado el proceso de geocodificación, es necesario determinar si se requiere una ubicación manual en caso de que este haya fallado. Los geocodificadores pueden o no incluir un identificador que indique la precisión de las coordenadas obtenidas para una dirección. Por ello, fue necesario realizar un mapeo para cada *Geocoder*, definiendo el “nivel de confianza” en la ubicación, con valores de alta, media o baja, para determinar si es necesario realizar una ubicación manual.

En el caso del servicio de la IDEuy, la precisión se determina mediante un indicador específico llamado *stateMsg*. Si este indicador devuelve el valor “Aproximado”, la confianza es considerada media; si no devuelve ninguna coordenada o si devuelve las coordenadas latitud: 0 y longitud: 0, la confianza es baja; en caso contrario, la confianza es alta. Por otro lado, *Nominatim* no cuenta con un identificador de precisión específico, por lo que si no devuelve ninguna coordenada o devuelve las coordenadas latitud: 0 y longitud: 0, la confianza es baja; en caso contrario, es alta.

En el Código 4.1, se muestra un ejemplo de un resultado obtenido del servicio de la IDEuy.

```
[
  {
    "type": "CALLEYPORTAL",
    "id": "879692",
    "address": "JULIO HERRERA Y REISSIG 48",
    "idCalle": 22845,
    "nomVia": "JULIO HERRERA Y REISSIG",
    "postalCode": "20162",
    "idLocalidad": 61,
    "localidad": "PUNTA DEL ESTE",
    "idDepartamento": 23,
    "departamento": "MALDONADO",
    ...
    "lat": -34.94533280947,
    "lng": -54.93368534375155,
    ...
    "stateMsg": "Aproximado",
    "source": "ide_uy",
    "ranking": 13.6,
    "state": 2
  }
]
```

Código 4.1: Respuesta obtenida del servicio de la IDEuy.

En este ejemplo, el campo *stateMsg* indica que la localización es aproximada, lo que implica que la confianza en la precisión de los datos es media.

Según el nivel de confianza, se decidió utilizar un código de colores para representar los resultados obtenidos, facilitando así la interpretación de los datos y mejorando la experiencia del usuario. El código de colores es el siguiente: rojo para baja confianza, amarillo para confianza media y verde para alta confianza.

Otra decisión orientada a mejorar la experiencia del usuario fue la implementación de un mecanismo que facilita la ubicación manual de puntos en el mapa. En situaciones de baja confianza, se asigna una coordenada de referencia basada en la forma normal utilizada. Si la forma normal incluye el Departamento, se utiliza la coordenada correspondiente a la capital departamental; de lo contrario, se asigna la coordenada de Montevideo. Estas coordenadas de referencia son previamente obtenidas mediante geocodificación con el servicio de la IDEuy y almacenadas en el sistema, lo que evita consultas repetitivas y agiliza la ubicación manual del marcador por parte del usuario.

Además, si la forma normal seleccionada es “Puntos de Interés” y no se ha cargado un archivo CSV con estos puntos, se realiza una búsqueda en el geocodificador para evitar que todas las direcciones tengan baja confianza debido a la falta de datos. Este enfoque garantiza que, incluso sin puntos de interés pre-cargados, el sistema pueda ofrecer resultados precisos y útiles.

Todas las decisiones tomadas respecto a la geocodificación de las direcciones se orientaron a garantizar que la interfaz fuera intuitiva, facilitando al usuario la identificación de las direcciones no ubicadas y simplificando el proceso de corrección manual para encontrar la ubicación correcta.

4.1.3. Formato de los datos importados

En los datos espaciales, es importante definir un único SRID para garantizar la compatibilidad y coherencia de los datos. Para ofrecer flexibilidad en la gestión de datos, se optó por permitir a los usuarios cargar coordenadas y puntos de interés en cualquier SRID que elijan. Sin embargo, para asegurar la interoperabilidad y la consistencia en el manejo de datos espaciales, todos los datos importados se traducen internamente al SRID 4326, que es el estándar utilizado por el servicio de la IDEuy.

4.2. Arquitectura

Se definieron tres componentes claves que forman la base del sistema.

- **Frontend para Analistas:** Interfaz diseñada para que los analistas puedan cargar e interactuar con los datos, realizar consultas específicas y utilizar herramientas para visualización en el mapa.
- **Frontend para Administradores:** Interfaz que ofrece las principales vistas para la administración de usuarios y configuración de los *workspaces*.
- **Backend:** Responsable de la lógica de negocio y la gestión de la base de datos espacial, proporcionando el soporte necesario para las operaciones del frontend.

Esta estructura garantiza que cada componente del sistema se enfoque en sus responsabilidades específicas, facilitando la interacción entre usuarios y datos, así como la integración con servicios externos.

4.2.1. Arquitectura de los frontends

Se empleó una arquitectura Cliente-Servidor para los frontends, lo que permite una mayor escalabilidad y eficiencia, al distribuir la carga de procesamiento y renderizado entre el cliente y el servidor, lo que mejora significativamente la experiencia del usuario.

A continuación, se describen las capas:

- **Cliente:** Es donde se definen las páginas del sistema y el *JavaScript* que se ejecuta en el navegador del usuario.
- **Servidor:** Gestiona las interacciones con el backend y los servicios externos.

4.2.2. Arquitectura del backend

Para el backend, se utilizó la arquitectura *Clean*, la cual promueve la separación de responsabilidades en diferentes capas, aislando los detalles de implementación (como bases de datos y bibliotecas externas) de la lógica de negocio. Esta separación facilita la independencia de capas, permitiendo que los cambios en una no afecten a las demás. Además, facilita el testeado individual de cada componente y aumenta tanto la escalabilidad como la mantenibilidad del sistema, creando un código más limpio y organizado que puede adaptarse y expandirse con mayor facilidad a lo largo del tiempo.

A continuación, se describen las capas:

- **Presentation Layer:** Contiene los controladores (*controllers*) que reciben y gestionan las peticiones realizadas por los frontends, retornando la información correspondiente.

- **Domain Layer:** Contiene las reglas de negocio de la aplicación. Esta es la capa central, independiente de las tecnologías externas, e incluye:
 - **Use Cases:** Son los que realizan las operaciones de negocio. Los mismos definen qué puede hacer la aplicación según las reglas de negocio y los datos recibidos.
 - **Entities:** Son las clases que representan objetos clave de la lógica de negocio, con sus atributos y su comportamiento.
 - **Interfaces:** Son las interfaces que la capa de datos implementa para interactuar con el dominio, manteniendo la independencia de los detalles externos.
- **Data Layer:** Posee las implementaciones para acceder a fuentes de datos externas, como la base de datos y los servicios externos (*Geocoders*).
 - **Repositories:** Implementaciones que interactúan con los DAOs o los geocodificadores para obtener y guardar información. Implementan las interfaces definidas en la capa de dominio.
 - **Data Access Objects (DAOs):** Implementan las operaciones concretas para la interacción con la base de datos.

4.2.3. Sistemas e integraciones externas

Para responder a los requerimientos de interoperabilidad, se decidió utilizar *GeoServer*, el cual permite la gestión y visualización eficiente de datos espaciales mediante estándares del OGC.

Además, se integraron servicios externos para funcionalidades específicas:

- *Geocoders*: Servicios utilizados para la geocodificación de direcciones, elegidos durante el análisis.
- *ID Uruguay*: Mecanismo de autenticación que facilita la gestión segura de usuarios mediante estándares *OAuth*.

En la Figura 4.7 se puede observar la interacción entre los distintos componentes descritos.

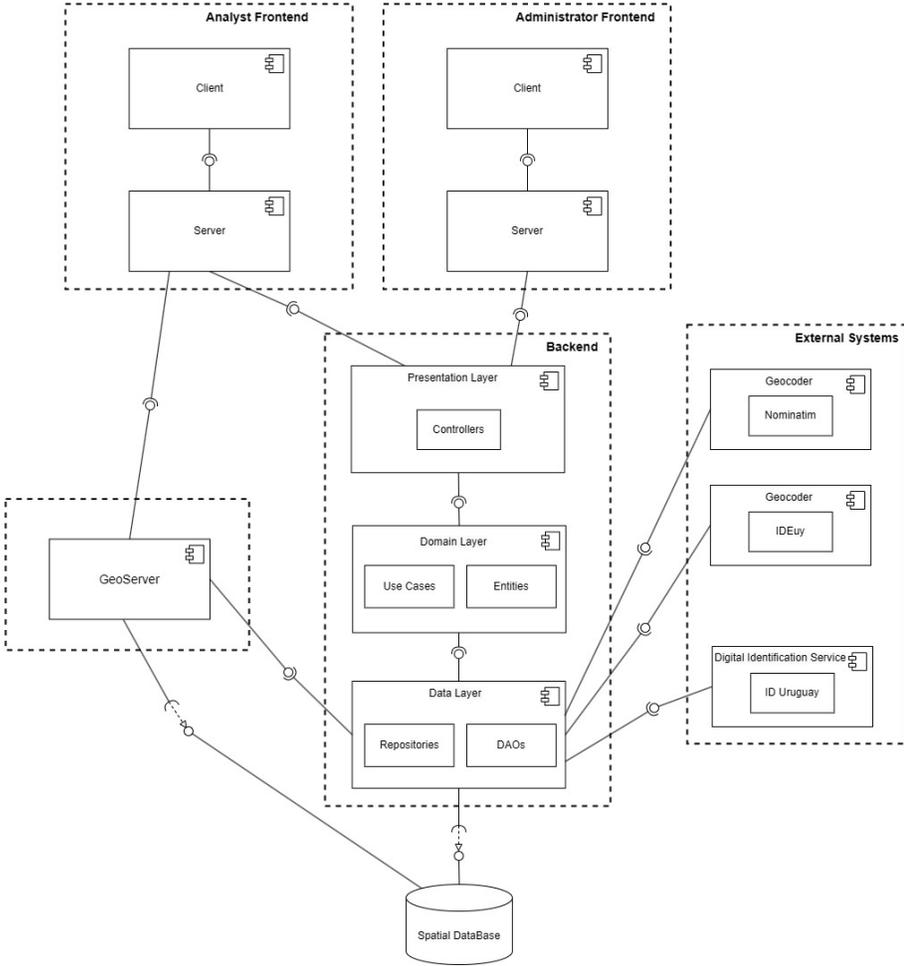


Figura 4.7: Vista Lógica.

4.2.4. Procesamiento con cola de mensajes

Para optimizar el rendimiento durante la carga masiva de datos, se diseñó el uso de una cola de mensajes, permitiendo el procesamiento asíncrono.

Una cola de mensajes es una tecnología que facilita la comunicación rápida y asíncrona entre aplicaciones mediante el intercambio de mensajes. Esta tecnología garantiza que los mensajes sean entregados de manera confiable y, en caso de que no puedan ser procesados de inmediato, los almacena de forma persistente hasta que el destinatario esté disponible. Las colas siguen una política *First In, First Out (FIFO)*, lo que asegura que los mensajes se procesen en el mismo orden en que fueron enviados (Hohpe y Woolf, 2003).

El proceso de transmisión de un mensaje sigue estos pasos:

1. **Crear:** La aplicación emisora genera el mensaje con los datos necesarios.
2. **Enviar:** La aplicación emisora envía el mensaje a un canal.
3. **Entregar:** El sistema de mensajería recibe el mensaje y lo dirige a la aplicación receptora.
4. **Procesar:** La aplicación receptora extrae los datos del mensaje y los utiliza.

Este mecanismo es útil para manejar cargas masivas de datos sin comprometer el rendimiento, es tolerante a fallos y asegura la persistencia de los mensajes.

En la Figura 4.8, se detalla el flujo del procesamiento. Cabe mencionar que las formas normales de dirección que se mencionan corresponden a las definidas en la Sección 4.1.2, las cuales facilitan la geocodificación de los datos.

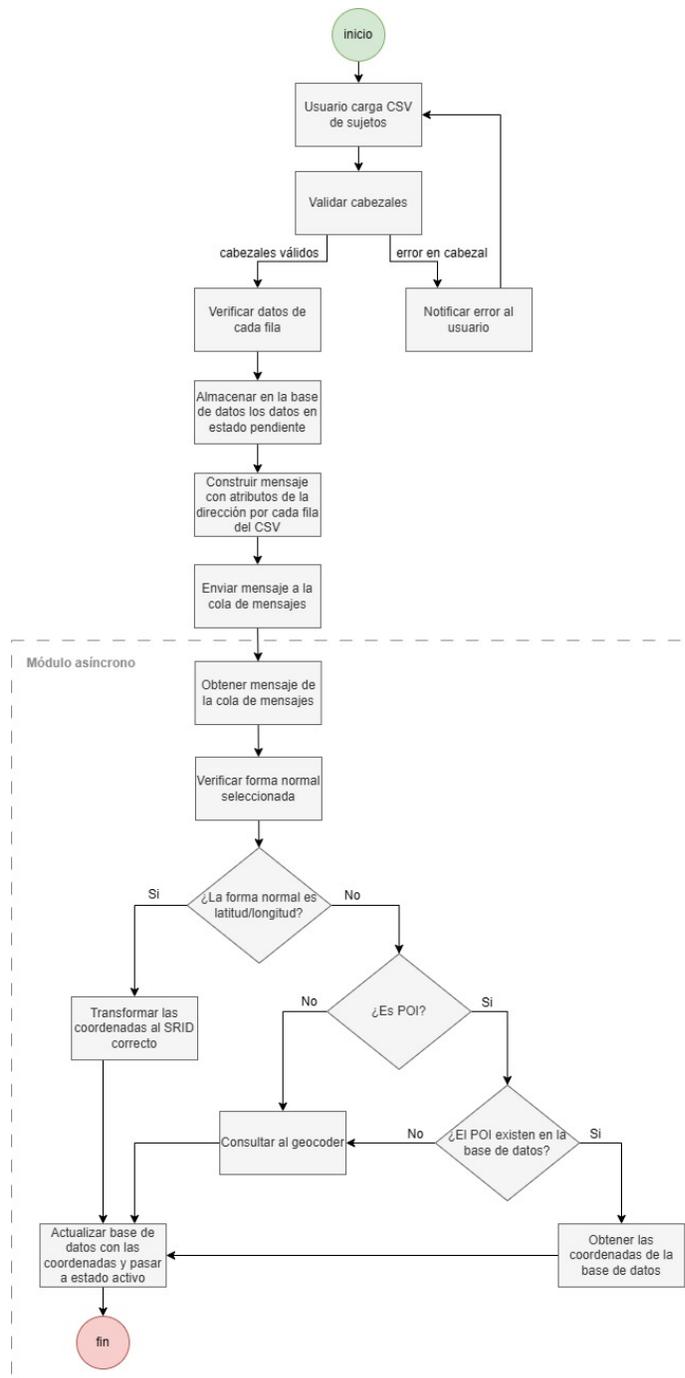


Figura 4.8: Flujo del procesamiento de datos y uso de la cola de mensajes.

Capítulo 5

Implementación

En este capítulo se explican los detalles de la implementación. Primero, se presentan las tecnologías y librerías utilizados para llevarla a cabo, seguido de una descripción del desarrollo.

5.1. Tecnologías utilizadas

Las tecnologías y librerías utilizadas en este proyecto fueron seleccionadas considerando la experiencia previa y la relevancia de cada herramienta en el ámbito del desarrollo web moderno. Se optó por tecnologías que no solo fueran actuales, sino que también contaran con una buena reputación en comunidades y foros de desarrollo.

- **Frontend para Analistas y para Administradores**

Para la implementación, se empleó *Next.js*¹, un framework de código abierto que simplifica la creación de aplicaciones web basadas en *React*, permitiendo el renderizado del lado del servidor.

En cuanto al manejo de mapas y visualización de datos geográficos, se utilizó *Leaflet*² junto con la librería *React-Leaflet*³, lo que asegura una integración fluida en aplicaciones *React*.

Como mapa base, se eligió *OpenStreetMap*⁴, una solución gratuita y detallada para la visualización de mapas, que ha sido ampliamente adoptada por varios de los visualizadores analizados.

Para llevar a cabo las funciones geoespaciales, se empleó la librería *Turf.js*⁵. En particular, se utilizó la función *pointsWithinPolygon* para determinar la cantidad de puntos ubicados dentro de una geometría poligonal, lo cual es

¹<https://nextjs.org/>

²<https://leafletjs.com/>

³<https://react-leaflet.js.org/>

⁴<https://www.openstreetmap.org/>

⁵<https://turfjs.org/>

fundamental para evaluar la distribución de los datos en áreas específicas. La función *distance* se utilizó para calcular la distancia entre dos puntos espaciales, y *nearestPoint* permitió identificar el punto más cercano a una ubicación dada dentro de un conjunto de puntos.

Para gestionar el estado en la visualización de los datos espaciales en el mapa, incluyendo las capas cargadas, las capas con puntos que representan a los sujetos y las capas resultantes de las funciones geoespaciales, se utilizó la librería *Zustand*⁶. Esta herramienta permitió controlar de manera eficiente la activación y desactivación de las capas, facilitando su visualización en tiempo real y mejorando la experiencia interactiva del usuario.

Finalmente, para la representación gráfica de datos, se optó por la librería *ECharts.js*⁷, conocida por su versatilidad y capacidad para crear gráficos interactivos y altamente personalizables.

■ Backend

El backend fue desarrollado utilizando *Spring Boot*⁸, un framework de código abierto basado en *Java* que simplifica la configuración y el despliegue de aplicaciones, permitiendo así concentrarse en la implementación de la lógica del negocio.

Para gestionar la cola de mensajes, se utilizó *Wildfly*⁹, que ofrece una plataforma robusta para la configuración y administración de colas de mensajes a través de *JMS-queue*.

En cuanto a las pruebas unitarias, se emplearon *JUnit Jupiter*¹⁰ y *Mockito*¹¹ para garantizar la calidad y funcionalidad del código.

■ Base de datos

Para el almacenamiento de los datos, se utilizó la base de datos *PostgreSQL*¹² junto con la extensión *PostGIS*¹³, conocida por sus avanzadas capacidades en la gestión de datos espaciales y su amplia adopción en proyectos geoespaciales.

⁶<https://zustand.docs.pmnd.rs/getting-started/introduction>

⁷<https://echarts.apache.org>

⁸<https://spring.io/projects/spring-boot>

⁹<https://www.wildfly.org/>

¹⁰<https://junit.org/junit5/docs/current/user-guide/>

¹³<https://postgis.net/>

■ GeoServer

Como se mencionó en el capítulo de análisis, se utilizó *GeoServer* para el manejo de datos espaciales. Esto requirió configurar la conexión entre *GeoServer* y la base de datos, permitiendo la interacción directa con los datos geoespaciales.

Para la visualización de datos en el cliente, se empleó el protocolo WFS, ya que facilita la manipulación de datos devueltos por *GeoServer*, lo cual es útil tanto para ejecutar funciones geoespaciales como para personalizar los marcadores mostrados en el mapa.

Para la aplicación de filtros avanzados, fue necesario instalar la extensión *QueryLayer*, debido a las limitaciones de WFS en el filtrado de datos en combinación con otras capas geográficas. Esta extensión permite realizar consultas con *Structured Query Language (SQL)* sobre las capas geoespaciales, facilitando así la manipulación directa de datos desde la base de datos.

A continuación, se presentan dos ejemplos de consultas geoespaciales. El primero muestra cómo se seleccionan puntos ubicados dentro de un radio determinado desde un punto central (ver Código 5.1), mientras que el segundo muestra puntos que se encuentran dentro de un área específica (ver Código 5.2).

```
cql_FILTER = DWITHIN(  
    address,  
    querySingle('uruguay1', 'the_geom',  
                'IN(''uruguay1.1'')'),  
    15,  
    kilometers)
```

Código 5.1: Ejemplo de selección de elementos a una distancia desde un punto de referencia.

- `DWITHIN(address, querySingle(...), 15, kilometers)`: Esta función selecciona elementos espaciales que se encuentren a una distancia máxima de 15 kilómetros del punto de referencia.
 - `address`: Es el nombre de la columna que contiene las coordenadas de los sujetos.
- `querySingle('uruguay1', 'the_geom', 'IN(''uruguay1.1'')')`: Esta subconsulta busca una geometría dentro de la capa cargada llamada `uruguay1`, que va a ser el punto de referencia.
 - `the_geom`: Es la columna que contiene las geometrías en la capa.
 - `IN(''uruguay1.1'')`: Selecciona una entidad específica de la capa que coincida con el identificador `uruguay1.1`.

```
cql_FILTER = WITHIN(  
    address ,  
    querySingle('uruguay1', 'the_geom',  
                'IN(''uruguay1.1'')')  
)
```

Código 5.2: Ejemplo de filtros de puntos que se encuentran dentro de un área específica.

- `WITHIN(address, querySingle(...))`: Esta función selecciona elementos espaciales que se encuentren dentro de un área de referencia. Con la subconsulta `querySingle`, al igual que en el anterior ejemplo, se obtiene la geometría del área de referencia.

5.2. Decisiones tecnológicas

En esta sección se describen dos decisiones tecnológicas tomadas durante el desarrollo del prototipo. La primera se enfoca en la carga de las capas geográficas, mientras que la segunda abarca el uso de funciones geoespaciales.

5.2.1. Carga de capas

Para mantener una clara separación de responsabilidades en la arquitectura de la plataforma, se decidió almacenar y gestionar las capas geográficas en *GeoServer*, tal como se estableció en la etapa de diseño. Esto desliga al backend de la gestión directa de las capas, concentrando sus recursos en la lógica de negocio y en el procesamiento de datos durante las cargas. *GeoServer*, al estar diseñado específicamente para gestionar datos espaciales, garantiza una mayor eficiencia y especialización en el manejo de las capas. Por ello, se optó por cargar las capas en formato *Shapefile (SHP)*, que es soportado nativamente por *GeoServer*, lo que facilita la integración.

Durante las pruebas de carga de capas, surgieron varios casos que podían afectar la integración del sistema. Por ejemplo, se detectaron archivos ZIP de SHP mal formados y otros archivos que no incluían correctamente el SRID. Esta última situación resultaba particularmente problemática, ya que, aunque la capa se cargaba con éxito en *GeoServer*, su visualización fallaba al intentar mostrarla en el mapa, complicando el diagnóstico de los errores. Para abordar estos problemas, se implementaron controles adicionales que verifican la validez de los archivos, solicitando el SRID al momento de la carga y garantizando que la capa sea funcional tras la carga.

Asimismo, se definió un proceso para identificar de forma unívoca cada entidad dentro de la capa. Una vez cargada en *GeoServer*, se listan los atributos disponibles para que el usuario pueda seleccionar uno como identificador principal. Esto garantiza que cada entidad pueda ser referenciada de manera precisa en

el análisis posterior. Por ejemplo, en una capa que representa los departamentos de Uruguay, se podría seleccionar el atributo “nombre” como identificador. De esta forma, al listar las entidades de la capa, se verían sus nombres específicos, como “Montevideo”, “Río Negro”, entre otros.

5.2.2. Funciones geospaciales

Se decidió utilizar *Turf.js*, a nivel de cliente, para llevar a cabo las funciones espaciales. Esta decisión sigue el mismo principio de separar responsabilidades, evitando sobrecargar al backend o a la base de datos, que están enfocados en procesar la lógica de negocio.

Dado que las funciones espaciales se activan de manera puntual durante el análisis, delegarlas al cliente minimiza el impacto en la performance general del sistema. Esto asegura que la plataforma mantenga su eficiencia sin comprometer el rendimiento del backend.

5.3. Despliegue en la nube

Para el despliegue en la nube, se utilizó el servicio de *Elastic Cloud* de Antel. Este servicio *PaaS* simplifica el proceso de despliegue al eliminar la necesidad de realizar instalaciones y configuraciones complejas. Se configuraron cuatro entornos distintos para gestionar los diversos componentes de la aplicación. Además, para garantizar la seguridad, se emplearon direcciones IP LAN internas para la interconexión de estos entornos. *Elastic Cloud* también ofrece escalabilidad, permitiendo asignar recursos adicionales según sea necesario, así como implementar balanceadores de carga y duplicar servidores para mejorar la disponibilidad y el rendimiento.

- **Entorno del frontend de Analistas:** Se creó un entorno dedicado exclusivamente al frontend de los usuarios analistas, donde se desplegó la aplicación en un servidor *Node.js*.
- **Entorno del frontend de Administradores:** De manera similar al entorno para analistas, se configuró un entorno con un servidor *Node.js* dedicado exclusivamente para los administradores de la plataforma.
- **Entorno del backend:** Se configuró un entorno específico para el backend, que incluye un servidor *Wildfly* y una base de datos *PostgreSQL* con la extensión *PostGIS*.
- **Entorno del GeoServer:** Se creó un entorno separado para el *GeoServer*, con un servidor *Tomcat*.

Se decidió utilizar el servicio de *Nominatim* a través de su API para la geocodificación, debido a su conveniencia. Sin embargo, también se contempla la opción de configurar un entorno propio e instalar *Nominatim* localmente si es

necesario.

En la Figura 5.1 se puede apreciar la representación de la Vista de Despliegue, mostrando la interacción entre los diferentes componentes, las tecnologías empleadas y los distintos tipos de servidores utilizados.

Los detalles específicos del despliegue, incluyendo las versiones utilizadas y las configuraciones necesarias para desplegarlo en la nube, están disponibles en el anexo correspondiente al manual de instalación.

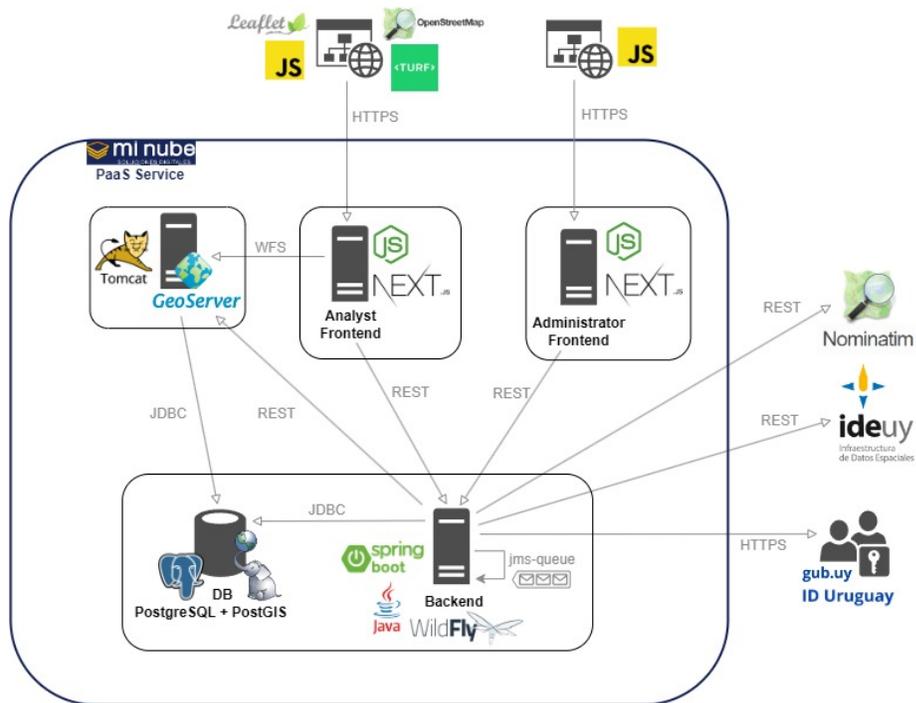


Figura 5.1: Vista de Despliegue.

5.4. Problemas encontrados y soluciones implementadas

En esta sección se presentan los principales problemas encontrados durante el desarrollo y las soluciones implementadas para resolverlos. A continuación, se abordan dos casos: la integración con el *GeoServer* y los problemas con la tecnología de la cola de mensajes debido a restricciones en el entorno de despliegue.

5.4.1. GeoServer

Durante la implementación, surgió un problema significativo en la integración entre *GeoServer* y la base de datos debido a una incompatibilidad entre el modelo conceptual diseñado y las expectativas del *GeoServer*. Al conectarse con la base de datos, *GeoServer* extraía la información geográfica de la tabla “Subject” (como se muestra en la Figura 5.2), pero dicha tabla no contiene el resto de la información necesaria para aplicar los filtros requeridos, lo que dificultaba el análisis adecuado de los datos.

Tabla: Subject

| id | address | address_confidence | created_at |
|----|-------------------------------|--------------------|------------|
| 1 | POINT(-34.605270, -56.260414) | Low | 2024-01-19 |
| 2 | POINT(-34.915110, -56.16586) | High | 2024-01-19 |

Figura 5.2: Ejemplo de la tabla de *subjects*.

Para superar esta limitación, fue necesario crear vistas a partir de consultas SQL en la base de datos y publicar capas (*layers*) basadas en estas vistas, lo que permitió realizar consultas filtradas de manera efectiva. En la Figura 5.3, se muestra un ejemplo de como se estructura la vista resultante.

Tabla: workspace_subjects_1

| subject_id | address | Edad | Dirección | Clasificación | Fecha | Cédula |
|------------|-------------------------------|------|---|---------------|------------|----------|
| 1 | POINT(-34.605270, -56.260414) | 30 | Ruta 5 KM 36500, Canelones, Uruguay | Rojo | 2024-03-30 | 12345678 |
| 2 | POINT(-34.915110, -56.16586) | 57 | Julio Herrera y Reising 738, Montevideo, Montevideo | Verde | 2023-07-15 | 51234567 |

Figura 5.3: Ejemplo de la vista de *subjects* de un *workspace* particular.

Además, cada vez que se añaden nuevos atributos a la base de datos, es necesario reconfigurar las vistas y actualizar las capas a través de la API REST de *GeoServer*, ya que el sistema no detecta automáticamente los cambios en la estructura de las tablas. Este procedimiento no estaba detallado en la documentación oficial de *GeoServer*, por lo que fue necesario realizar una investigación exhaustiva para encontrar la solución adecuada.

5.4.2. Cola de mensajes

Inicialmente, se optó por *RabbitMQ* como tecnología para la cola de mensajes, debido a su facilidad de instalación en entornos locales, lo que permitió que el desarrollo avanzara basándose en esta implementación. No obstante, al intentar desplegar el sistema en *Elastic Cloud*, surgió un inconveniente clave: no se disponía de permisos de *sudo* en el servidor, lo que impidió la instalación de *RabbitMQ*.

Elastic Cloud ofrece importantes ventajas, como se mencionó en la sección anterior, destacándose la simplicidad para desplegar aplicaciones *Spring Boot*. Sin embargo, la falta de permisos de *sudo* implica que ciertos servicios y configuraciones no pueden instalarse ni gestionarse manualmente. Aunque era posible obtener estos permisos contratando un servicio más cercano a un modelo de *Infrastructure as a Service (IaaS)* en lugar de *PaaS*, como es el caso de *Elastic Cloud*, esto habría requerido migrar todo el despliegue ya realizado, además de gestionar un servidor Linux desde cero, incluyendo instalaciones y configuraciones manuales.

Debido a estas limitaciones, se decidió cambiar la tecnología de la cola de mensajes a *Java Message Service (JMS)* y utilizar el servidor de aplicaciones *WildFly* en lugar de la configuración original. Esta decisión implicó algunos cambios:

- Reconfiguración del entorno para adaptarlo a la nueva arquitectura *WildFly* con *JMS*.
- Ajustes en la instalación local para alinear el entorno con la nueva infraestructura en la nube.
- *WildFly* requiere más recursos que el servidor *Spring Boot*, por lo que fue necesario aumentar la asignación de RAM y espacio en disco en *Elastic Cloud* para garantizar un funcionamiento adecuado.

Estos ajustes permitieron superar las limitaciones técnicas y lograr una implementación estable del sistema en el entorno productivo.

5.5. Pruebas

En cuanto a las pruebas de la aplicación, se realizaron pruebas unitarias en el backend, alcanzando una cobertura mayor al 80 % del código. Además, durante

todo el proceso de desarrollo, se llevaron a cabo revisiones de código mediante la herramienta de *pull requests* en *GitHub*, lo que permitió promover la calidad del código y fomentar una colaboración efectiva.

También se llevaron a cabo pruebas de sistema, evaluando los distintos flujos de la aplicación. Inicialmente se probaron los flujos críticos, seguidos por el resto de los flujos, para asegurar el correcto funcionamiento integral de la aplicación.

Durante las reuniones con los tutores, se realizó una validación continua de la aplicación. Los tutores ofrecieron sugerencias para ajustes y mejoras, y validaron el desarrollo para asegurar el cumplimiento de los requisitos. A partir de estas validaciones, se identificó la necesidad de mejorar la accesibilidad y comprensibilidad de la interfaz gráfica de usuario (GUI). En respuesta, se implementó un botón de ayuda que incluye un tour interactivo para guiar a los usuarios en la creación de atributos y la carga de datos. También se elaboraron manuales de usuario para facilitar el uso de la plataforma.

5.5.1. Pruebas de performance de carga de datos

La carga de datos es una de las partes más críticas de la aplicación, ya que está diseñada para manejar volúmenes de información sin comprometer la funcionalidad del sistema. Para asegurar un rendimiento adecuado bajo estas condiciones, se realizaron pruebas orientadas a optimizar el procesamiento de datos mediante el uso de la cola de mensajes. Durante estas pruebas, se evaluó el equilibrio entre la cantidad de consumidores y el uso de memoria, dado que incrementar el número de consumidores acelera el procesamiento, pero conlleva un mayor consumo de recursos. Este análisis permitió identificar el balance óptimo entre la capacidad de procesamiento y la eficiencia en el uso de los recursos del sistema.

En las pruebas de carga, se identificaron dos tipos principales de datos: uno relacionado con la búsqueda de direcciones a través de un geocodificador externo y otro centrado en la búsqueda de direcciones dentro de los Puntos de Interés (POIs). El primero, al depender de un sistema externo, mostró tiempos de respuesta variables, lo que obligó a ajustar los tiempos de espera (*timeouts*) en las consultas y la cantidad de consumidores (hilos), ya que el servidor del servicio de la IDEuy no soportaba un número elevado de solicitudes simultáneas y comenzaba a devolver errores. En contraste, la segunda prueba, al trabajar con datos locales predefinidos, tuvo tiempos de procesamiento significativamente menores.

En las tablas se detallan las configuraciones aplicadas. Si tras tres intentos no se recibía una respuesta del geocodificador, se consideraba que había ocurrido un *timeout*.

- Configuración de la prueba: Cantidad de sujetos: 200, tiempo de espera: 30 segundos, espera entre reintentos: 1 segundo, cantidad de reintentos: 3, *Geocoder*: Servicio de la IDEuy (Ver Tabla 5.1).

| Cantidad de hilos | Tiempo total | Cantidad de timeouts | Reintentos |
|-------------------|--------------|----------------------|------------|
| 2 a 4 hilos | 9 min | 6 | 30 |
| 2 a 3 hilos | 10 min | 5 | 36 |
| 1 a 2 hilos | 42 min | 46 | 141 |
| 1 único hilo | 11 min | 0 | 0 |

Tabla 5.1: Primera prueba de performance.

En el análisis de las direcciones que presentaban demoras, se observó que al aumentar la cantidad de hilos se generaban más solicitudes simultáneas al geocodificador, lo que resultaba en un empeoramiento de los tiempos de respuesta. Esto degradaba el rendimiento del sistema externo y provocaba *timeouts* en varias solicitudes. También se realizaron pruebas reduciendo el tiempo de espera (*timeout*), lo que causó que la mayoría de las solicitudes fallaran.

Por lo tanto, aunque aumentar los hilos y reducir el tiempo de espera disminuye el tiempo de procesamiento, este enfoque provoca que los sujetos sean ubicados en direcciones por defecto, perdiendo así el valor de utilizar el geocodificador, ya que no se reciben respuestas en la mayoría de los casos.

Basado en las pruebas realizadas y las limitaciones del sistema externo de la IDEuy, se decidió utilizar un único hilo (un solo consumidor), con la siguiente configuración: 30 segundos de tiempo de espera, 3 reintentos por solicitud y 1 segundo de espera entre cada intento.

Para las pruebas con direcciones POI, se probó configuraciones sin hilos (ver Tabla 5.2) y con 1 a 2 hilos (ver Tabla 5.3), obteniendo los siguientes resultados:

| Cantidad de sujetos | Tiempo total |
|---------------------|--------------|
| 200 | 4 seg |
| 1000 | 20 seg |
| 10000 | 3 min 10 seg |

Tabla 5.2: Prueba con direcciones POI sin hilos.

| Cantidad de sujetos | Tiempo total |
|---------------------|--------------|
| 200 | 2 seg |
| 1000 | 12 seg |
| 10000 | 2 min |

Tabla 5.3: Prueba con direcciones POI con 1 a 2 hilos

En este caso, los resultados obtenidos en las pruebas con múltiples hilos mostraron un mejor rendimiento en los tiempos de procesamiento en comparación

con el uso de un único hilo. Sin embargo, dado que el uso de hilos genera un impacto negativo significativo en las solicitudes hacia los geocodificadores, se decidió optar por una política de procesamiento sin hilos, a pesar de que esto reduce la efectividad en este caso de uso.

5.5.2. Pruebas de performance de geocodificadores

Para las pruebas de los geocodificadores, se utilizaron datos abiertos de accidentes de tránsito en todo el país, lo que permitió contar con un listado considerable de direcciones reales en Uruguay. En total, se seleccionaron 200 datos de sujetos, asegurando que hubiera al menos una dirección por cada departamento. La prueba fue realizada en ambos geocodificadores utilizando estos datos y se obtuvieron los siguientes resultados (ver Tabla 5.4):

| Confianza | IDEuy | Nominatim |
|-----------|--------|-----------|
| Alta | 59,5 % | 3 % |
| Media | 5 % | 0 % |
| Baja | 35,5 % | 97 % |

Tabla 5.4: Prueba de performance con 200 datos de sujetos en el Uruguay.

Tal como indicaban los estudios previos, las direcciones ubicadas por el servicio de la IDEuy presentaron un mayor porcentaje de confianza en comparación con *Nominatim*, alineándose con los resultados obtenidos durante la etapa de investigación del estado del arte.

La menor tasa de éxito en la geolocalización podría estar relacionada con el formato de las direcciones. La forma normal utilizada fue la *Opción 3*, que incluía el departamento, y las direcciones en el CSV estaban en formato “calle esquina calle” o “ruta km”, lo que también pudo haber influido en los resultados obtenidos.

Capítulo 6

Caso de estudio

Este capítulo presenta un caso de estudio aplicado al ámbito de la salud, con el objetivo de validar la funcionalidad de la aplicación desarrollada, evaluando su rendimiento y capacidad de manejo de datos. El caso seleccionado permite mostrar la aplicabilidad práctica de la solución utilizando datos basados en un escenario real, destacando su potencial para mejorar la gestión de servicios de emergencias médicas.

6.1. Contexto del caso de estudio

En este caso de estudio, los sujetos representan a personas que buscan atención en un centro hospitalario, específicamente en el servicio de emergencia. En este contexto, se recopilan datos al momento de la atención, incluyendo la dirección de residencia, datos personales y detalles de la situación que motiva la consulta.

Durante el proceso de atención, se realiza un *triage*, que es un proceso de clasificación que permite evaluar la severidad de la condición de cada persona y asignarla a una categoría de prioridad según la urgencia de su situación.

En un escenario real, se utilizaría la dirección específica de cada persona para el registro de los pacientes. Sin embargo, en este caso de estudio, con el fin de preservar la privacidad de las personas, se emplea en su lugar la “unidad asistencial” de referencia, es decir, la policlínica u hospital correspondiente a su área geográfica. Estas unidades asistenciales, gestionadas por la *Administración de los Servicios de Salud del Estado*¹ (ASSE), son instituciones reales y están disponibles como datos abiertos, accesibles al público.

¹<https://gis.asse.uy/gisasse/sigi/>

6.2. Datos del negocio

Para este caso de estudio, se dispone de un archivo CSV con los encabezados utilizados en un servicio de emergencias, de modo que el archivo refleja la estructura de datos empleada en la práctica. Aunque los encabezados del archivo se conservan tal como en un entorno real, los datos de cada fila, que representan a los sujetos que acuden a la emergencia, son completamente ficticios. En el sistema, estos encabezados se cargan como atributos, los cuales son los siguientes:

- **ID_Persona:** Corresponde a un identificador único de la persona, como la cédula de identidad o un identificador de paciente dentro del sistema de salud.
- **Fecha:** Indica la fecha en que se dio el evento, es decir, la visita a la emergencia.
- **Edad:** Refleja la edad del paciente en el momento de la visita.
- **Sexo:** Indica el sexo biológico del paciente.
- **Clasificación:** Sistema de clasificación por color que se utiliza en el *triage* de emergencias. Este sistema incluye las siguientes categorías:
 - Azul: Consultas que no constituyen emergencias, como solicitud de hora para policlínica o repetición de medicación.
 - Verde: Emergencias sin peligro inmediato, como dolores leves.
 - Amarillo: Emergencias que requieren atención pero no son de vida o muerte, como trabajo de parto.
 - Naranja: Emergencias con necesidad de atención rápida, donde la demora podría implicar riesgo de vida.
 - Rojo: Urgencias que requieren atención inmediata, como un paro cardíaco o un accidente cerebrovascular.
- **Sintomas_guia:** Breve descripción de los síntomas presentados por el paciente.
- **Departamento:** Departamento de la unidad asistencial.
- **Unidad_asistencial:** Indica la policlínica u hospital público de referencia del paciente.

Además, se dispone de un archivo CSV que contiene un listado de unidades asistenciales, incluyendo sus coordenadas de latitud y longitud.

6.3. Ejecución en la aplicación

A continuación, se describe el flujo de trabajo para ejecutar el caso de estudio. Un usuario analista, al iniciar sesión, debe seleccionar primero un espacio de trabajo. Para este caso específico, se creó un espacio de trabajo denominado “Pacientes de emergencia”. El siguiente paso consiste en crear los atributos necesarios, tal como se listaron en la sección anterior. Es importante señalar que, al crear los atributos de dirección, se optó por la opción de puntos de interés dentro de las formas normales predefinidas (ver Figura 6.1).

| ATRIBUTO | TIPO | IDENTIFICADOR | DIRECCION | ACCIONES |
|---|----------------------|----------------------------------|----------------------------------|----------------------|
| <input type="checkbox"/> ID_persona | Cadena de caracteres | <input checked="" type="radio"/> | | <input type="text"/> |
| <input type="checkbox"/> Fecha | Fecha | | | <input type="text"/> |
| <input type="checkbox"/> Edad | Número | | | <input type="text"/> |
| <input type="checkbox"/> Sexo | Cadena de caracteres | | | <input type="text"/> |
| <input type="checkbox"/> Clasificación | Cadena de caracteres | | | <input type="text"/> |
| <input type="checkbox"/> Síntomas_guia | Cadena de caracteres | | | <input type="text"/> |
| <input type="checkbox"/> Departamento | Cadena de caracteres | | | <input type="text"/> |
| <input type="checkbox"/> Localidad | Cadena de caracteres | | | <input type="text"/> |
| <input type="checkbox"/> Unidad_asistencial | Cadena de caracteres | | <input checked="" type="radio"/> | <input type="text"/> |

Figura 6.1: Atributos creados en el sistema.

Luego, el usuario debe cargar un archivo CSV con los puntos de interés (ver Figura 6.2) y otro archivo CSV con los datos de los pacientes. El CSV inicial contenía datos de 200 sujetos, y se añadieron más archivos hasta alcanzar los 10.000 sujetos. Como se ilustra en la Figura 6.3, todos los datos fueron correctamente geocalizados. En este caso, no fue necesario utilizar el servicio de geocodificación, ya que las ubicaciones se basaron en los POIs previamente cargados.

| NOMBRE | LATITUD | LONGITUD | SRID | UBICACIÓN |
|---|---------|----------|------|-----------|
| CENTRO DE SALUD PIEDRAS BLANCAS (ANEXO) | -34.819 | -56.14 | 4326 | |
| CENTRO DE SALUD CIUDAD DEL PLATA | -34.767 | -56.389 | 4326 | |
| CENTRO DE SALUD SAYAGO | -34.832 | -56.219 | 4326 | |
| POLICLINICA SOCA | -34.684 | -55.704 | 4326 | |
| POLICLINICA TARARRIAS | -34.266 | -57.617 | 4326 | |
| POLICLINICA TOMAS GOMENSORO | -30.429 | -57.44 | 4326 | |
| POLICLINICA TRANQUERAS | -31.194 | -55.762 | 4326 | |
| CENTRO DE SALUD UNION | -34.857 | -56.132 | 4326 | |
| POLICLINICA VICHADERO | -31.777 | -54.693 | 4326 | |
| HOSPITAL DE LA MUJER | -34.898 | -56.164 | 4326 | |

Figura 6.2: Unidades asistenciales cargadas en el sistema.

| ID_PERSONA | FECHA | EDAD | SEXO | CLASIFICACION | SINTOMAS_DUSA | DEPARTAMENTO | LOCALIDAD | UNIDAD_ASISTENCIAL | UBICACIÓN |
|------------|------------|------|------|---------------|--|--------------|-------------------|-----------------------------------|-----------|
| 10003 | 2024-03-30 | 60 | M | Azul | Consulta administrativa | SORIANO | MERCEDES | CENTRO DEPARTAMENTAL DE SORIANO | |
| 10005 | 2023-02-22 | 34 | F | Amarillo | Herida profunda sin sangrado masivo | ARTIGAS | ARTIGAS | CENTRO DEPARTAMENTAL ARTIGAS | |
| 10001 | 2024-05-07 | 8 | F | Naranja | Fractura expuesta | CANELONES | EL PINAR | POLICLINICA PINAR | |
| 10007 | 2023-05-11 | 45 | F | Rojo | Paro cardiorrespiratorio | MONTEVIDEO | MONTEVIDEO | CENTRO DE SALUD GIORDANO | |
| 10007 | 2024-04-21 | 46 | F | Verde | Tos persistente | MONTEVIDEO | MONTEVIDEO | CENTRO DE SALUD GIORDANO | |
| 10011 | 2023-07-24 | 7 | M | Rojo | Dolor torácico intenso (posible infarto) | LAVALLEJA | SOLIS DE MATAJOJO | CENTRO DE SALUD SOLIS DE MATAJOJO | |
| 10009 | 2024-01-19 | 77 | M | Verde | Heridas superficiales | CANELONES | CANELONES | CENTRO DEPARTAMENTAL CANELONES | |
| 10012 | 2024-11-14 | 26 | F | Naranja | Dolor abdominal (posible apendicitis) | ROCHA | LA PALOMA | POLICLINICA LA PALOMA | |
| 10015 | 2024- | 73 | M | Naranja | Dificultad respiratoria | CANELONES | LAS PIEDRAS | CENTRO AUXILIAR LAS | |

Figura 6.3: Datos cargados en el sistema.

Para llevar a cabo el análisis geoespacial, el siguiente paso es cargar las capas geográficas en la sección de *Análisis de Datos*. Las capas cargadas incluyen áreas administrativas que representan los departamentos de Uruguay, así como capas de puntos que abarcan todas las unidades asistenciales y aquellas que brindan servicios de maternidad (ver Figura 6.4).

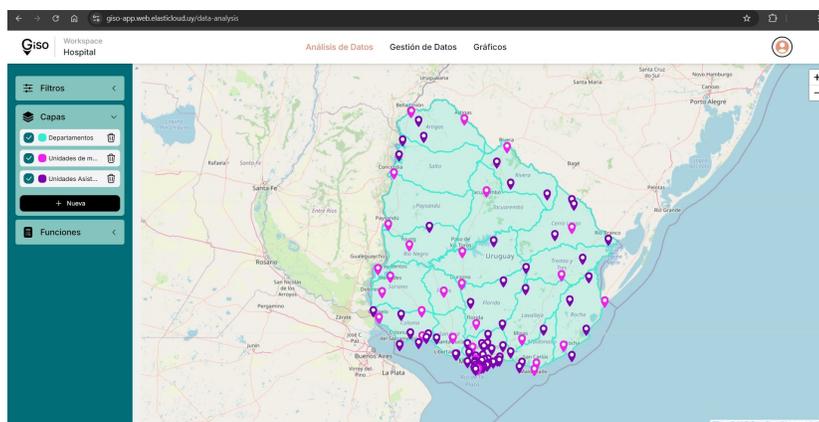


Figura 6.4: Capas geográficas cargadas en el sistema.

En este escenario de salud, se plantearon varias consultas de análisis geoespaciales con el objetivo de evaluar el acceso a los servicios de salud. A continuación, se detallan los pasos ejecutados para cada uno de estos análisis:

1. **Contar la cantidad de pacientes en cada área administrativa:** Se generó una capa utilizando la función espacial de conteo, seleccionando todos los sujetos y la capa de departamentos para calcular el número de pacientes en cada área administrativa (ver Figura 6.5).
2. **Calcular la distancia promedio desde el hogar de un paciente hasta el servicio de salud más cercano:** Se generó una capa utilizando la función espacial de distancia promedio, seleccionando todos los sujetos y la capa de unidades asistenciales para determinar la distancia media desde cada hogar al servicio de salud más próximo.
3. **Contar el número de pacientes dentro del radio de una clínica:** Se creó un filtro geográfico configurando una unidad asistencial, como el *Centro Auxiliar Young*, como punto central con un radio de 5 kilómetros. Esto permitió contabilizar el número de pacientes ubicados dentro de esa área (ver Figura 6.6).
4. **Contar el número de pacientes femeninos dentro del radio de un servicio de maternidad:** Se creó un filtro geográfico utilizando una unidad asistencial de maternidad, como el *Hospital de la Mujer*, como punto central con un radio de 5 kilómetros. Además, se aplicó un filtro

por atributo basado en los síntomas guía para identificar específicamente los casos de embarazo (ver Figura 6.7).

5. **Calcular la distancia promedio desde el hogar de pacientes embarazadas hasta el servicio de maternidad más cercano:** Primero, se creó un filtro por atributo basado en los síntomas guía para identificar los casos de embarazo. Luego, se generó una capa utilizando la función espacial de distancia promedio, seleccionando el filtro de pacientes embarazadas antes creado y la capa de unidades asistenciales de maternidad (ver Figura 6.8).

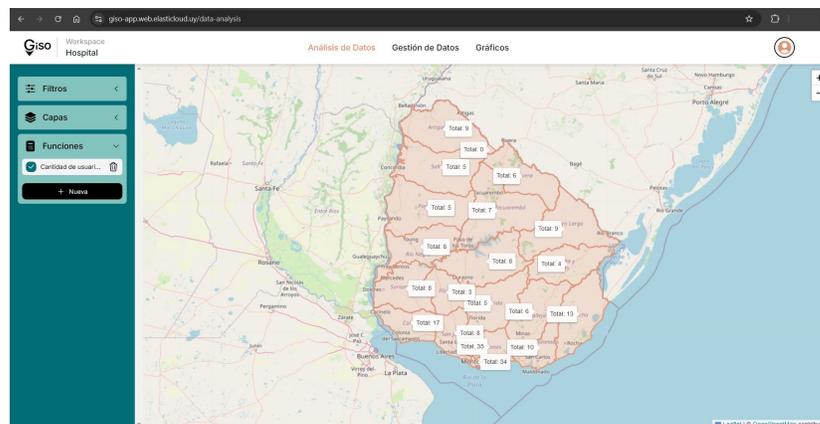


Figura 6.5: Cálculo de la cantidad de pacientes dentro de cada área administrativa (Consulta 1).

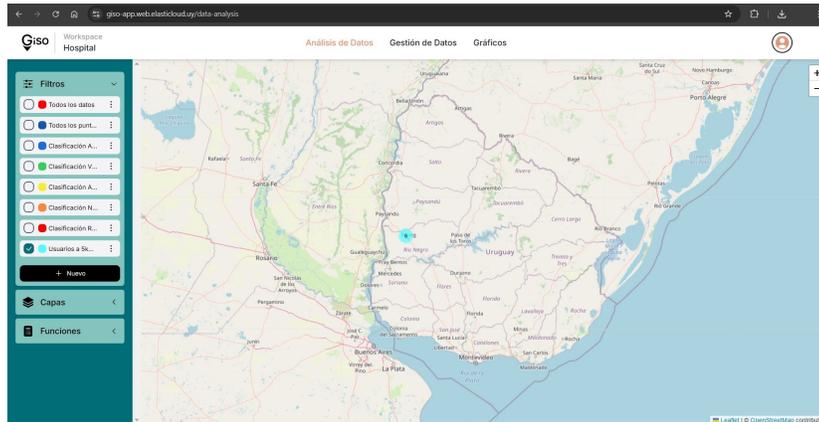


Figura 6.6: Cálculo del número de pacientes dentro del radio de una clínica (Consulta 3).

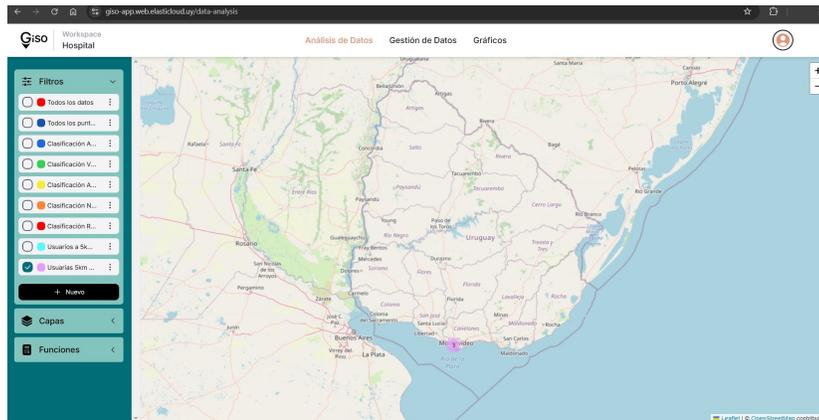


Figura 6.7: Cálculo del número de pacientes femeninos dentro del radio de un servicio de maternidad (Consulta 4).

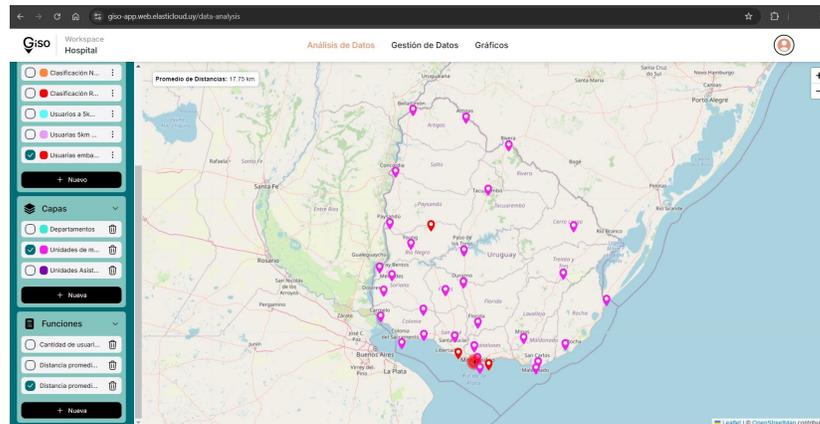


Figura 6.8: Cálculo de la distancia promedio desde el hogar de pacientes embarazadas hasta el servicio de maternidad más cercano (Consulta 5).

En la Figura 6.9 se muestra el mapa con los 10.000 sujetos cargados, los puntos de interés y la capa de departamentos. Los círculos numerados representan agrupaciones de puntos; el número en cada círculo indica la cantidad de puntos agrupados en esa área.

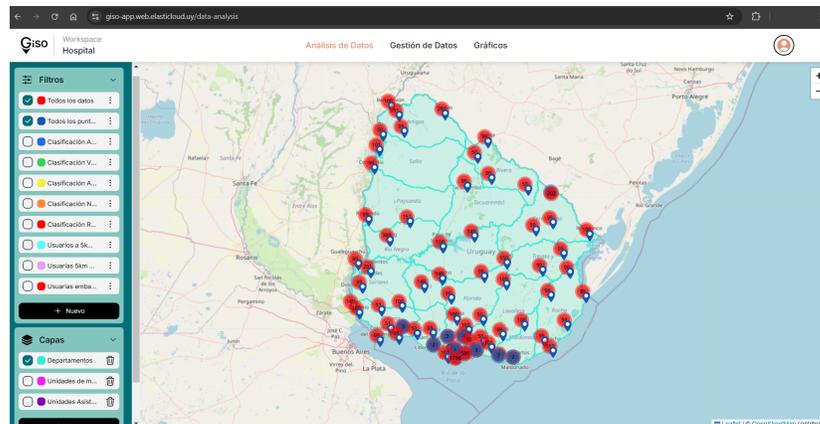


Figura 6.9: Mapa con los 10.000 sujetos cargados en el sistema.

Finalmente, el usuario puede visualizar estadísticas y gráficos personalizables en la sección de *Gráficos*. Estos gráficos incluyen el número de sujetos por tipo de clasificación, un gráfico de torta que muestra la proporción de hombres versus mujeres, y la cantidad de sujetos por departamento (ver Figura 6.10).

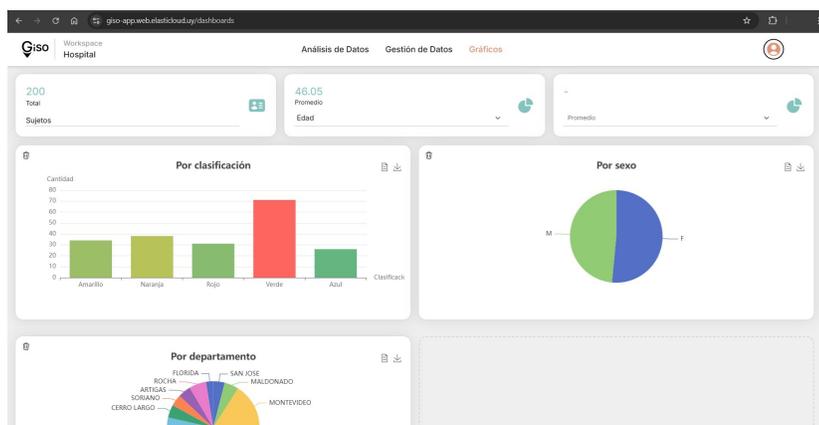


Figura 6.10: Gráficas personalizables creadas en el sistema.

6.3.1. Geocodificación de los puntos

Además de los datos utilizados en el caso de estudio principal, se cargó un conjunto de datos diferente correspondiente a accidentes de tránsito. En este caso, se utilizó un conjunto de datos con direcciones en lugar de puntos de interés, dado que los datos de servicios de emergencias proporcionaban ubicaciones precisas. Este enfoque permitió observar el proceso de geolocalización mediante el uso de geocodificadores.

En la Figura 6.11 se muestran los datos de los puntos geocodificados: los puntos correctamente localizados se indican en color verde, mientras que aquellos ubicados manualmente se presentan en verde con una “M”. Los puntos con baja confianza en la geocodificación están marcados en rojo, y aquellos con confianza media se muestran en amarillo.

| FECHAYHORA | TIPODESNUESTRO | EDAD | FALLECIDOALOSDIAS | VEHICULO | SEXO | DEPARTAMENTO | LUGARDELSNUESTRO | X | UBICACIÓN |
|---------------------|--------------------------|------|-------------------|-----------|------|----------------|---|-----------------|-----------|
| 2019-04-08 13:40:00 | COLISION ENTRE VEHICULOS | 68 | 0 | AUTO | M | FLORIDA | Ruta 5 KM 94.800 | 574270.27845281 | 🟢 |
| 2019-04-08 07:17:00 | COLISION ENTRE VEHICULOS | 20 | 0 | MOTO | M | CANELONES | AVENIDA EL DORADO ESQ. VIA FERREA | 571762.3953 | 🔴 |
| 2019-04-08 05:50:00 | DESPISTE | 21 | 4 | MOTO | M | CANELONES | RUTA 5 KM 45.700 Y 11 | 566916.69916212 | 🟡 |
| 2019-04-10 15:40:00 | COLISION ENTRE VEHICULOS | 15 | 1 | BICICLETA | F | TACUAREMBO | GENERAL FLORES ESQ. CONTINUACION LUIS CASTAGNETTO | 595949.2827 | 🟢 |
| 2019-04-11 03:40:00 | COLISION ENTRE VEHICULOS | 67 | 0 | AUTO | M | FLORIDA | RUTA 5 KM 100.000 | 570532.8 | 🟢 |
| 2019-04-12 22:30:00 | ATROPELLO DE PEATON | 49 | 0 | PEATON | M | TACUAREMBO | RUTA 5 KM. 251 | 546260.51 | 🟢 |
| 2019-04-13 00:20:00 | COLISION ENTRE VEHICULOS | 55 | 0 | AUTO | M | PAYSANDU | RUTA 3 KM 367.400 | 400853.16 | 🟢 |
| 2019-04-14 06:32:00 | DESPISTE | 37 | 1 | CAMIONETA | M | TREINTA Y TRES | RUTA 98 KM 4.000 | 741431.49228359 | 🔴 |

Figura 6.11: Datos cargados en el sistema utilizando un geocodificador.

6.4. Análisis de resultados

La evaluación general de la plataforma muestra que la aplicación es capaz de ejecutar el flujo completo utilizando datos realistas, confirmando así su funcionalidad integral. Además, ofrece herramientas de visualización que permiten interpretar los datos y extraer conclusiones rápidamente, optimizando así el valor de los archivos CSV al proporcionar una visualización interactiva en mapas, una capacidad que supera a las herramientas convencionales de manejo de CSV.

Gracias al mapa interactivo con funciones de filtrado, los analistas pueden explorar los datos geospaciales de manera dinámica. Esta funcionalidad se complementa con gráficos adicionales, que refuerzan el análisis estadístico y lo vuelven visualmente comprensible.

Los resultados obtenidos reflejan que la plataforma no solo cumple con los objetivos establecidos, sino que también proporciona una herramienta para la gestión de datos y la evaluación del acceso a los servicios de salud.

Capítulo 7

Gestión del proyecto

En este capítulo se aborda la gestión del proyecto, detallando la planificación realizada, las desviaciones que surgieron durante su desarrollo, y las herramientas utilizadas para llevarlo a cabo.

7.1. Planificación

Al inicio del proyecto, se realizó un cronograma tentativo de las fases principales del desarrollo, según lo establecido en las reuniones con los tutores (ver Figura 7.1). Este cronograma establecía cuatro fases:

1. Estado del arte: En esta fase inicial, se plantea un estudio de los conceptos teóricos fundamentales, análisis de herramientas existentes, revisión de trabajos relacionados, y pruebas de concepto con geocodificadores.
2. Definición de requerimientos: Esta fase implica la definición de los requerimientos del sistema, el diseño de la aplicación y su arquitectura. Además, se realiza la configuración del ambiente de desarrollo necesario para la implementación.
3. Implementación del sistema: Durante esta fase, se desarrolla la aplicación siguiendo los lineamientos establecidos en las etapas previas.
4. Pruebas del sistema y documentación final: Finalmente, en la última fase se realizan pruebas del sistema para procurar un mejor funcionamiento, acompañadas de la elaboración de la documentación final del proyecto.

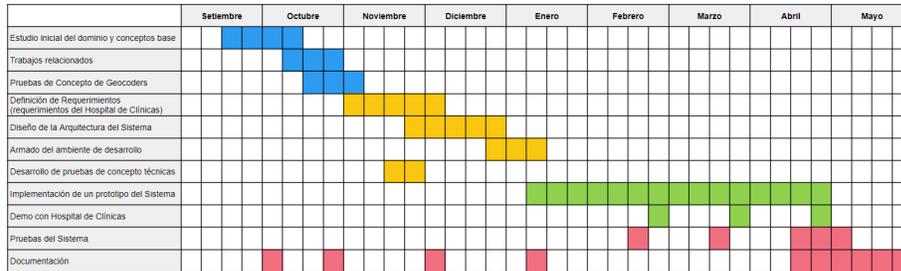


Figura 7.1: Diagrama Gantt inicial.

7.2. Desviaciones

Durante el desarrollo del proyecto, se presentaron algunas desviaciones respecto al cronograma original, lo que resultó en una extensión de varios meses en el plazo previsto (ver Figura 7.2). Estas desviaciones se debieron principalmente a imprevistos surgidos durante el proceso y a la necesidad de profundizar en ciertos temas para asegurar una comprensión adecuada y garantizar la calidad del resultado final.

El primer desvío ocurrió durante la primera etapa del proyecto, extendiéndose un mes más de lo planificado. Esto se debió a que el proceso de familiarización con los GIS y los aspectos técnicos asociados tomó más tiempo de lo previsto, dado que era la primera vez que los miembros del equipo se enfrentaban a esta temática.

La segunda etapa, correspondiente a la definición de requerimientos y el diseño del sistema, también sufrió una extensión considerable. Factores relacionados con la organización y dificultades en la coordinación de tiempos entre los tutores y los integrantes del equipo durante diciembre, sumados a las vacaciones de enero, llevaron a que esta fase se prolongara hasta abril.

Además, la etapa de implementación se extendió un mes más debido a una pausa en agosto. Este retraso se debió a factores ajenos al proyecto, ya que uno de los integrantes del equipo debió asumir responsabilidades laborales adicionales, lo que redujo su disponibilidad para trabajar en el proyecto durante ese tiempo.

Por otro lado, aunque no representó una desviación, en agosto surgió una oportunidad que implicó trabajo adicional. Los tutores invitaron al equipo a participar en *Ingeniería de Muestra*, un evento anual en octubre donde se presentan los proyectos de grado e investigación de la *Facultad de Ingeniería de la UdelaR*. Esto sumó la tarea de diseñar un póster para el stand, trabajo que se realizó durante septiembre.



Figura 7.2: Diagrama Gantt final.

7.3. Herramientas

Para llevar a cabo este proyecto, se utilizaron diversas herramientas que facilitaron la planificación, desarrollo y documentación. A continuación, se detallan las principales herramientas empleadas y su papel en el proceso.

- *Trello*¹: herramienta colaborativa utilizada para listar y distribuir las tareas del proyecto, facilitando la asignación y el seguimiento del trabajo entre los integrantes.
- *Mattermost*²: plataforma de comunicación utilizada para la coordinación y el intercambio con los tutores.
- *Owncloud*³: servicio de almacenamiento en la nube, en el cual se compartió documentación con los tutores.
- *GitHub*⁴: plataforma de control de versiones que permitió gestionar y mantener el código fuente del proyecto, facilitando la colaboración entre los miembros del equipo.
- *Overleaf*⁵: editor de texto en línea basado en LaTeX, utilizado para la redacción simultánea de la documentación del proyecto por parte de los miembros del equipo.
- *Zoom*⁶: plataforma de videoconferencias utilizada para reuniones virtuales durante el proyecto.

¹<https://trello.com/>

²<https://mattermost.com/>

³<https://owncloud.com/>

⁴<https://github.com/>

⁵<https://es.overleaf.com/>

⁶<https://zoom.us/>

Capítulo 8

Conclusiones y trabajo futuro

En este capítulo, se resumen los resultados principales del proyecto, evaluando los logros alcanzados en relación con los objetivos iniciales y las dificultades encontradas durante el desarrollo. Además, se proponen mejoras futuras para el proyecto.

8.1. Conclusiones

Se completó con éxito el diseño e implementación del sistema propuesto, cumpliendo con todos los requerimientos establecidos. La principal contribución de este proyecto fue desarrollar una plataforma que permite a usuarios, independientemente de su experiencia técnica, realizar análisis espaciales complejos, adaptable a diversos dominios.

Inicialmente, el proyecto requirió un profundo estudio en datos espaciales y GIS, un desafío considerable dado que el equipo no contaba con experiencia previa en el área. Sin embargo, gracias al apoyo de los tutores especializados en la materia y a un estudio constante, se logró adquirir un conocimiento sólido en esta temática, lo que permitió alcanzar los objetivos propuestos.

La etapa más crítica del desarrollo fue el diseño del modelo conceptual, que debía ser adaptable a diferentes dominios, y la arquitectura necesaria para la integración con *GeoServer*. En este proceso, el intercambio en las reuniones de validación resultó fundamental: plantear los problemas encontrados y colaborar en la búsqueda de soluciones permitió cumplir con los requerimientos necesarios de manera efectiva.

Para el desarrollo del proyecto, la gestión adecuada fue crucial para planificar e implementar las funcionalidades de manera ordenada, lo que permitió desplegar el prototipo en la nube de forma incremental. También, el conocimiento previo en las tecnologías principales del frontend, backend y despliegue en la

nube facilitó la implementación, permitiendo al equipo centrarse en aprender y aplicar tecnologías nuevas como *GeoServer*, *PostGIS*, *Leaflet* y *Turf.js*.

Durante las pruebas del sistema, especialmente al utilizar el servicio de geocodificación de la IDEuy, se identificaron problemas significativos de rendimiento del sistema externo. A pesar de los esfuerzos por optimizar el procesamiento de datos, el desempeño del prototipo seguía estando condicionado por la velocidad de respuesta del servicio de la IDEuy. Dado que uno de los objetivos del proyecto era garantizar la geocodificación precisa de la mayor cantidad posible de direcciones, mejorar los tiempos de procesamiento a costa de reducir la cantidad de direcciones correctamente geocodificadas no era una opción viable.

La factibilidad técnica lograda con datos realistas, como emergencias hospitalarias, y datos reales, como accidentes de tránsito en Uruguay, es muy importante. No solo comprueba la capacidad del sistema para adaptarse a distintos tipos de información, sino que también resalta su eficacia en la gestión de escenarios del mundo real.

Por otra parte, durante la instancia de *Ingeniería de Muestra*, muchas de las personas que visitaron el stand compartieron observaciones sobre el potencial del sistema en diversos contextos. Entre las aplicaciones sugeridas, destacaron su posible uso en epidemiología, en localización de balizas en el mapa y en la gestión de datos para clientes empresariales, lo que refleja el interés y la versatilidad percibida de la plataforma. Además, se recibieron comentarios positivos sobre la interfaz, resaltando su claridad en la presentación de la información, lo que facilita la interpretación de los datos y potencia su utilidad en distintos escenarios.

Se concluye que se ha desarrollado una herramienta versátil, que puede ser empleada por diversos organismos para realizar geocodificación asistida y análisis geoespaciales sin necesidad de conocimientos técnicos avanzados en GIS o programación.

8.2. Trabajo futuro

Para optimizar la funcionalidad del sistema, se han identificado varias áreas de mejora. En primer lugar, se podría implementar la ofuscación de las coordenadas obtenidas mediante geocodificación, transformándolas en zonas basadas en una cuadrícula, como manzanas, secciones o segmentos censales, lo que mejoraría la protección de los datos sensibles. Además, se podría encriptar la información confidencial para salvaguardar su privacidad en caso de brechas de seguridad.

Por otro lado, se podría ampliar el número de geocodificadores integrados. La arquitectura actual del sistema permite esta expansión; solo se requiere desarrollar una lógica adicional para mapear los formatos estándar a las distintas API de geocodificación y gestionar el nivel de confianza de las respuestas.

También se podría implementar un sistema de notificaciones que informe sobre el progreso de la geolocalización, proporcionando datos como el porcentaje de información procesada y avisando sobre la incorporación de nuevos datos.

Adicionalmente, se podrían integrar nuevos predicados espaciales, como el cálculo de distancias entre puntos y líneas y el análisis mediante isócronas. Esta última funcionalidad permitiría definir áreas de cobertura basadas en tiempo o distancia a lo largo de calles y rutas. En el contexto de salud, por ejemplo, considerando un punto de atención central, como una unidad asistencial, se podrían identificar los puntos que estarían dentro de una isócrona que represente el tiempo estimado de llegada de una ambulancia.

En cuanto a la gestión de datos, se podría mejorar unificando la información de sujetos según su atributo identificador, lo que permitiría consolidar datos cargados en múltiples instancias y facilitar así su agrupación.

Otra mejora posible es expandir las opciones en la sección de Gráficas, incorporando tipos adicionales como gráficos de líneas. Esto permitiría a los usuarios visualizar tendencias y patrones en los datos de manera más clara.

En relación con el rendimiento y la infraestructura, se ha observado que a medida que aumenta el volumen de datos, los tiempos de procesamiento de los filtros se incrementan, lo que impacta en la velocidad que se despliegan los datos en el mapa. Para mitigar este problema, se podrían explorar opciones como la asignación de más memoria al servidor o la optimización de la configuración de *GeoServer*.

Por otra parte, sería conveniente implementar un proceso que elimine físicamente los archivos SHP del sistema de archivos del servidor de *GeoServer*. En la configuración actual, al eliminar un almacén de datos, los archivos se eliminan solo de manera lógica, pero permanecen físicamente en el servidor, ocupando espacio innecesario.

Finalmente, en el panel de administración, se podría integrar la API de *Elastic Cloud* para visualizar información crucial sobre el uso de disco, memoria RAM, CPU y otros aspectos relevantes para la gestión de la infraestructura.

Referencias

- Bernabé Poveda, M. A., y López Vázquez, C. M. (2012). *Fundamentos de las infraestructuras de datos espaciales (ide)*. Madrid, España: UPM Press.
- Bui, T. Q., y Pham, H. M. (2016). Web-based gis for spatial pattern detection: application to malaria incidence in vietnam. *Springerplus*.
- Chaer, Nadia. (2015). *Sistemas de referencia de coordenadas*. <https://siglibreuruguay.wordpress.com/2015/12/23/sistemas-de-referencia-de-coordenadas/>. (Accessed: 2024-08)
- Delmelle, E., Cahill Delmelle, E., Casas, I., y Barto, T. (2011). H.e.l.p: A gis-based health exploratory analysis tool for practitioners. *Applied Spatial Analysis and Policy*, 4, 113–137.
- Delmelle, E. M., Zhu, H., Tang, W., y Casas, I. (2014). A web-based geospatial toolkit for the monitoring of dengue fever. *Applied Geography*, 52, 144–152.
- Dr J de Smith, M., Prof F Goodchild, M., Prof A Longley, P., y Associates. (2021). *Geospatial analysis*. UK: www.spatialanalysisonline.com.
- ESRI. (2024). *What is gis?* <https://www.esri.com/en-us/what-is-gis/overview>. (Accessed: 2024-07)
- Goldberg, D. W., Wilson, J. P., y Knoblock, C. A. (2007). From text to geographic coordinates: The current state of geocoding. *Urisa Journal*, 19, 33.
- González, L., Foti, P., Sosa, R., y Rienzi, B. (2010). Benefits and challenges of using geographic information systems to enhance social security services. En *6th international policy and research conference on social security*.
- Herring, John R. (2011). *Implementation standard for geographic information - simple feature access - part 1: Common architecture*. Open Geospatial Consortium Inc. (Accessed: 2024-07)
- Hohpe, G., y Woolf, B. (2003). *Enterprise integration patterns : Designing, building, and deploying messaging solutions*. Addison-Wesley Professional.
- Infraestructura de Datos Espaciales. (2022). *Sistema Único de direcciones*. <https://www.gub.uy/infraestructura-datos-espaciales/sudir>. (Accessed: 2024-11)
- Infraestructura de Datos Espaciales. (2024a). *Modelo de direcciones geográficas del uruguay*. <https://www.gub.uy/infraestructura-datos-espaciales/comunicacion/publicaciones/modelo-direcciones-geograficas-del-uruguay>. (Accessed: 2024-11)

- Infraestructura de Datos Espaciales. (2024b). *Nodos ideuy*. <https://www.gub.uy/infraestructura-datos-espaciales/nodos-ideuy>. (Accessed: 2024-08)
- Instituto Geográfico Militar. (2022). *Situación geográfica*. <https://igm.gub.uy/situacion-geografica/>. (Accessed: 2024-08)
- Longley, P. A., Maguire, D. J., Rhind, D. W., Goodchild, M., y Goodchild, M. F. (2010). *Geographic information systems science*. United States of America: John Wiley Sons Inc.
- Mell, P., y Grance, T. (2011). The nist definition of cloud computing. *NIST SP 800-145*, 7.
- QGis. (2024). *Sistema de referencia de coordenadas*. https://docs.qgis.org/3.34/es/docs/gentle_gis_introduction/coordinate_reference_systems.html. (Accessed: 2024-07)
- Registro Nacional de Leyes y Decretos. (2013). *Artículo 75 de la ley n° 18362*.
- Shekhar, S., Xiong, H., y Zhou, X. (2017). *Encyclopedia of gis*. Cham, Switzerland: Springer Cham.
- Sosa, R., Rienzi, B., González, L., Cortés, J., Sediey, V., y Segura, V. (2024, November). *Towards a platform for assisted geocoding and geoanalysis: the case of health information*. (unpublished)