



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Ionita , Florin Alexandru

Tutor/a: Lemus Zúñiga, Lenin Guillermo

CURSO ACADÉMICO: 2022/2023



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica

Universitat Politècnica de València

Infraestructura basada en tecnologies de cartografia en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Ionita, Florin Alexandru

Tutor: Lenin Guillermo Lemus Zúñiga

Curso escolar 2022/2023

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

Resumen

Este trabajo tiene como objetivo principal el diseño e implementación de un sistema de información geográfica, con la finalidad de representar datos geoespaciales en un mapa interactivo.

El objetivo secundario es dotar al sistema de una granularidad de representación cartográfica a nivel de municipio, código postal y barrio.

Para cumplir estos objetivos, se ha elaborado un plan de trabajo que consiste de tres fases. Una fase inicial dedicada a la formación y adquisición de conocimientos relacionados con la cartografía y la tecnología *web-mapping*, para luego proceder a idear un diseño conceptual del sistema que se pretende crear. Se continúa con una segunda fase en la que se van a iniciar e instalar todos los servicios y componentes de *software* necesarios para crear el *SIG*. Se concluye esta fase con una validación de la infraestructura. Por último, se dedica la fase final para realizar pruebas de concepto que pongan a prueba las capacidades del *SIG* en el contexto de los objetivos propuestos, seguido por una conclusión y una reflexión con respecto a posibles mejoras que se pueden llevar a cabo.

Palabras clave: *SIG*, *web-mapping*, capa, mapa, geoespacial, geometría

Abstract

This work aims to design and implement a geographic information system with the purpose of representing geospatial data on an interactive map.

A secondary objective is to provide the system with a cartographic rendering granularity at the municipality, postal code, and neighbourhood levels.

To achieve these objectives, a work plan consisting of three phases has been developed. The initial phase is dedicated to training and acquiring knowledge related to cartography and web-mapping technology, followed by creating a conceptual design of the intended system. The second phase involves the initiation and installation of all the necessary services and software components to create the *GIS*, ending with an infrastructure validation. Finally, the last phase is dedicated to conducting proof-of-concept tests to assess the capabilities of the *GIS* in the context of the proposed objectives, followed by a conclusion and reflection on potential improvements that can be implemented.

Keywords: *GIS*, *web-mapping*, layer, map, geospatial, geometry

Índice General

1.	Introducción	11
1.1.	Objetivo.....	12
1.2.	Metodología	12
2.	Estado del Arte.....	15
2.1.	Geometría y Geografía.....	15
2.2.	Sistema de Información Geográfica (SIG)	15
2.3.	Modelos de Datos Geográficos.....	16
2.3.1.	Datos Vectoriales	16
2.3.2.	Datos Ráster	17
2.4.	Proyección Cartográfica	18
2.5.	Sistema de Referencia de Coordenadas	20
2.5.1.	El Identificador de Referencia Espacial	20
2.6.	Base de Datos Espacial	21
2.7.	Servidor Geoespacial.....	22
2.7.1.	Capas	22
2.7.2.	Servicios Geoespaciales	22
2.8.	Los Visores.....	23
2.9.	Aplicaciones Prácticas.....	24
3.	Análisis del Problema.....	27
3.1.	Análisis del Marco Legal y Ético	27
3.1.1.	Análisis de la Protección de Datos.....	27
3.2.	Identificación y Análisis de Soluciones Posibles	27
3.3.	Solución propuesta	28
3.4.	Plan de trabajo	28
3.5.	Presupuesto	30



Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

4.	Diseño de la Solución	33
4.1.	Especificación de Requisitos	33
4.2.	Arquitectura del Sistema.....	35
4.3.	Diseño Detallado	36
4.4.	Tecnología Utilizada.....	37
5.	Desarrollo de la Solución Propuesta.....	43
5.1.	Configuración de los servicios de <i>Google Cloud Platform</i>	43
5.1.1.	Nuevo Proyecto en <i>GCP</i>.....	43
5.1.2.	Nueva Instancia de Servicio <i>Cloud SQL</i>.....	43
5.1.3.	Nueva Instancia de Servicio <i>Compute Engine</i>	45
5.2.	Configuración de <i>PostgreSQL</i>	47
5.3.	Instalación del <i>GeoServer</i>	50
6.	Validación	53
6.1.	Riesgo de Incendio por Municipio en la Comunidad Valenciana	53
6.1.1.	Los Datos	53
6.1.1.1.	Municipios de la Comunidad Valenciana	53
6.1.1.2.	Riesgo de Incendios.....	55
6.1.1.3.	Riesgo de Incendios por Municipio y Día	55
6.1.2.	Las Capas	56
6.1.2.1.	Capa de los Municipios	58
6.1.2.2.	Capa de Puntos de Riesgos.....	59
6.1.3.	El Mapa.....	61
6.1.3.1.	Grupo de Capas	61
6.1.3.2.	Estilos.....	63
6.1.3.2.1.	Capa con Transparencia y Malla de Colores.....	63
6.1.3.2.2.	Capa con Transparencia y Mapa de Calor	67
6.1.3.3.	Habilitar Servicios para Clientes	68
6.1.4.	Comentarios	70
6.2.	Tráfico en la localidad de Valencia	71



6.2.1.	Los Datos	71
6.2.2.	Carga y Limpieza de datos.....	71
6.2.3.	Las Capas	73
6.2.3.1.	Capa con Tramos e Información de Transito	74
6.2.3.2.	Capa de <i>OpenStreetMap</i> proporcionado por <i>Terrestris</i>	79
6.2.4.	El Mapa.....	83
6.2.5.	Visualizador <i>Leaflet</i>	85
6.2.6.	Visualizador <i>Leaflet</i> con funcionalidad extendida.....	88
6.3.	Comentarios sobre la Validación	91
7.	Conclusiones	93
7.1.	Relación del Trabajo Desarrollado con los Estudios Cursados y las Competencias Transversales	93
8.	Trabajos Futuros.....	95
9.	Anexo	97
9.1.	Relación del Trabajo con los Objetivos de Desarrollo Sostenible de la Agenda 2030 de Naciones Unidas	97
10.	Glosario	99
11.	Referencias Bibliográficas.....	101
12.	Bibliografía	103

Índice de Figuras

Figura 1. Ejemplo de pilas de software SIG. Comercial a la izquierda, de código abierto a la derecha. Fuente: adaptada de [2].....	16
Figura 2. Ejemplo de geometría de tipo línea Fuente: adaptada de [3]	17
Figura 3. Ejemplo de geometría de tipo polígono. Fuente: adaptada de [3]	17
Figura 4. Estructura Ráster. Fuente: [4]	18
Figura 5. Proyección de Winkel-Tripel. Fuente: [5]	19
Figura 6. Proyección de Mercator. Fuente: [6]	20
Figura 7. Ejemplos de datos de entrada y salida de un servidor geoespacial.	22
Figura 8. Esquema del Desglose del Trabajo.	29
Figura 9. Gestión de procesos del producto SIG.	34
Figura 10. Componentes de la arquitectura del sistema.	35
Figura 12. Esquema de sistemas e interacción “Cliente-Servidor”.	37
Figura 13. Información del proyecto “tfg-gis”	43
Figura 14. Información de la instancia “postgis-server”	44
Figura 15. Configuración de conexiones de la instancia “postgis-server”.	45
Figura 16. Estimación de costes de la instancia “gis-geo-server”	45
Figura 17. Configuración de red de la instancia “gis-geo-server”	46
Figura 18. Información sobre las interfaces de red de la instancia “gis-geo-server”	47
Figura 19. Conexión remota a la máquina virtual “gis-geo-server”.	47
Figura 20. Código SQL para la creación de la base de datos “tfg”.	48
Figura 21. Código SQL para crear la extensión “postgis”.	48
Figura 22. Código SQL para agregar la extensión “postgis_raster”.	49
Figura 23. Código SQL para consultar las extensiones instaladas en la base de datos.	49
Figura 24. Código SQL para consultar la correcta creación de las vistas asociadas a las extensiones “postgis” y “postgis_raster”.	49
Figura 25. Creación de la variable de entorno GEOSERVER_HOME.	50
Figura 26. Binarios del GeoServer descomprimidos.	50
Figura 27. Asignación de nuevo dueño al directorio.	50
Figura 28. Arranque del GeoServer.	50
Figura 29. Pantalla de inicio de GeoServer.	51
Figura 30. Código SQL para la creación del esquema “com_valenciana”.	53
Figura 31. Estructura de la tabla “municipios_cv”.	54
Figura 32. Geometrias de la tabla “municipios_cv” representadas por el visor de mapas de DBEaver.	54

Figura 33. Representación visual de los puntos geográficos con la información sobre el riesgo de incendios de la tabla "datos_riesgo_cv".	55
Figura 34. Malla de riesgos de incendio, contenida en la vista materializada, formada por geometrías intersectadas de las tablas "municipios_cv" y "datos_riesgo_cv".	56
Figura 35. Ubicación de la sección "Datos".	57
Figura 36. Espacio de trabajo "WORKSPACE_tfg_CV".	57
Figura 37. Configuración del almacén de datos "DB_tfg_SCHEMA_com_valenciana".	58
Figura 38. Capa "municipios_cv" con atributos del municipio Morella.	59
Figura 39. Capa "cv_riesgo_maximo_ultimo_dia_geom_intersectada" con una malla de geometrías e información sobre riesgos de incendio máximos, del último día, de la Comunidad Valenciana.	60
Figura 40. Orden de dibujo de las capas del grupo "INCENDIOS_max_riesgo_dia_cv_geom_intersectadas".	61
Figura 41. Atributos devueltos por el grupo de capas "INCENDIOS_max_riesgo_dia_cv_geom_intersectadas".	62
Figura 42. Definición y código CSS del estilo "municipios_transparentes".	64
Figura 43. Definición y código CSS del estilo "colores_riesgo_cv".	65
Figura 44. Grupo de capas "INCENDIOS_max_riesgo_dia_cv_geom_intersectadas_colores" con el orden de dibujo y los estilos aplicados.	66
Figura 45. Mapa de la Comunidad Valenciana con el riesgo máximo de incendios representado por colores.	66
Figura 46. Código SLD del estilo "SLD_heatmap".	67
Figura 47. Grupo de capas "INCENDIOS_max_riesgo_dia_cv_geom_intersectadas_heatmap" con estilo "SLD_heatmap".	68
Figura 48. Habilitar servicios web de difusión de mapas del GeoServer para el espacio de trabajo "WORKSPACE_tfg_CV".	69
Figura 49. Proyecto en QGIS que usa la capa "cv_riesgo_maximo_ultimo_dia_geom_intersectada".	69
Figura 50. Estructura de la tabla "tramos_geom_info".	72
Figura 51. Estructura de la tabla "primitiva_info_trafico".	72
Figura 52. Código SQL para la creación de la vista materializada "capa_geom_tramos_con_info_trafico_ultimo_dia_hora_mv".	73
Figura 53. Espacio de trabajo "WORKSPACE_tfg_trafico_V" y sus características.	74
Figura 54. Configuración fuente de datos "DB_tfg_SCHEMA_trafico_valencia".	75
Figura 55. Información de la capa "tramos_con_info_trafico_ultimo_dia_hora".	76
Figura 56. Visualización de la capa "tramos_con_info_trafico_ultimo_dia_hora".	76
Figura 57. Código CSS para el estilo "estilos_trafico".	77

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

Figura 58. Capa "tramos_con_info_trafico_ultimo_dia_hora" con el estilo "estilos_trafico" aplicado.	78
Figura 59. Capa alternativa "tramos_con_info_trafico_dia_y_hora_especificos" con el estilo "estilos_trafico" aplicado.	78
Figura 60. Selección de origen de datos en GeoServer.....	79
Figura 61. Detalles de conexión a servicio WMS de Terrestris.....	80
Figura 62. Almacén de datos "TERRESTRIS_GRAY".	80
Figura 63. Capas proporcionadas por el servicio WMS de Terrestris.	81
Figura 64. Detalles de la capa "capa_OSM_terrestris_gris".....	82
Figura 65. Previsualización de la capa "capa_OSM_terrestris_gris".	82
Figura 66. Nombre y Espacio de trabajo del grupo de capas.	83
Figura 67. Límites, capas, estilos y orden de dibujo del grupo de capas.	84
Figura 68. Mapa dibujado por OpenLayers, centrado sobre Valencia y con información de tráfico de la última hora disponible.	84
Figura 69. Opciones del espacio de trabajo "WORKSPACE_tfg_trafico_V".	85
Figura 70. Localización de "tfg_trafico_valencia_ejemplo_encuadre.html"	85
Figura 71. Referencia hacia la librería Leaflet.....	86
Figura 72. Capas y mapa mediante objetos "map" y "tileLayer" de Leaflet.....	87
Figura 73. Web con el mapa del tráfico de Valencia usando Leaflet.	87
Figura 74. Mapa en Leaflet compuesto por dos capas de OpenStreetMap y Mapbox, y una capa con información de tráfico del SIG del proyecto.	88
Figura 75. Código JavaScript con Leaflet para añadir escala, control de capas y leyenda al mapa.	89
Figura 76. Código JavaScript con Leaflet para añadir una función de consulta de propiedades al mapa.	89
Figura 77. Mapa Leaflet con tráfico de Valencia, selector de escala, leyenda, función de consulta y capa base de MapBox seleccionada.....	90
Figura 78. Mapa Leaflet con tráfico de Valencia, selector de escala, leyenda, función de consulta y capa base de OpenStreetMap seleccionada.	90

Índice de Tablas

Tabla 1. Estimaciones de la duración de las actividades de la fase 1.	29
Tabla 2. Estimaciones de la duración de las actividades de la fase 1.	30
Tabla 3. Estimaciones de la duración de las actividades de la fase 2.	30



Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.



1. Introducción

La sociedad está embarcada hacia un futuro digital y en constante evolución. Organismos y empresas deben acelerar su transformación, a través de la digitalización y digitalización de sus servicios.

Cada vez hay una mayor cantidad de datos generados y la necesidad de representarlos adecuadamente. Una gran parte de estos datos contienen información geográfica relevante, que es mucho más intuitiva si se representa de forma visual en un mapa interactivo. Este es un aspecto común para el que la informática ha ido desarrollando soluciones desde los años 60.

Una de las tecnologías utilizadas para cubrir estas necesidades es el **sistema de información geográfica**, de ahora en adelante *SIG*. Los *SIG* son sistemas con una capacidad destacada para almacenar, editar, transmitir y visualizar información asociada a localizaciones geográficas. El resultado son mapas con una multitud de posibles características, pero siempre con un aspecto en común, ser el formato mejor adaptado para visualizar información de carácter geográfico.

Antes del auge de la web, los programas *SIG* estaban restringidos a gente con conocimientos y formación en cartografía. Eran programas locales de difícil acceso para usuarios fuera del ámbito estrictamente cartográfico. Con el auge de la web una nueva familia de aplicaciones *SIG* en la web ha sido creada. Es conocida como *web-mapping*.

Para ilustrar las bondades de los *SIG*, la empresa *ESRI*, pionera y líder en el mercado, los define de esta manera: *“Un sistema de información geográfica (SIG) es un sistema que crea, administra, analiza y representa cartográficamente todo tipo de datos. Un SIG ayuda a los usuarios a comprender los patrones, las relaciones y el contexto geográfico. Entre las ventajas se encuentran la mejora de la comunicación y la eficiencia, así como de la gestión y la toma de decisiones”* [1].

Partiendo del caso concreto de los grupos de investigación FAKTUMLAB del instituto ITACA de la UPV, que se dedican a la creación de aplicaciones en el ámbito de la salud, se desea dedicar este trabajo a la implementación de una infraestructura *SIG* que permita crear mapas, con los que dar soporte a las actividades de este tipo de usuarios. Estos mapas, que pueden ser interactivos, van a facilitar la detección de patrones o situaciones a nivel de barrio, código postal o municipio, que luego pueden ser usados en la toma de decisiones o la mejora de protocolos.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

1.1. Objetivo

El objetivo principal de este trabajo es el diseño, implementación y validación de una infraestructura de información geográfica (*SIG*) y la aplicación de técnicas de cartografía *WEB* para almacenar, visualizar e interactuar con mapas (datos geoespaciales).

Para lograr este objetivo se analizarán diferentes técnicas y tecnologías que permitan la visualización y la interactividad entre el usuario y los datos, para alcanzar un objetivo secundario correspondiente a visualizar información a nivel de municipio, código postal y barrio.

1.2. Metodología

La metodología que se ha seguido para cumplir con los objetivos propuestos se divide en los pasos que se describen a continuación:

- **Formación:** Adquirir conocimientos sobre los sistemas *SIG* y las técnicas de *web-mapping*, a través de un curso, con el fin de comprender el funcionamiento y los conceptos clave.
- **Búsqueda bibliográfica:** Realizar una investigación de sistemas similares disponibles en el mercado, con el objetivo de obtener información sobre características, costos, eficiencia, etc., que puedan ser relevantes para nuestra propuesta.
- **Establecimiento de requisitos:** Definir los requisitos específicos para nuestro sistema *SIG*, teniendo en cuenta aspectos como el costo, la eficiencia y otras necesidades específicas.
- **Propuesta de arquitectura:** Diseñar una arquitectura adecuada para nuestro sistema *SIG*, considerando los requisitos establecidos previamente.
- **Implementación:** Llevar a cabo la implementación de la arquitectura propuesta, utilizando herramientas y tecnologías adecuadas.
- **Validación de la infraestructura:** Utilizar la infraestructura diseñada para crear dos aplicaciones reales.
- **Análisis de resultados:** Evaluar los resultados obtenidos tras la implementación, teniendo en cuenta aspectos como los costos y la eficiencia del sistema.

Además, hay algunas tareas específicas relacionadas con la implementación:

- Búsqueda de una base de datos capaz de almacenar datos geoespaciales.
- Modelado de un esquema y modelo de datos que cumpla con los requisitos establecidos previamente.
- Elección del servidor *SIG* para la parte de la aplicación que se ejecuta del lado del servidor, de ahora en adelante **backend**.
- Selección del formato y servicio más adecuados para enviar los datos a la parte de la aplicación que se ejecuta del lado del usuario, de ahora en adelante **frontend**, teniendo en cuenta la interactividad y la velocidad necesarias.

- Decisión sobre la tecnología o librería a utilizar en el *frontend*, considerando el rendimiento y la complejidad, evitando agregar servidores adicionales si es posible.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

2. Estado del Arte

Con el fin de proponer una arquitectura, en este capítulo se describen los principios y el software que se usa en la cartografía web.

2.1. Geometría y Geografía

Los dos términos pueden hacer alusión a una gran cantidad de disciplinas y materias. Por eso, es importante establecer sus definiciones en el ámbito de los SIG.

En general, se identifican como geometría y geografía a dos formas distintas de localizar y representar un objeto real.

Una **geometría** o **dato geométrico** es la representación de una figura en un plano o espacio cartesiano. Estas figuras suelen ser representaciones de objetos reales. Por lo tanto, las geometrías son útiles para estudiar las características de dichos objetos como, por ejemplo, el área que ocupa un proyecto urbano, el perímetro de una zona de riesgo de incendios o la longitud de una calle. Al ser estas figuras representadas en planos cartesianos, sus características son siempre relativas, por eso las unidades de representación de las geometrías suelen ser los grados.

Una **geografía** o **dato geográfico** es la representación de la posición de un objeto relativo a la superficie terrestre. Este tipo de datos son útiles para representar posiciones en la esfera terrestre o calcular distancias entre ellas. Sin embargo, su uso no está muy extendido en el ámbito SIG porque conllevan una complejidad elevada y consecuentemente un coste computacional más elevado. La unidad más común para este tipo de datos es el metro, la unidad de longitud del sistema internacional de unidades.

2.2. Sistema de Información Geográfica (SIG)

Un sistema de información geográfica es un marco de trabajo (*framework*) para recopilar, gestionar o analizar datos. Los SIG pueden contener muchos tipos de datos, pero siempre con un aspecto en común, debe haber como mínimo un tipo de datos que sea de carácter espacial. Los datos espaciales se usan para analizar la ubicación y organizar capas de información en visualizaciones, utilizando mapas y escenas en 3 dimensiones.

Al principio, los SIG se crearon como aplicaciones de escritorio. Programas comerciales como *ESRI* y *ArcGIS* o programas de código abierto como *QGIS*, ofrecían la capacidad para insertar, gestionar, analizar y visualizar datos geográficos.

Los datos se insertan en estos programas mediante ficheros locales. Como ejemplos:

- Para insertar datos vectoriales, se usan archivos sencillos como los *Shapefile* o ficheros en formato *GeoJSON*.
- Para insertar datos ráster, se pueden usar archivos en formato *GeoTIFF*.

Muchos de estos formatos de ficheros fueron diseñados para trabajar en local y con el tiempo han dado paso a varios tipos de limitaciones:

- Límites espaciales para el tamaño de los ficheros.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

- Cantidad limitada de atributos o directamente no soportados.
- Poca estandarización. Uso de formatos binarios no transmisibles.
- Formatos multi-fichero.
- Estructuras de datos muy complejas.

A medida que la *WEB* ha evolucionado, también lo han hecho los *SIG*, y los defectos anteriores se han visto mitigados o solucionados. Para adaptarse a la nube, se han creado pilas de paquetes de software *SIG*, tanto comerciales como de código abierto (ver figura 1), que han dado lugar a la cartografía *WEB*, también conocida como **web-mapping**.

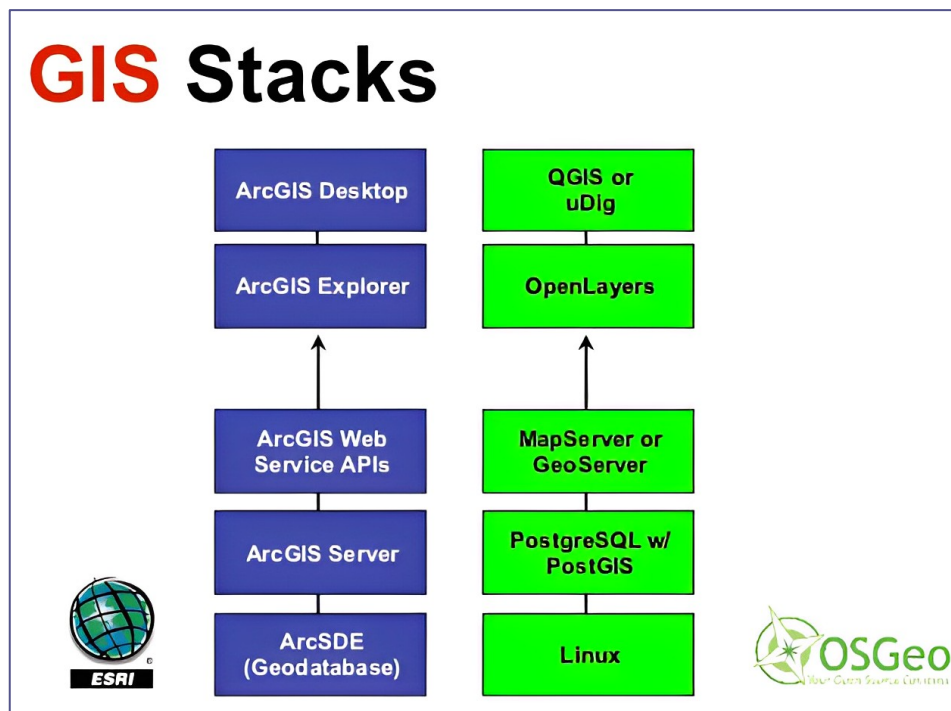


Figura 1. Ejemplo de pilas de software SIG. Comercial a la izquierda, de código abierto a la derecha.
Fuente: adaptada de [2]

2.3. Modelos de Datos Geográficos

En la actualidad, existen dos tipos de datos espaciales: los vectoriales y los rasterizados. A continuación, se proporciona una definición para cada uno de ellos.

2.3.1. Datos Vectoriales

Los datos vectoriales son estructuras de datos que ofrecen la capacidad de representar entidades geométricas a través de vértices. Un vértice es el nombre que recibe un punto localizado en un plano de dos dimensiones (X, Y) u, opcionalmente, en un plano de tres dimensiones (X, Y, Z). Adicionalmente, se puede adjuntar información descriptiva en forma de atributos.

Existen tres tipos de datos vectoriales:

- Punto. Geometría formada por un solo vértice.

- Línea (ver figura 2). Geometría formada por varios vértices en donde el primer y el último vértice son diferentes.

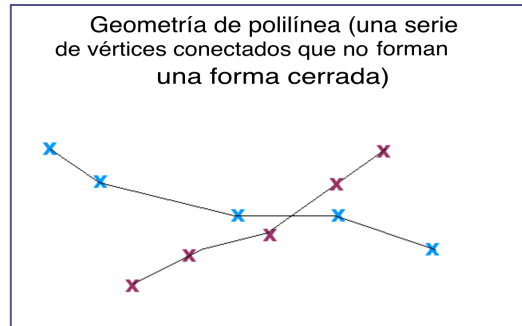


Figura 2. Ejemplo de geometría de tipo línea
Fuente: adaptada de [3]

- Polígono. (figura 3) Geometría formada por varios vértices donde el primer y último vértice son iguales.

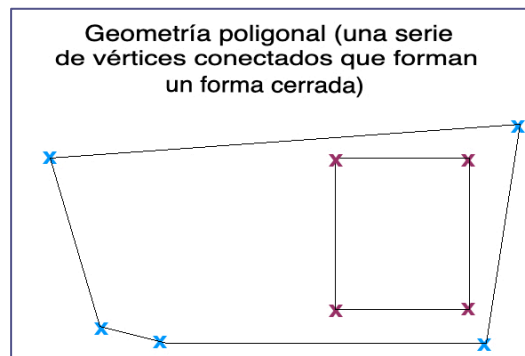


Figura 3. Ejemplo de geometría de tipo polígono.
Fuente: adaptada de [3]

Sus ventajas más importantes son:

- Se pueden escalar sin perder calidad.
- Ocupan poca memoria.
- Es posible optar por generar visualmente los datos en un servidor o en una máquina cliente, por ejemplo, un navegador web.
- Son idóneos para el análisis espacial. Por ejemplo, calcular distancias.

Sin embargo, también tienen sus limitaciones:

- Si son muy complejos, pueden requerir de una gran capacidad de cómputo.
- La mayoría de las aplicaciones SIG no permiten mezclar más de un tipo de geometría en una sola capa.

2.3.2. Datos Ráster

Un ráster o “*mapa de bits*” consta de una matriz de celdas (o píxeles) organizadas en filas y columnas (o una cuadrícula) en la que cada celda contiene un valor que representa información, como la temperatura.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

Los datos ráster se utilizan cuando se desea mostrar información que varía de forma continua, de manera que no resulta eficiente dividirla en entidades vectoriales. Por ejemplo, para representar la superficie terrestre, la forma más eficiente de hacerlo es usando datos ráster, porque está sufriendo cambios constantes. La estructura de los datos ráster es básicamente la misma que la de las imágenes fotográficas. Están compuestos por filas y columnas de píxeles (ver figura 4). Cada píxel representa una región geográfica, y el valor en ese píxel representa una característica de dicha región.

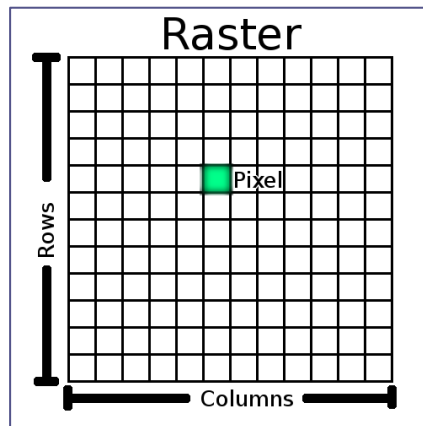


Figura 4. Estructura Ráster.
Fuente: [4]

Entre los usos más comunes:

- Gran utilidad para proporcionar información visual de forma adyacente a los datos vectoriales.
- Representación gráfica de información visual como, por ejemplo, los mapas de calor típicamente usados en meteorología.

Sin embargo, sufren de varias desventajas:

- El nivel de detalle esta siempre limitado por la resolución en la captura o generación del gráfico.
- Los conjuntos muy detallados pueden ocupar mucha memoria y afectar el rendimiento del SIG.
- Los conjuntos más ligeros sufren de pérdida de calidad si se altera la escala del mapa.

2.4. Proyección Cartográfica

La proyección cartográfica se puede describir como una serie de procesos que se usan para representar la superficie curva de la tierra en un plano bidimensional. Dicho de otra manera, mediante la proyección cartográfica se intenta representar la totalidad o partes de la superficie de la tierra.

El resultado de estos procesos son los mapas. Todos los mapas son representaciones, con mayor o menor precisión, de ciertos aspectos de la realidad. Están diseñados no sólo para representar objetos, sino también su forma y distribución espacial.

Para ilustrar, se tienen dos ejemplos de uso común en las figuras 5 y 6. La primera es la proyección de *Winkel-Tripel* y la segunda es la proyección de *Mercator*. Se puede observar como las características de la tierra, como la distancia o posición relativa de los continentes, es muy diferentes según el tipo de proyección. Esto no es un problema porque cada proyección tiene sus ventajas. Por ejemplo, la proyección de *Winkel-Tripel* es adecuada para el cálculo de distancias o superficies mientras que la de *Mercator* es ideal para la navegación porque permite trazar rumbos mediante líneas rectas.



Figura 5. Proyección de Winkel-Tripel.
Fuente: [5]

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

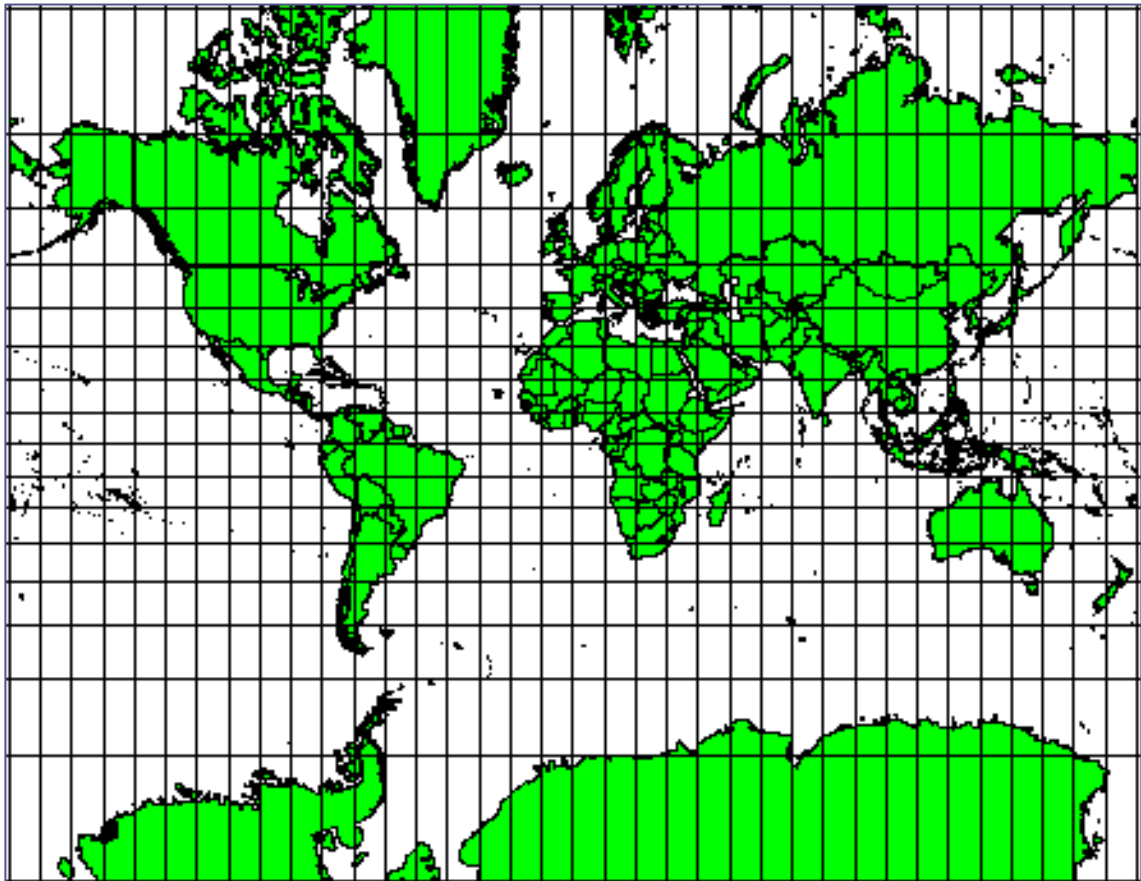


Figura 6. Proyección de Mercator.
Fuente: [6]

2.5. Sistema de Referencia de Coordenadas

Un sistema de referencia de coordenadas (*SRC*) establece, con la ayuda de coordenadas, la relación entre un mapa proyectado y lugares reales en la tierra.

Como se ha explicado en el sub-capítulo anterior, existen diferentes tipos de proyecciones cartográficas, con características dispares. La representación visual de una coordenada en un mapa no puede ser igual para dos mapas con diferente proyección. Se han desarrollado diferentes tipos de *SRC* para poder localizar los objetos en un mapa de forma adecuada.

Para decidir qué sistema de coordenadas de referencia usar, además de la proyección cartográfica, hay que tener en cuenta los siguientes aspectos:

- La extensión regional de la zona con la que se desea trabajar.
- El análisis que se quiere hacer.
- La disponibilidad de datos.

2.5.1. El Identificador de Referencia Espacial

Los *SRC* usados en las aplicaciones *SIG* llevan asociados un código exclusivo llamado *SRID*, el Identificador de Referencia Espacial. El *SRID* define los parámetros

del Sistema de Coordenadas de nuestros datos. Es muy importante elegir el SRID adecuado si se quiere representar o combinar diversas capas.

La mayoría de *SRID* han sido definidos por el *EPSG (European Petroleum Survey Group)*, una antigua organización científica relacionada con la industria petrolera de Europa. Estos códigos están disponibles en un registro¹ mantenido por el *IOGP*, la asociación internacional de productores de gas y petróleo.

Algunos de los sistemas de coordenadas más usados son:

- *EPSG:4326*. Usado entre otros para el Sistema de Posicionamiento Global (*GPS*)
- *EPSG:3857*. Utilizado por herramientas web como *Google Maps* u *OpenStreetMap*.

2.6. Base de Datos Espacial

Una **base de datos espacial** es una base de datos relacional con capacidad de usar, analizar o transformar datos espaciales. Esta capacidad no suele ser nativa por lo que debe hacerse uso de extensiones. En este tipo de bases de datos es necesario establecer el *SRID* para definir la localización y la relación entre los distintos objetos.

Hay tres características que definen a una base de datos espacial:

- Proporciona tipos de dato espaciales: geometrías y geografías.
- Proporciona funciones espaciales. Funciones capaces de crear, analizar, editar o transformar datos espaciales.
- Utiliza índices espaciales multidimensionales. (Los índices son un tipo de punteros que sirven para agilizar las consultas a una base de datos)

Las bases de datos espaciales son uno de los pilares fundamentales de la cartografía en la web. Se usan para reemplazar los antiguos ficheros locales y mitigar todas las desventajas que tenían estos. Además, ofrecen todas las ventajas de las bases de datos tradicionales:

- Reducción de errores y pérdidas de datos.
- Escalabilidad. Fáciles de extender o replicar.
- Recurrencia. Múltiples sistemas pueden acceder de forma paralela.
- Seguridad. Sistema de usuarios y permisos.
- Potentes herramientas para acceder a los datos.

Existe una gran variedad de bases de datos espaciales. Desde soluciones simples como *SQLite* y la extensión *Spatialite* hasta potentes bases de datos como *Oracle*, *MS SQL Server* o *PostgreSQL* ampliadas por extensiones como *ArcGIS* o *PostGIS*.

¹ <https://epsg.org/home.html>

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

2.7. Servidor Geoespacial

Un **servidor geoespacial**, también llamado **servidor de datos geográficos**, es un tipo de servidor *WEB* que dispone de un conjunto de funciones específicas para permitir a los usuarios compartir y editar datos geoespaciales. Al igual que los servidores *WEB*, están diseñados usando la arquitectura cliente-servidor y funcionan a través de los protocolos *HTTP* o su variante segura, *HTTPS*.

En general, el propósito de un servidor geoespacial es leer datos desde diversas fuentes, procesarlas y enviarlas a usuarios finales para ser consumidas (ver figura 7). Entre las fuentes de datos tenemos principalmente los servicios *WEB*, las bases de datos y los archivos locales. Al mismo tiempo, el servidor puede servir los datos utilizando los mismos formatos que ha recibido de entrada.

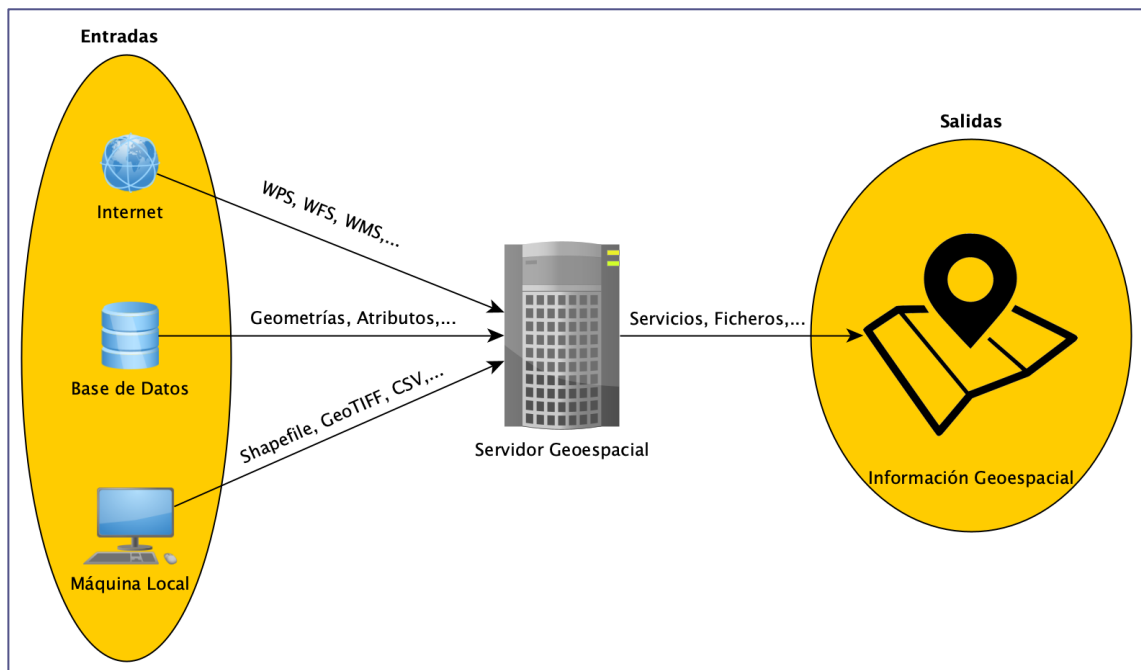


Figura 7. Ejemplos de datos de entrada y salida de un servidor geoespacial.

2.7.1. Capas

Las capas son conjuntos lógicos de datos geográficos que se utilizan para crear mapas y escenas.

Los tipos de datos que se pueden almacenar en una capa son muy variados. Se pueden tener desde entidades simples como fechas o números hasta objetos complejos como los datos vectoriales o rasterizados, mencionados en el subcapítulo 2.3.

Se pueden usar los mismos datos en varias capas, lo que permite crear diferentes visualizaciones y distintos tipos de capas a partir de ellos. Una capa se puede usar en diversos mapas.

2.7.2. Servicios Geoespaciales

Los servicios geoespaciales son un subconjunto de servicios *WEB* diseñados para permitir la comunicación e interoperabilidad entre distintos *SIG*. Son de especial interés

los definidos por el *Consortio Abierto Geoespacial*², conocido como OGC, al ser ampliamente usados por el sector. Entre los más destacados tenemos:

- **Web Map Service (WMS).**

Un servicio que sirve imágenes de mapas a partir de información geoespacial. Permite operaciones sencillas de visualización como activar o desactivar capas, controlar la transparencia, el orden o la escala. Un servidor geoespacial puede usar distintos WMS como fuente de datos, combinarlas y servir-las al usuario.

- **Web Map Tile Service (WMTS).**

Es un servicio que proporciona imágenes de mapas pre-generadas, divididas en teselas (*tiles*), para su visualización en aplicaciones y servicios WEB. Estas teselas son imágenes pequeñas y cuadradas que cubren diferentes áreas geográficas y niveles de *zoom*. Al utilizar un WMTS, se puede acceder a estas teselas de forma rápida y eficiente, permitiendo una visualización fluida y ágil de los mapas en diferentes escalas y áreas geográficas.

- **Web Coverage Service (WCS).**

Es un servicio que permite el acceso y la recuperación de datos y coberturas geoespaciales. A diferencia de los servicios anteriores que se centran en la visualización de mapas, el WCS se enfoca en proporcionar acceso a los datos subyacentes utilizados para generar mapas y modelos geoespaciales.

- **Web Feature Service (WFS).**

Es un servicio que permite la recuperación y edición de datos geoespaciales vectoriales, como puntos, líneas y polígonos, a través de servicios web. El WFS proporciona operaciones para consultar, filtrar, recuperar y actualizar características geoespaciales en tiempo real. Esto permite el acceso y la manipulación de datos geoespaciales más detallados y complejos, lo que resulta útil para aplicaciones que requieren interacción y edición de datos geográficos.

- **Web Processing Service (WPS).**

Es un servicio que permite la ejecución y el procesamiento de operaciones y algoritmos geoespaciales en un entorno distribuido. El WPS facilita el acceso a capacidades de procesamiento y análisis espacial a través de servicios WEB, lo que permite realizar tareas como análisis de proximidad, cálculo de rutas óptimas, interpolación espacial, entre otros

2.8. Los Visores

Un visor es un tipo de SIG que permite manipular datos geoespaciales. Es la parte del sistema con la que interactúa el usuario para explorar y analizar información geográfica de manera gráfica e interactiva. El visor SIG proporciona herramientas y funcionalidades que permiten realizar diversas operaciones, como:

- Visualizar mapas.
- Consultar o interactuar con información asociada a objetos o eventos geoespaciales.

² <https://www.ogc.org/standards/>

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

- Analizar datos.
- Compartir y colaborar.

Ejemplos de visores *SIG* populares incluyen:

- **QGIS.**

Un *software SIG* de escritorio, de código abierto y gratuito que permite visualizar, analizar y gestionar datos espaciales. Ofrece herramientas y funcionalidades que incluyen la creación y edición de mapas, el análisis espacial, la generación de informes y la interoperabilidad con otros formatos y *software SIG*.

- **ArcGIS Online.**

Una plataforma *SIG* en la nube que incluye un visor web con múltiples funcionalidades para visualizar y analizar datos geoespaciales, compartir mapas y colaborar en proyectos.

- **OpenLayers.**

Una biblioteca de *JavaScript* de código abierto que permite crear aplicaciones de mapas interactivos en la *WEB*, brindando herramientas para visualizar y manipular datos geoespaciales.

- **Google Earth.**

Una aplicación de software desarrollada por *Google* que permite explorar el mundo en forma de un globo virtual tridimensional. Utiliza imágenes satelitales y datos geoespaciales.

2.9. Aplicaciones Prácticas

La tecnología de los *Sistemas de Información Geográfica (SIG)* ha encontrado diversas aplicaciones prácticas en diferentes sectores. A continuación, se presentan algunos ejemplos:

- **Planificación urbana.**

Los *SIG* han sido ampliamente utilizados en la planificación urbana y el desarrollo de ciudades inteligentes. Permiten analizar y visualizar datos geoespaciales para la toma de decisiones relacionadas con el uso del suelo, la infraestructura, el transporte o el medio ambiente. Ejemplo de proyectos en este ámbito puede ser la web de servicios digitales de la diputación de Barcelona³.

- **Gestión de recursos naturales.**

Los *SIG* son usados en la gestión y conservación de recursos naturales como bosques, cuencas hidrográficas, reservas naturales, etc. Permiten monitorizar y analizar cambios en el paisaje, identificar áreas de importancia ecológica, evaluar la biodiversidad y tomar decisiones informadas para su preservación. Un ejemplo destacado es el Sistema de Información Forestal de Brasil⁴.

- **Gestión de desastres naturales.**

³ <https://serveisdigitals.diba.cat/>

⁴ <https://snif.florestal.gov.br/pt-br/estatisticas-florestais>

Los *SIG* desempeñan un papel crucial en la gestión de desastres naturales como terremotos, inundaciones e incendios forestales. Ayudan en la planificación de evacuaciones, identificación de áreas de riesgo, distribución de recursos de emergencia y seguimiento de la respuesta a los desastres. Un ejemplo el sistema de alerta y monitorización de tsunamis en el océano Pacífico, *NOAA*⁵, desarrollado por el departamento del comercio de los Estados Unidos

- **Gestión del transporte y logística.**

Los *SIG* son utilizados para optimizar la gestión, planificar rutas eficientes, evaluar el tráfico, gestionar flotas de vehículos y analizar la accesibilidad a diferentes áreas. Ejemplos de aplicaciones reconocidas en este campo son los sistemas de navegación *GPS* como *Google Maps* o *Waze*.

Estos ejemplos ilustran algunas de las aplicaciones prácticas de los *SIG* en distintos sectores. Su versatilidad los convierte en herramientas eficaces para la toma de decisiones basadas en información geoespacial precisa y actualizada.

⁵ <https://www.tsunami.gov/>

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

3. Análisis del Problema

La digitación y la digitalización ha traído como consecuencia que empresas y organismos de distinta índole, generen y dispongan, cada vez más, de una gran cantidad de datos y una gran cantidad de usuarios interesados en el análisis de dichos datos. Una proporción grande de estos datos incluyen atributos geográficos. En estos casos, resulta interesante organizar y representar estos datos de forma visual en un mapa. Esto ha dado lugar a una creciente demanda de soluciones *SIG* por parte de usuarios no expertos.

Antes del auge de la *WEB*, los programas *SIG* estaban restringidos a gente con conocimientos y formación en cartografía. Eran programas locales de difícil acceso para usuarios fuera del ámbito estrictamente cartográfico. Con el auge de la *WEB* una nueva familia de aplicaciones *SIG* fueron creadas y denominadas *web-mapping*.

Hoy en día se dispone de herramientas para crear, recopilar y editar datos de forma sencilla, cargarlos en servidores con capacidades geoespaciales y hacerlos accesibles de forma remota a una gran cantidad de usuarios.

Con este trabajo se quiere desarrollar una infraestructura que permita la visualización de información cartográfica de forma interactiva y a través de la web. Se quiere utilizar este *SIG* para la representación de datos a nivel de municipio, código postal o barrio, con el objetivo de identificar patrones, relaciones o situaciones, que facilitan la toma de decisiones.

3.1. Análisis del Marco Legal y Ético

En este apartado van a analizar algunos aspectos legales y éticos que pueden afectar a ciertos usos del proyecto.

3.1.1. Análisis de la Protección de Datos

En nuestras pruebas no usaremos datos de carácter personal o que puedan infringir derechos de propiedad intelectual.

Sin embargo, se debe mencionar que la seguridad en este tipo de aplicaciones debe ser ajustada según el origen y la naturaleza de los datos. Nuestra infraestructura tendrá capacidad de separación de proyectos, definición de usuarios y de roles. Según el caso, es importante aplicar estrictas políticas de protección y anonimización de datos.

También hay que mencionar que, como en todo sistema *WEB*, es recomendable escalar la seguridad de nuestra aplicación. Por ejemplo, configurar un cortafuego o establecer un perímetro de seguridad mediante una red privada virtual (*VPN*) para encapsular cualquier sistema que no necesite estar expuesto al público.

3.2. Identificación y Análisis de Soluciones Posibles

Existen diversas soluciones para llevar a cabo esta infraestructura. Desde el auge de internet, las soluciones *SIG* se han dividido en dos grandes bloques:

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

- En el primer bloque tenemos los SIG de escritorio. En la actualidad, son programas muy completos, capaces de conectarse a fuentes de datos remotas (*WFS*, *WPS*, etc.) o locales (*shapefiles*, *GeoTIFF*). Existen programas comerciales como ArcGIS⁶ o de código abierto como QGIS⁷ disponibles para el público. Sin embargo, no son los más indicados fuera del ámbito geográfico por culpa de una curva de aprendizaje moderada-alta y una limitada capacidad de compartir y exponer los datos al público general.
- La segunda opción y la de más interés para este proyecto son los SIG en la WEB. El perfil del usuario para este SIG es el público general. Por este motivo se quiere servir mapas de forma sencilla e intuitiva. Una base de datos espacial y un servidor geoespacial son los dos elementos imprescindibles para construir este tipo de soluciones. Ofrecen capacidad de almacenar datos recopilados desde diversas fuentes, combinarlos, procesarlos y configurar un acceso granular para usuarios o servicios. Múltiples tecnologías adicionales como *Leaflet* o *OpenLayers* pueden ser usadas para dotar los mapas de capacidades avanzadas de visualización o interactividad.

3.3. Solución propuesta

Teniendo en cuenta los objetivos del proyecto y las opciones tecnológicas disponibles, se ha decidido crear una infraestructura *web-mapping*. Para almacenar los datos, se utilizará una base de datos relacional, *PostgreSQL*⁸, que va a ser dotada de capacidades geoespaciales mediante la extensión *PostGIS*⁹. Estos datos serán accesibles desde el exterior gracias a un servidor geoespacial, *GeoServer*¹⁰. Se dispone así, de todas las herramientas necesarias para crear capas con el nivel de detalle deseado (provincial, municipal, barrio, etc.) y exponer los datos de forma interactiva para el usuario.

Opcionalmente, se utilizarán librerías de renderizado de capas (*Leaflet* o *OpenLayers*) para refinar la solución.

Para la validación de la infraestructura creada se realizará una prueba de concepto, en la que se ofrece la infraestructura y las pruebas a través de internet. Por esta razón, se ha decidido usar los servicios de alojamiento de *Google Cloud*¹¹.

3.4. Plan de trabajo

En este subcapítulo se va a detallar el plan de trabajo mediante una simple estructura de desglose del trabajo (*EDT*).

⁶ <https://www.esri.com/en-us/arcgis/products/arcgis-pro/overview>

⁷ <https://www.qgis.org/es/site/>

⁸ <https://www.postgresql.org/>

⁹ <http://postgis.net/>

¹⁰ <https://geoserver.org/>

¹¹ <https://cloud.google.com/?hl=es>

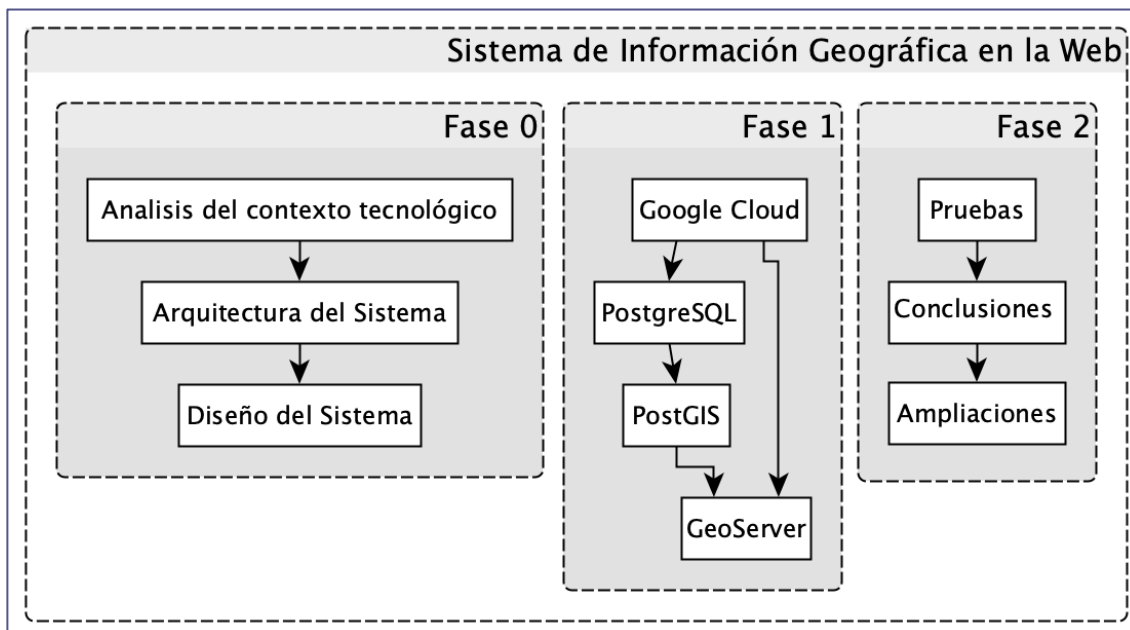


Figura 8. Esquema del Desglose del Trabajo.

En la figura 8 se puede observar el esquema que representa las distintas fases para la implementación del proyecto. Hay que tener en cuenta que se dispone de 360 horas para llevar a cabo todas las actividades. A continuación, se van a detallar cada una de las fases:

- **Fase 0**

En la fase inicial, se va a estudiar el contexto tecnológico y llevar a cabo un diseño conceptual de nuestro proyecto. Se necesita aprender los conceptos básicos sobre cartografía para luego hacer una investigación sobre los distintos paradigmas y tipos de tecnologías. Después del periodo de formación, se va a elegir una arquitectura para luego proceder a crear un diseño conceptual de un sistema que cumpla con los objetivos de este trabajo. Se estima una duración de 105 horas detalladas en la tabla 1.

Actividad	Duración estimada
Introducción e historia de los SIG	5 horas
Estudio de conceptos cartográficos	20 horas
Estudio de bases de datos espaciales	35 horas
Estudio de servidores SIG	30 horas
Estudio de tecnologías frontend SIG	15 horas

Tabla 1. Estimaciones de la duración de las actividades de la fase 1.

- **Fase 1**

A lo largo de esta fase se procederá con la creación y configuración de los sistemas, tecnologías y servicios elegidos. Se van a iniciar y configurar los servicios de *Google Cloud*, se creará la base de datos *PostgreSQL* y se va a configurar para el soporte de datos geoespaciales. A continuación, se debe crear una máquina virtual para albergar el servidor geoespacial *GeoServer* y luego, habilitar las conexiones con la base de datos y el exterior. Al final, se van a llevar

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

a cabo pruebas para determinar el correcto funcionamiento de todos los elementos. Se estima una duración de 65 horas detallada en la tabla 2.

Actividad	Duración estimada
Estudio de la tecnología de <i>GCP</i>	15 horas
Creación y configuración de la máquina virtual <i>Compute Engine</i>	10 horas
Instalación de <i>GeoServer</i> en <i>Compute Engine</i>	10 horas
Creación y configuración de la instancia <i>Cloud SQL</i> del tipo <i>PostgreSQL</i>	10 horas
Instalación y configuración de <i>PostGIS</i>	10 horas
Configuración de la red y el cortafuegos de nuestro proyecto <i>GCP</i>	5 horas
Prueba del sistema	5 horas

Tabla 2. Estimaciones de la duración de las actividades de la fase 1.

- **Fase 2**

Durante esta fase final, se van a desarrollar dos ejemplos para probar el funcionamiento y estudiar el potencial del *SIG*. Se crearán mapas para representar el riesgo de incendios en la Comunidad Valenciana y el tráfico en Valencia. Al final, se hará un análisis y se van a extraer conclusiones y hacer comentarios sobre posibles ampliaciones. Se estima una duración de 155 horas detallada en la Tabla 3.

Actividad	Duración estimada
Creación y pruebas del mapa de incendios de la Comunidad Valenciana.	45
Creación y pruebas del mapa de tráfico de Valencia.	30
Redacción de la memoria del proyecto.	80

Tabla 3. Estimaciones de la duración de las actividades de la fase 2.

En resumen, el plan de trabajo tiene una duración aproximada de 325 horas. Se espera que las 35 horas restantes sean dedicadas a reuniones con el tutor y la preparación de la defensa.

3.5. Presupuesto

En este apartado se procede a realizar una estimación del presupuesto asociado al desarrollo y mantenimiento del proyecto, considerando tanto los recursos humanos como los materiales necesarios.

Con respecto a los recursos humanos, se han usado datos extraídos de la *WEB indeed*¹² para estimar el coste por hora. Se sugiere que el proyecto cuente con un equipo compuesto por los siguientes miembros:

- Un administrador de sistemas para crear, configurar y mantener los servicios de *GCP*. Además, también tendrá que instalar, configurar y mantener el *GeoServer* y la base de datos *PostgreSQL*. El coste sería de aproximadamente 14,57 euros por hora.
- Un desarrollador *SQL* para crear y mantener los esquemas de la base de datos. Además, también tendrá que hacerse cargo de crear y mantener los procesos de extracción, transformación y carga de datos. El coste sería de aproximadamente 15,28 euros por hora.
- Un desarrollador *WEB* para desarrollar todos los elementos relacionados con la creación de los mapas cartográficos. Se estima un coste de 14,85 euros por hora.

Con respecto a los recursos informáticos, se han usado datos extraídos de las webs de *GCP*¹³ y *Lenovo*¹⁴. Los recursos necesarios serían:

- Una instancia del servicio *Compute Engine* de *GCP* en el que instalar el *GeoServer*. Tiene un coste aproximado de 0,04 euros por hora.
- Una instancia del servicio *Cloud SQL* de *GCP* que incluye la base de datos *PostgreSQL*. Tiene un coste aproximado de 0,09 euros por hora.
- Tres equipos de trabajo *Lenovo ThinkBook 14* con un coste unitario de 999 euros. Asumiendo un periodo de amortización de dos años y 1800 horas laborales anuales, se estima un coste de 0,28 euros por hora.

Resumiendo, el coste por hora estimado sería de 45,11 euros. Usando las estimaciones que se han hecho durante el apartado 3.4, calculamos que se van a necesitar 245 horas para la implementación. Se han descontado las horas necesarias para la redacción de la memoria y la preparación de la defensa. El coste final aproximado sería de 11.052 euros. Todos los impuestos están incluidos.

¹² <https://es.indeed.com/career/salaries>

¹³ <https://cloud.google.com/pricing?hl=es>

¹⁴ <https://www.lenovo.com/es/es/laptops/thinkbook/thinkbook-series/ThinkBook-14-Gen-4-14-inch-Intel/p/21DH00D1SP>

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

4. Diseño de la Solución

Este capítulo está dedicado a explicar con detalle el diseño seleccionado para la infraestructura. Primero, se va a crear una especificación de requisitos software para luego presentar la arquitectura, las tecnologías utilizadas y los detalles de la implementación.

4.1. Especificación de Requisitos

A continuación, se va a realizar la especificación de requisitos software para el proyecto siguiendo la norma *IEEE 830*.

1. Introducción.

1.1. Propósito.

Desarrollar un *SIG* que facilite la visualización de información a nivel de municipio, código postal y barrio.

1.2. Alcance.

Está destinado al público general y grupos de investigación como *FAKTUMLAB* del instituto *ITACA* de la *UPV*.

2. Descripción.

2.1. Perspectiva del producto.

La infraestructura bajo desarrollo facilitará la creación de aplicaciones que muestren información a nivel de municipio, código postal y barrio. Se podrá integrar con otras herramientas y sistemas a través de servicios *WEB*.

2.2. Funciones del producto.

Se ha creado en la figura 9 un diagrama con la gestión de procesos del *SIG* por parte de dos actores: el desarrollador y el usuario. Siendo el objetivo del producto la creación de un mapa interactivo, el desarrollador será el encargado de crear la base de datos y cargar los datos, además de crear los elementos del mapa y publicarlo. El usuario debe poder acceder al mapa y personalizar la visualización.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

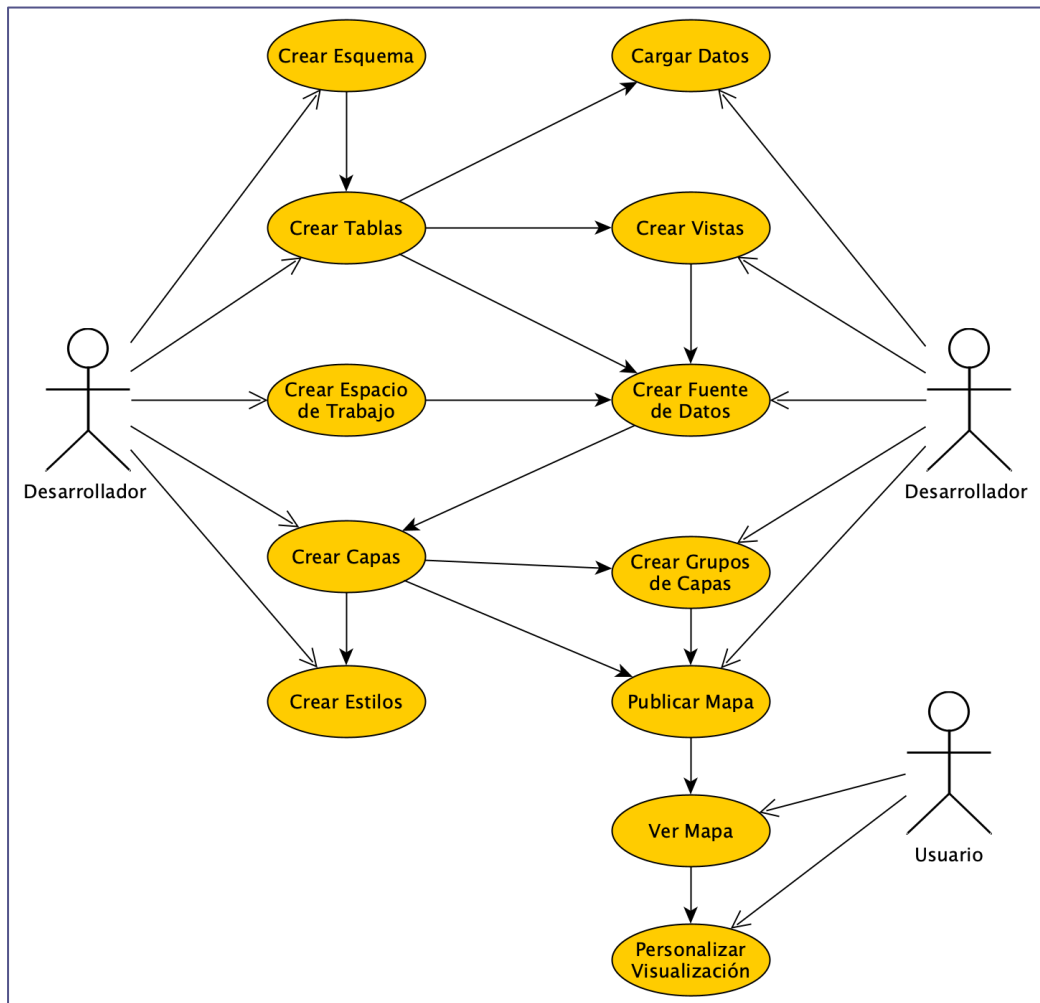


Figura 9. Gestión de procesos del producto SIG.

3. Requisitos específicos.

3.1. Requisitos funcionales.

- **RF1. Almacenamiento de datos geospaciales y de carácter general.**

El sistema debe ser capaz de almacenar los principales tipos de datos geospaciales, geometrías y geografías. El tamaño y la cantidad de datos pueden ser variados. El sistema debe soportar funciones espaciales como por ejemplo calcular la intersección o la distancia entre dos datos espaciales.

- **RF2. Visualización de datos geospaciales.**

El sistema debe ser capaz de representar los datos geospaciales de forma cartográfica. Debe soportar la creación y agrupación por capas además de la aplicación de estilos personalizados, como, por ejemplo, el uso de gradientes de color según.

- **RF3. Interactividad.**

El usuario puede hacer peticiones al sistema para mostrar información de interés relacionada a un dato espacial. Por ejemplo, el usuario puede seleccionar un objeto geométrico en un mapa y recibirá como respuesta

datos de interés que se hayan almacenado de forma adyacente al objeto geométrico.

- **RF4. Interoperabilidad.**

El sistema debe soportar y habilitar servicios *WEB* geoespaciales para poder transmitir la información espacial.

3.2. Requisitos no funcionales.

- **RNF1. Rendimiento.**

El sistema debe ser capaz de manejar cantidades moderadas de volúmenes de datos geoespaciales de manera eficiente y ofrecer tiempos de respuesta aceptables.

- **RNF2. Costes.**

El sistema debe tener un coste bajo, haciendo uso de *software* libre siempre que se pueda y manteniendo los gastos en infraestructura bajo control.

4.2. Arquitectura del Sistema

Considerando los requisitos establecidos, se ha elegido el modelo cliente-servidor por su versatilidad y sencillez. Para la arquitectura, se ha definido un sistema formado conceptualmente por tres elementos, observables en la figura 10.

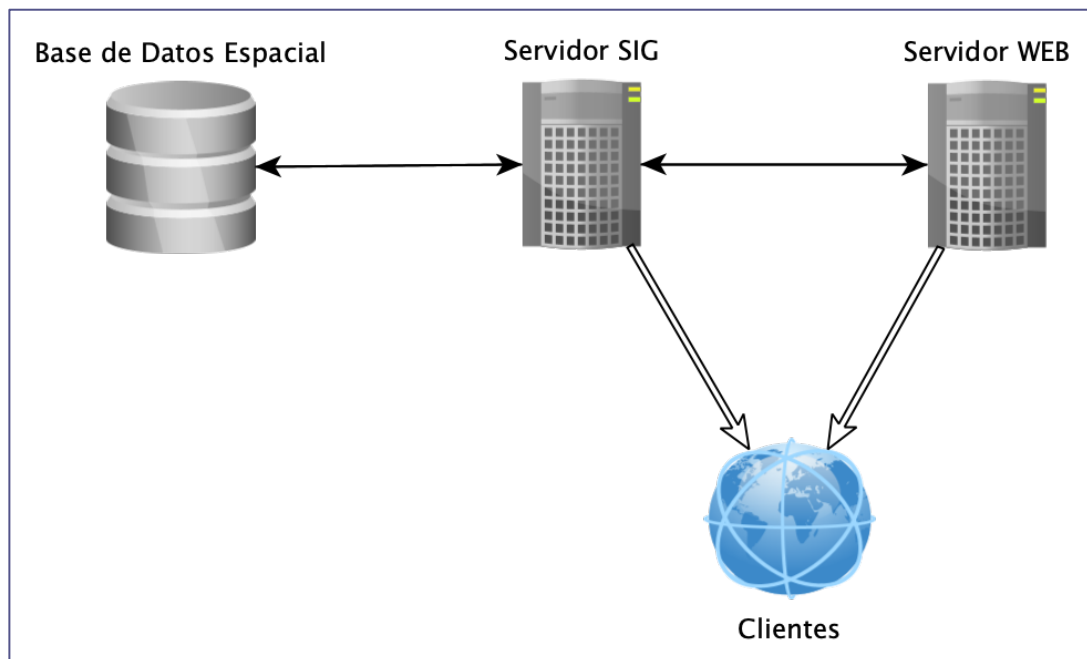


Figura 10. Componentes de la arquitectura del sistema.

Para este trabajo se ha decidido que la recopilación y carga de datos se va a hacer de forma manual dado que este aspecto queda fuera de los objetivos.

La base de datos y el servidor *SIG* se usarán para almacenar, definir y mostrar las capas de interés. La base de datos espacial almacenará los objetos geoespaciales y los datos asociados. El servidor *SIG* se encargará de leerlos y transformarlos en capas. Dichas capas podrán ser consultadas a través de servicios geoespaciales, específicamente habilitados en el servidor *SIG*.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

También se va a habilitar un servidor *WEB* que puede leer la información disponible en el *SIG* y transmitirla en forma de mapas a los clientes. Si bien esta acción puede completarse utilizando directamente el servidor *SIG*, el resultado puede mejorar substancialmente si se usan tecnologías específicas para la cartografía en la *WEB*. El servidor *WEB* nos facilitara esta tarea. El objetivo de este elemento será ofrecer capacidades de refinamiento que puede emplearse de forma opcional antes de publicar el mapa para el consumo de los clientes.

4.3. Diseño Detallado

A continuación, se detallan los componentes elegidos para llevar a cabo la implementación del *SIG* descrito en el apartado anterior. Las decisiones se han tomado en base a los requisitos establecidos en el apartado 4.1.

Con respecto al requisito funcional *RF1*, se ha elegido una base de datos *PostgreSQL* para almacenar los datos espaciales. Fundamental para esta tarea será el uso de la extensión *PostGIS*.

Para cumplir con los requisitos *RF2* y *RF3*, se ha elegido un servidor *GeoServer*. Si bien se entiende que las capacidades de este servidor serían suficientes para cumplir los requisitos, se ha decidido añadir un servidor local *Jetty* como apoyo. Se utilizará el servidor *Jetty* para refinar los resultados de *GeoServer* mediante la librería *Leaflet*.

Para poder analizar y cumplir con el requisito no funcional *RNF1*, se ha seleccionado *Google Cloud Platform (GCP)* como proveedor de computación en la nube. De esta manera se va a poder simular un entorno de producción.

Todo el *software* excepto los servicios de *GCP* es libre y de carácter abierto. Es necesario para mantener los costes reducidos y poder cumplir el requisito *RNF2*.

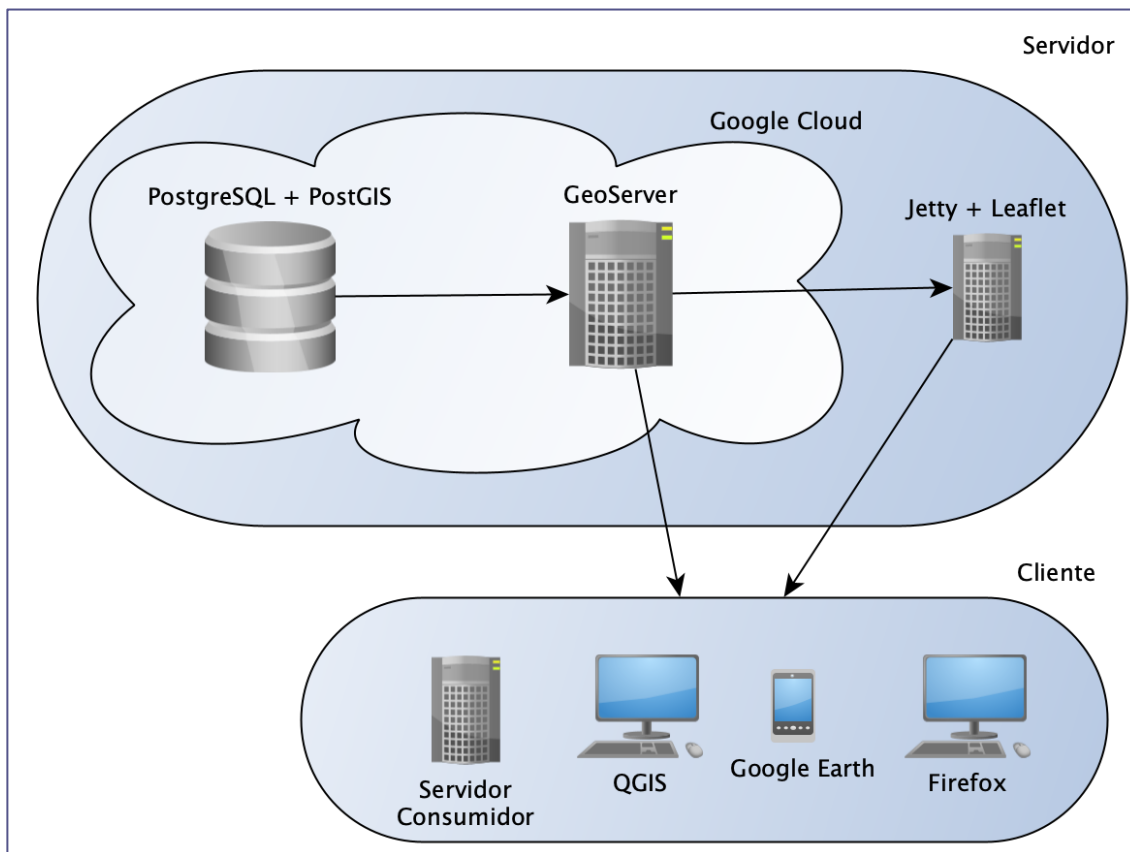


Figura 11. Esquema de sistemas e interacción "Cliente-Servidor".

4.4. Tecnología Utilizada

En este apartado se detallan las tecnologías usadas para desarrollar el sistema.

- **Google Cloud Platform (GCP)**

Es una plataforma que proporciona una multitud de servicios de computación en la WEB. GCP usa un modelo de distribución de *software* como servicio (SaaS). Se pueden crear máquinas virtuales, servidores o bases de datos relacionales, completamente gestionados y fácil de integrar, migrar o escalar, dependiendo de la necesidad. Los servicios que más interesan son:

- **Compute Engine**

Un servicio de computación con el que se pueden crear y ejecutar máquinas virtuales en la infraestructura de Google.

- **Cloud SQL**

Es un servicio totalmente gestionado de bases de datos relacionales de MySQL, PostgreSQL y SQL Server. Ejecuta las bases de datos, sus extensiones, sus marcas de configuración y su ecosistema de desarrollo, pero sin la dificultad de gestionarlas manualmente.

- **Debian 11**

Sistema operativo basado en un núcleo Linux. También conocido como Debian GNU/Linux, es un *software* de código abierto desarrollado y mantenido por una

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

amplia comunidad de voluntarios alrededor del mundo. Es un sistema operativo ligero y eficiente ideal para modelos “aaS” por favorecer el ahorro económico y la facilidad de escalar o migrar los equipos.

▪ **PostgreSQL**

Base de datos relacional de código abierto. Ofrece todas las capacidades esperadas de un gestor de bases de datos relacionales y se puede instalar en la mayoría de los sistemas operativos modernos. Dispone de una gran cantidad de extensiones, siendo una de ellas, *PostGIS*, de especial interés para este trabajo. Es ideal para almacenar grandes cantidades de datos estructurados y acceder a ellos de forma rápida y eficiente.

○ **PostGIS**

Es una extensión para *PostgreSQL*. Es un software de gran relevancia para el sector de los *SIG*, que se usa tanto en aplicaciones de código abierto como comerciales. Esto le confiere una gran ventaja por el uso de formatos de datos estandarizados que facilitan la compatibilidad entre diversos sistemas. Por ejemplo, datos de *OpenStreetMap*¹⁵ y *Mundialis*¹⁶ se pueden descargar, insertar en la base de datos usando un *SRC* común como el *EPSG:4326* y operar sobre ellos en conjunto.

Las características que aporta *PostGIS* a *PostgreSQL* se pueden resumir en:

- Tipos de datos espaciales, geometrías y geografías, para almacenar puntos, líneas, polígonos o coordenadas.
- Funciones espaciales para consultar las propiedades y relaciones espaciales. La mayoría de las funciones se pueden agrupar en una de las siguientes categorías: conversión, gestión, recuperación, comparación o generación.
- Índices espaciales que optimizan el rendimiento de las operaciones espaciales.

▪ **GeoServer**

Es un servidor de código abierto especializado en compartir y editar datos geoespaciales. Está diseñado para la interoperabilidad y es capaz de procesar, propagar o combinar datos desde diversas fuentes como, por ejemplo: *Google Maps*, *Bing Maps* o servicios *WFS*. Es un servidor con una comunidad activa y compatible con los protocolos que establece el *Open Geospatial Consortium*¹⁷. Para gestionar los datos, *GeoServer* ofrece diversos mecanismos:

- **El espacio de trabajo.** Un espacio separado y aislado relacionado con un proyecto.
- **El almacén de datos.** Un contenedor que sirve de conexión a una fuente de datos. Los datos deben estar en un formato vectorial o ráster. Las fuentes pueden ser ficheros, servicios *WEB* o esquemas de una base de datos.

¹⁵ <https://www.openstreetmap.org/export#map=13/39.4714/-0.3732>

¹⁶ <https://www.mundialis.de/en/ows-mundialis/>

¹⁷ <https://www.ogc.org/standards/>

- **Las capas.** Conjunto de datos ráster o vectoriales que contienen los elementos geográficos. *GeoServer* almacena los atributos, el encuadre, el tipo de proyección y los estilos asociados, en este objeto.
- **El grupo de capas.** Es una estructura que permite la agrupación jerárquica de capas o grupos de capas. Ha sido específicamente diseñado para simplificar las peticiones *WMS*, uno de los protocolos más usados en la cartografía *WEB*. El grupo hereda toda la información asociada a cada una de las capas utilizadas.
- **Los estilos.** Son directivas de visualización para la representación de los datos geoespaciales. Se aplican individualmente a las capas. *GeoServer* asocia siempre un estilo predeterminado a las capas. Por defecto, se utiliza un lenguaje llamado *Styled Layer Descriptor (SLD)*.

Es de interés mencionar que ciertas versiones de *GeoServer* incluyen un servidor *WEB (Jetty)*, al que vamos a dar uso en este proyecto.

- **Jetty**

También conocido como *Eclipse Jetty*, es un servidor *WEB* de código abierto escrito en *Java* y gestionado por la fundación *Eclipse*, que ofrece capacidades *WEB*. Puede servir datos de forma estática o dinámica. *Jetty* está más orientado a la comunicación entre maquinas o marcos de trabajo, a diferencia de los servidores *WEB* tradicionales.

- **Tomcat**

Es un potente servidor *WEB*, gratis y de código abierto, gestionado por la fundación de *software Apache*. Es sencillo de instalar y gestionar, y al ser de uso muy extendido, hay mucha documentación disponible. Será de utilidad para servir código *JavaScript* a usuarios finales.

- **Node.js**

Node es un entorno de ejecución de código *JavaScript* desarrollado especialmente para poder ejecutar código *JavaScript* en servidores. Permite una gestión de librerías modular, en forma de paquetes, que favorece la gestión y eficiencia del código.

- **OpenLayers**

Es una librería de código abierto para *JavaScript* que ofrece capacidades de visualización de mapas dinámicos en navegadores *WEB*. Ofrece conexión a servicios de capas teseladas, como por ejemplo los mapas de *OpenStreetMap*, y capas vectoriales clásicas, como las que pueden ser servidas por *GeoServer*, que luego se pueden combinar y dotar de interactividad.

- **Leaflet**

Leaflet es otra librería de código abierto para *JavaScript* que permite representar mapas en navegadores *WEB*. Es una librería ligera, especialmente diseñada para ser ejecutada en dispositivos móviles. Ofrece capacidades, parecidas a las de *OpenLayers*, de conectarse a servicios externos o internos para cargar capas que luego se pueden combinar y dotar de cierta interactividad. *Leaflet* también dispone de una amplia librería de extensiones para ampliar sus capacidades.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

- **QGIS-LTR**

QGIS es un *SIG* de escritorio, gratis y de código abierto, disponible para varios sistemas operativos. También hay versiones adaptadas para dispositivos móviles. Permite crear, editar, visualizar, analizar y publicar información geoespacial.

- **Lenguajes de Programación:**

- **SQL**

SQL es un lenguaje de programación que se usa para realizar consultas a bases de datos relacionales. Si bien esta estandarizado por la Organización Internacional de Normalización, bajo el estándar *ISO/IEC 9075*, no todos los gestores de bases de datos respetan la norma. *PostgreSQL*, gestor usado en nuestro proyecto, es uno de los pocos que intenta adherirse a la mayoría de los estándares.

- **HTML**

HTML es el lenguaje de marcado estándar para la elaboración de páginas *WEB*. Es un estándar gestionado por el consorcio *WWW (W3C)*. Es uno de los tres pilares de las aplicaciones web modernas.

- **CSS**

CSS es un lenguaje de programación usado para definir las características gráficas de los diversos elementos generados por un lenguaje de marcado como, por ejemplo, *HTML*. *CSS* es también uno de los tres pilares de la *WEB*. Sin embargo, su uso no es exclusivo para páginas o aplicaciones *WEB*. En nuestro trabajo nos será útil para dotar nuestros mapas de una mayor capacidad de representar visualmente la información.

- **Styled Layer Descriptor (SLD)**

SLD es un lenguaje de marcado para datos geoespaciales basado en *XML* y gestionado por la *OGC*. Es el lenguaje por defecto de *GeoServer* para aplicar estilos a las capas.

- **JavaScript**

Es un lenguaje de programación para aplicaciones *WEB*. Ha sido adoptado por todos los navegadores *WEB*, siendo junto a *CSS* y *HTML* uno de los tres pilares de la *WEB*. Es un lenguaje de alto nivel, compilado en tiempo de ejecución, generalmente usado en el lado del cliente. Sin embargo, también existen motores de ejecución para *JavaScript* en el lado del servidor, siendo el más famoso, *Node.js*.

- **Entornos de Desarrollo:**

- **pgAdmin 4**

Un programa de escritorio para la administración de bases de datos *PostgreSQL*. Es un software libre y de código abierto escrito en *Python* y *JavaScript*, y disponible para diversos sistemas operativos.

- **DBeaver CE**

DBeaver en su edición de la comunidad, es una suite de *software*, gratuita y de código abierto, para la administración de bases de datos. Es un programa de escritorio, disponible para los sistemas operativos más

extendidos. Si bien es de código abierto, las conexiones a bases de datos comerciales como *OracleDB* pueden requerir controladores propietarios. Puede gestionar tanto bases de datos relacionales como alternativas (*NoSQL*). Lo hemos usado en el proyecto como alternativa a *pgAdmin 4* gracias a su mejor capacidad de gestionar consultas y cargas de datos de gran tamaño.

- **Sublime Text**

Es un editor de texto ligero basado en *Vim*. A pesar de ser *software* propietario, se ofrece en una versión de prueba ilimitada que resulta suficiente para las necesidades de este proyecto.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

5. Desarrollo de la Solución Propuesta

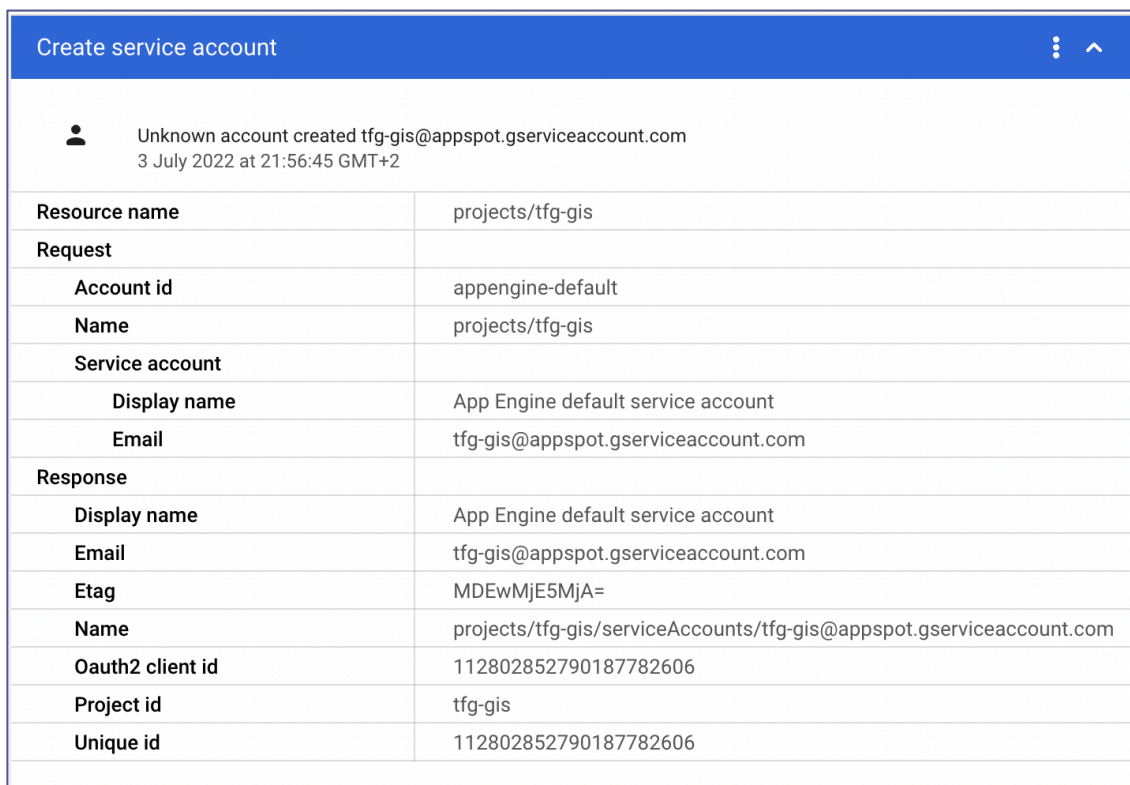
En este capítulo se va a describir la instalación y configuración de todos los componentes necesarios para la implementación de la infraestructura.

5.1. Configuración de los servicios de *Google Cloud Platform*

Para poder hacer uso de los servicios de *GCP* se necesita usar una cuenta de *Google* y registrarla en la plataforma.

5.1.1. Nuevo Proyecto en *GCP*

Una vez registrados, el primer paso es crear un nuevo proyecto mediante el gestor de recursos de *Google Cloud*. El nombre de nuestro proyecto será “**tfg-gis**”.




Create service account	
	Unknown account created tfg-gis@appspot.gserviceaccount.com 3 July 2022 at 21:56:45 GMT+2
Resource name	projects/tfg-gis
Request	
Account id	appengine-default
Name	projects/tfg-gis
Service account	
Display name	App Engine default service account
Email	tfg-gis@appspot.gserviceaccount.com
Response	
Display name	App Engine default service account
Email	tfg-gis@appspot.gserviceaccount.com
Etag	MDEwMjE5MjA=
Name	projects/tfg-gis/serviceAccounts/tfg-gis@appspot.gserviceaccount.com
Oauth2 client id	112802852790187782606
Project id	tfg-gis
Unique id	112802852790187782606

Figura 12. Información del proyecto “tfg-gis”

5.1.2. Nueva Instancia de Servicio *Cloud SQL*

Como se ha comentado en el subcapítulo 3.3, la base de datos relacional elegida para el proyecto es *PostgreSQL*. Para crear la base de datos se utilizará el servicio *Cloud SQL* de *GCP*. Mediante el asistente de servicios de *Cloud SQL* de *GCP*, se crea una nueva instancia llamada “**postgis-server**”. La versión de *PostgreSQL* elegida es la

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

14.4. Las características de la máquina, que se pueden consultar en la figura 14, son modestas para intentar limitar los costes.

The screenshot displays the configuration page for an AWS RDS instance named 'postgis-server'. It is divided into two main sections: 'Instance info' and 'Summary'.

Instance info:

- Instance ID:** postgis-server
- PASSWORD POLICY:** Expanded, showing 'Database version' as PostgreSQL 14. A note states: 'This instance has the latest supported version'.
- Choose region and zonal availability:** A section with a note: 'For better performance, keep your data close to the services that need it. Region is permanent, while zone can be changed any time.' It includes a 'Region' dropdown set to 'europa-west3 (Frankfurt)' and 'Zonal availability' options: 'Single zone' (selected) and 'Multiple zones (highly available)'. A note for 'Single zone' says: 'In case of outage, no failover. Not recommended for production.' A note for 'Multiple zones' says: 'Automatic failover to another zone within your selected region. Recommended for production instances. Increases cost.'

Summary:

Region	europa-west3 (Frankfurt)
DB version	PostgreSQL 14.4
vCPUs	1 vCPU
Memory	3.75 GB
Storage	20 GB
Network throughput (MB/s)	250 of 2,000
Disk throughput (MB/s)	Read: 9.6 of 240.0 Write: 9.6 of 72.0
IOPS	Read: 600 of 15,000 Write: 600 of 4,500
Connections	Public IP
Backup	Automated
Availability	Single zone
Point-in-time recovery	Enabled

Figura 13. Información de la instancia "postgis-server".

Ahora, hay que dirigirse al submenú de las conexiones, mostrado en la figura 15, para continuar con la configuración. Se necesita pedir la asignación de una IP pública para poder acceder a la base de datos desde internet. A la instancia Cloud SQL se le ha asignado la dirección "34.141.86.229".

Sin embargo, aún no se tiene acceso desde el exterior de la red de la instancia. Cloud SQL tiene un cortafuegos habilitado por defecto al que se tiene que proporcionar rangos de direcciones IP para permitir conexiones entrantes. Para este proyecto, se ha habilitado un rango que permita acceso desde las máquinas de desarrollo y una dirección IP que habilite la comunicación con el servidor geográfico.

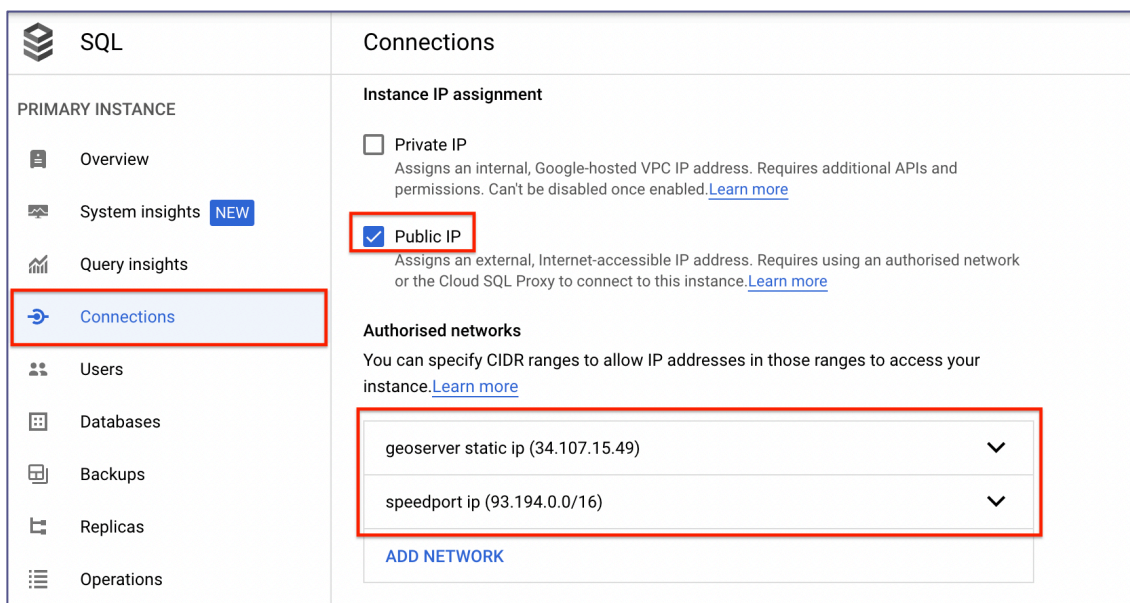


Figura 14. Configuración de conexiones de la instancia “postgis-server”.

La configuración de la instancia *Cloud SQL* ha sido completada.

5.1.3. Nueva Instancia de Servicio *Compute Engine*

Se procede con la creación de una nueva instancia *Compute Engine*, llamada “**gis-geo-server**”, usando el asistente de *Google Cloud*. Se elige una maquina “*E2-medium*” situada en la región “*europa-west3-c*” por su proximidad al equipo de desarrollo. Al igual que la base de datos, se puede observar en la figura 16 que es una máquina relativamente modesta. El motivo principal es contener los costes.

Monthly estimate	
US\$32.71	
That's about US\$0.04 hourly	
Pay for what you use: No upfront costs and per-second billing	
Item	Monthly estimate
2 vCPU + 4 GB memory	US\$31.51
10 GB balanced persistent disk	US\$1.20
Sustained use discount	-US\$0.00
Total	US\$32.71

Figura 15. Estimación de costes de la instancia “gis-geo-server”.

El sistema operativo será un *Debian GNU/Linux*, versión 11. Se ha elegido una distribución *Debian* por ser estable, segura y sobre todo ligera. Es un sistema con una gran comunidad detrás y que ha servido de base para muchas de las distribuciones de *Linux* más populares.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

Con estos datos, *Google Cloud* ya puede iniciar el proceso de creación de la instancia. Es un proceso que puede llegar a durar varias decenas de minutos dependiendo del tamaño de la instancia.

Una vez creada la instancia, hay que dirigirse al menú “*editar*” de la instancia y en las opciones de “*Networking*”, subsección “*Firewalls*”, se debe habilitar el tráfico “*HTTP*” y “*HTTPS*”.

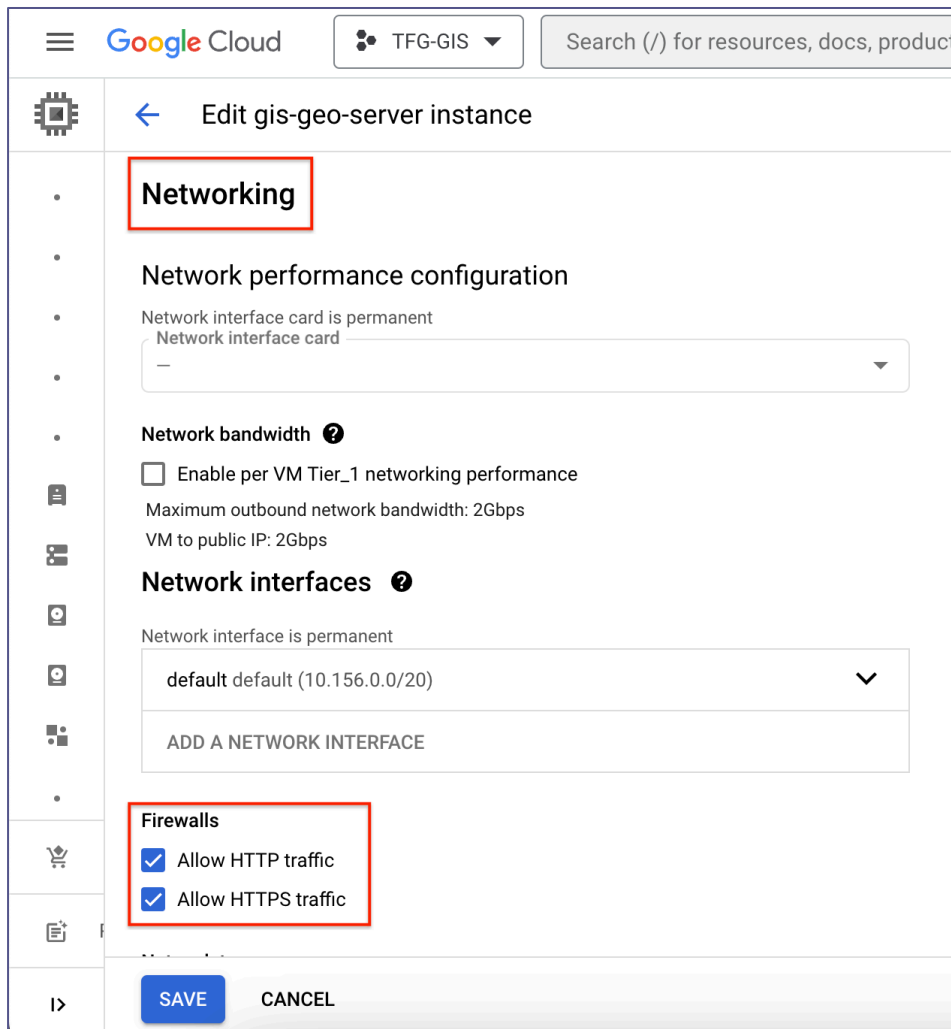


Figura 16. Configuración de red de la instancia “gis-geo-server”

Por último, en el menú de interfaces de red, se puede consultar la dirección *IP* externa que ha sido asignada a la instancia: “34.107.15.49”.

The screenshot shows the Google Cloud console interface for a network interface. At the top, there's a search bar and navigation options. Below that, the page title is 'Network interface details'. A dropdown menu shows 'Selected network interface: nic0'. The main content area is titled 'Network interface details' and contains a table with the following data:

Name	Network	Subnetwork	Primary internal IP address	Alias IP ranges	External IP address	Network Service Tier
nic0	default	default	10.156.0.2	–	34.107.15.49	Premium

Figura 17. Información sobre las interfaces de red de la instancia “gis-geo-server”.

La máquina virtual esta lista. Se puede usar una conexión *SSH* para acceder a ella.

The screenshot shows a terminal window titled 'SSH-in-browser'. The terminal output is as follows:

```
Linux gis 5.10.0-16-cloud-amd64 #1 SMP Debian 5.10.127-2 (2022-07-23) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Aug 11 13:41:43 2022 from 35.235.241.178
-bash: export: `.`: not a valid identifier
-bash: export: `/home/alex_ionita_1991/.profile': not a valid identifier
alex_ionita_1991@gis:~$
```

Figura 18. Conexión remota a la máquina virtual “gis-geo-server”.

5.2. Configuración de *PostgreSQL*

Con la instancia *PostgreSQL* en ejecución, el primer paso será crear una base de datos. El nombre utilizado ha sido “*tfg*”. La acción se puede llevar a cabo utilizando el código de la figura 20.



Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

```
-- Database: tfg

-- DROP DATABASE IF EXISTS tfg;

CREATE DATABASE tfg
  WITH
  OWNER = postgres
  ENCODING = 'UTF8'
  LC_COLLATE = 'en_US.UTF8'
  LC_CTYPE = 'en_US.UTF8'
  TABLESPACE = pg_default
  CONNECTION LIMIT = -1;

COMMENT ON DATABASE tfg
  IS 'Base de datos para las capas del tfg';
```

Figura 19. Código SQL para la creación de la base de datos “tfg”.

Una vez creada la base de datos, se necesita crear y asignar las extensiones de *PostGIS* a la base de datos usando el código indicado en las figuras 21 y 22. Se seleccionan las extensiones “*postgis*” y “*postgis_raster*”, que proporcionan tipos de datos y funciones para datos geográficos y geométricos.

```
1  -- Extension: postgis
2
3  -- DROP EXTENSION postgis;
4
5  CREATE EXTENSION IF NOT EXISTS postgis
6     SCHEMA public
7     VERSION "3.1.4";
8
```

Figura 20. Código SQL para crear la extensión “postgis”.

```

1  -- Extension: postgis_raster
2
3  -- DROP EXTENSION postgis_raster;
4
5  CREATE EXTENSION IF NOT EXISTS postgis_raster
6      SCHEMA public
7      VERSION "3.1.4";
8

```

Figura 21. Código SQL para agregar la extensión “postgis_raster”.

Ejecutando la siguiente consulta SQL se pueden averiguar las extensiones instaladas: “select * from pg_available_extensions where installed_version is not null;”. En la figura 23 se pueden observar las dos extensiones de PostGIS que se han instalado.

```

1  SELECT * FROM pg_available_extensions where installed_version is not null;

```

	name	default_version	installed_version	comment
	name	text	text	text
1	tablefunc	1.0	1.0	functions that manipulate whole tables, including crosstab
2	plpgsql	1.0	1.0	PL/pgSQL procedural language
3	postgis	3.1.4	3.1.4	PostGIS geometry and geography spatial types and functions
4	postgis_raster	3.1.4	3.1.4	PostGIS raster types and functions

Figura 22. Código SQL para consultar las extensiones instaladas en la base de datos.

Una segunda consulta que se puede ejecutar para asegurarnos de que las extensiones se han instalado debidamente es: “select * from information_schema.views where table_catalog='tfg' and table_schema='public';”. Cuando se instalan las extensiones, las vistas resaltadas en la figura 24 deben ser creadas en el esquema público de la base de datos.

```

3  select * from information_schema.views where table_catalog='tfg' and table_schema='public';

```

	table_catalog	table_schema	table_name	view_definition
	name	name	name	character varying
1	tfg	public	geography_columns	[...] SELECT current_database() AS f_table_catalog,
2	tfg	public	geometry_columns	[...] SELECT (current_database())::character varying(256) AS f_table_catalog,
3	tfg	public	riesgo_dia_municipio_2021_07_31	[...] SELECT mc.geom,
4	tfg	public	raster_columns	[...] SELECT current_database() AS r_table_catalog,
5	tfg	public	raster_overviews	[...] SELECT current_database() AS o_table_catalog,

Figura 23. Código SQL para consultar la correcta creación de las vistas asociadas a las extensiones “postgis” y “postgis_raster”.

Finalmente, la base de datos está preparada para datos y funciones geoespaciales.



Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

5.3. Instalación del *GeoServer*

Instalar un servidor *GeoServer* es relativamente sencillo. En este caso, se ha elegido la última versión estable, “2.21.0”, desde el directorio de descarga de la *WEB*¹⁸ de *GeoServer*. Los pasos por seguir son:

- Acceder a la máquina virtual “gis-geo-server” mediante la herramienta para conexiones *SSH* de *Google Cloud*.
- Crear un nuevo directorio “/usr/shr/geoserver” y enlazarlo a una nueva variable del entorno llamada “*GEOSERVER_HOME*” (ver figura 25).

```
alex_ionita_1991@gis:/usr/share$ echo "export GEOSERVER_HOME=/usr/share/geoserver" >> ~/.profile
. ~/.profile
alex_ionita_1991@gis:/usr/share$ echo $GEOSERVER_HOME
/usr/share/geoserver
```

Figura 24. Creación de la variable de entorno *GEOSERVER_HOME*.

- Descargar el archivo que contiene el binario del *GeoServer* y proceder a descomprimirlo en el directorio “/usr/shr/geoserver”.

```
alex_ionita_1991@gis:/home$ ls -l /usr/share/geoserver/
total 108040
-rwxrwxrwx 1 alex_ionita_1991 root      28637 May  7 18:41 NOTICE.md
-rwxrwxrwx 1 alex_ionita_1991 root        425 May  7 18:41 README.txt
-rwxrwxrwx 1 alex_ionita_1991 root     3945 Apr 21 06:09 RUNNING.txt
-rwxrwxrwx 1 alex_ionita_1991 root       331 May 22 09:43 VERSION.txt
drwxrwxrwx 2 alex_ionita_1991 root      4096 May 22 09:51 bin
drwxrwxrwx 20 alex_ionita_1991 root     4096 Aug 16 15:56 data_dir
drwxrwxrwx 2 alex_ionita_1991 root      4096 Apr 21 06:09 lib
-rwxrwxrwx 1 alex_ionita_1991 alex_ionita_1991 110376595 Jun 29 14:45 geoserver-2.21.0-bin.zip
drwxrwxrwx 2 alex_ionita_1991 root      4096 May 22 09:51 lib
drwxrwxrwx 2 alex_ionita_1991 root      4096 May 22 09:51 libexec
drwxrwxrwx 2 alex_ionita_1991 root      4096 Apr 21 06:09 logs
drwxrwxrwx 2 alex_ionita_1991 root      4096 Apr 21 06:09 modules
drwxrwxrwx 2 alex_ionita_1991 root      4096 Apr 21 06:09 webapp
-rwxrwxrwx 1 alex_ionita_1991 root       5697 Apr 21 06:09 start.ini
-rwxrwxrwx 1 alex_ionita_1991 root    163958 May  7 18:41 start.jar
drwxrwxrwx 3 alex_ionita_1991 root      4096 May 22 09:51 webapp
```

Figura 25. Binarios del *GeoServer* descomprimidos.

- Asignar al usuario de la máquina, en este caso “alex_ionita_1991”, como propietario del directorio “/usr/shr/geoserver” y, de forma recurrente, de todos los elementos contenidos dentro de dicho directorio (ver figura 27).

```
alex_ionita_1991@gis:/home$ sudo chown -R alex_ionita_1991 /usr/share/geoserver/
```

Figura 26. Asignación de nuevo dueño al directorio.

- Iniciar el servidor ejecutando el script “startup.sh” localizado dentro del directorio “/usr/share/geoserver/bin”.

```
alex_ionita_1991@gis:/home$ sh /usr/share/geoserver/bin/startup.sh &
```

Figura 27. Arranque del *GeoServer*.

Para comprobar la correcta instalación, se puede acceder a la siguiente dirección a través de un navegador *WEB*: “<http://34.107.15.49:8080/geoserver/web/?0>”. Esta dirección debe llevar a la pantalla de bienvenida del servidor. Se puede observar en la figura 29 que el *GeoServer* ha sido instalado con éxito.

¹⁸ <https://geoserver.org/download/>

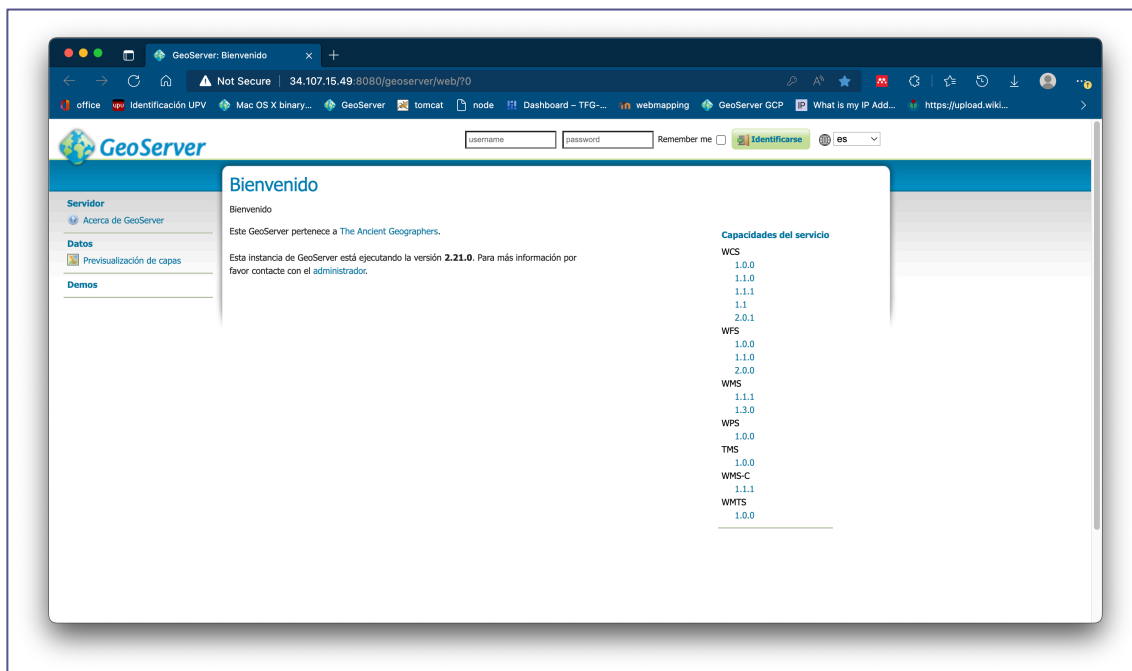


Figura 28. Pantalla de inicio de GeoServer.

El último paso será instalar un conjunto de extensiones de utilidad para el proyecto. Hay que acceder al directorio de descargas de código fuente en la *WEB* de GeoServer¹⁹ y descargar las siguientes extensiones:

- **CSS Styling** desde el apartado “*Cartography*”. Una extensión que habilita el uso de un lenguaje derivado de *CSS* para aplicar estilos a nuestras capas.
- **WPS**, localizado en el apartado “*Services*”. Una extensión que habilita el uso del servicio *Web Processing Service (WPS)*.

Los archivos con las extensiones deben ser extraídos dentro del directorio “\GeoServer\webapps\geoserver\WEB-INF\lib” y luego se debe reiniciar el GeoServer.

Merece atención el directorio “\GeoServer\webapps\geoserver\data_dir\www” que se puede usar para almacenar y ejecutar ficheros *HTML*. GeoServer contiene *Jetty*, un servidor *WEB* que ejecuta estos archivos y los sirve a través del enlace “*http://<host:port>/geoserver/www*”.

¹⁹ <https://geoserver.org/release/2.21.0/>

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

6. Validación

En este capítulo, se va a mostrar y validar el correcto funcionamiento del *SIG* creado, además de analizar si cumple con los objetivos que se han propuesto en el capítulo 1. Se han obtenido datos para proponer dos casos de uso que pueden resultar de interés dentro del contexto de este proyecto, dichos casos son:

- Creación de un mapa interactivo de la Comunidad Valenciana donde poder consultar el riesgo de incendio. En el mapa se representará información a nivel de municipio y se le aplicará un estilo que represente la correspondiente situación de riesgo de un incendio.
- Creación de un mapa interactivo de la localidad de Valencia para poder consultar el tráfico de determinados tramos. En este mapa se podrá visualizar información a nivel de barrio y también se le aplicará un estilo para crear señas visuales que permitan identificar, de forma aproximada, el caudal de tráfico de cada tramo.

6.1. Riesgo de Incendio por Municipio en la Comunidad Valenciana

Esta prueba consiste en la creación de un mapa interactivo de riesgos de incendio en la Comunidad Valenciana. Se van a representar los riesgos a nivel de municipio. Específicamente, el objetivo será representar el riesgo máximo por municipio del último día del que se disponga de datos. La representación debe ser interactiva además de proporcionar información de manera visual.

6.1.1. Los Datos

Se van a organizar los datos de esta prueba dentro de un nuevo esquema, llamado “*com_valenciana*”, dentro de la base de datos.

```
5 CREATE SCHEMA IF NOT EXISTS com_valenciana
6 AUTHORIZATION postgres;
```

Figura 29. Código SQL para la creación del esquema “*com_valenciana*”.

6.1.1.1. Municipios de la Comunidad Valenciana

Se debe crear una tabla en la que almacenar las geometrías y atributos de los municipios. La información sobre los municipios se ha descargado desde la aplicación *WEB*²⁰ que *ESRI* ha habilitado para compartir datos geográficos de carácter abierto. Este paquete de datos contiene todos los municipios de España.

²⁰<https://opendata.esri.es/datasets/municipios-ign/explore?location=38.981766%2C-0.857376%2C6.79>

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

Se crea una tabla llamada “*municipios_cv*” (ver figura 31) en la que se guardan los datos filtrados, correspondientes a la Comunidad Valenciana, del paquete de datos descargado.

	Column Name	#	Data type
Columns	gid	1	int4
Constraints	objectid	2	int2
Foreign Keys	inspireid	3	varchar(24)
Indexes	natcode	4	varchar(11)
Dependencies	nameunit	5	varchar(81)
References	codnut1	6	varchar(3)
Partitions	codnut2	7	varchar(4)
Triggers	codnut3	8	varchar(5)
Rules	codigoine	9	varchar(5)
Statistics	shape_leng	10	numeric
Permissions	shape_area	11	numeric
DDL	geom	12	geometry(multipolygon, 4326)
Virtual			

Figura 30. Estructura de la tabla “*municipios_cv*”.

Entre otros, se dispone de datos como el nombre del municipio, “*nameunit*”, el código postal, “*codigoine*”, o las propias geometrías para la representación vectorial de cada municipio. Se puede usar el visor de geometrías de *DBeaver* (ver figura 32) para observar la representación cartográfica de las geometrías de la Comunidad Valenciana.

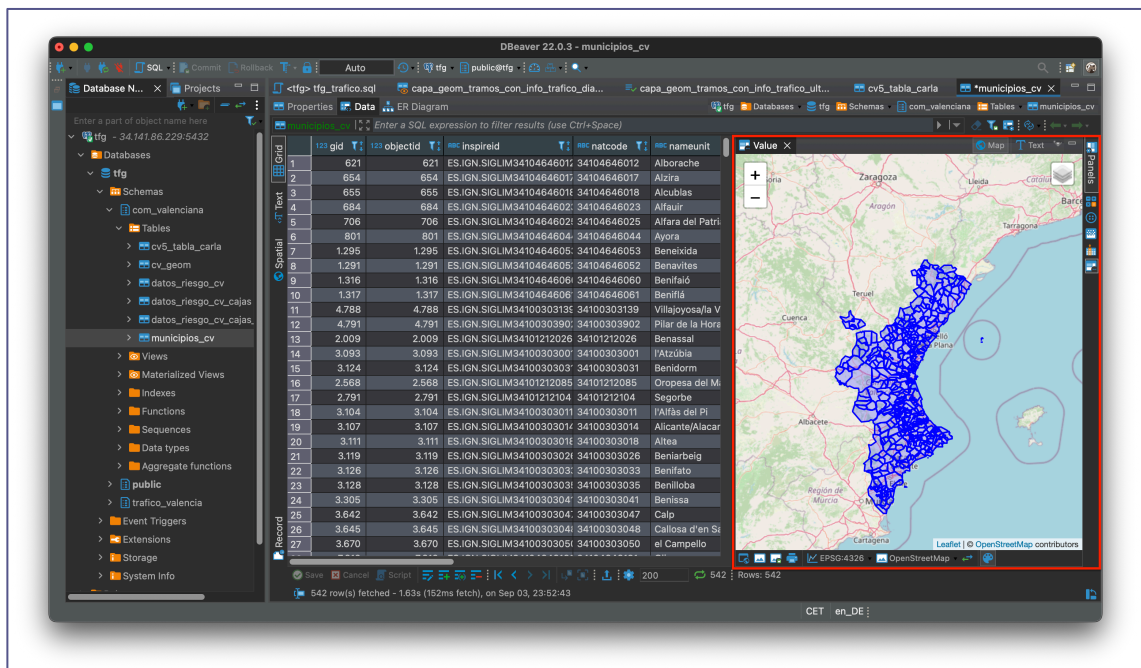


Figura 31. Geometrías de la tabla “*municipios_cv*” representadas por el visor de mapas de *DBeaver*.

6.1.1.2. Riesgo de Incendios

Este paso consiste en crear una tabla, en la que almacenar la información del riesgo de incendios. Este segundo conjunto de datos ha sido proporcionado por el tutor del proyecto.

Contiene aproximadamente 576.000 puntos geográficos (ver figura 33), entre las latitudes 0,95° y -2° y las longitudes 41° y 37,05°, que corresponden a un intervalo temporal iniciado el día 31 de julio de 2021 a las 02:00h hasta el día 5 de agosto de 2021 a las 01:00h. Para cada punto e intervalo temporal se dispone de información sobre la humedad relativa, la temperatura, el riesgo de tormenta o el riesgo de incendio, siendo este último indispensable para nuestra prueba. Se procede a eliminar los datos no pertenecientes a la Comunidad Valenciana y se carga el resultado en una tabla llamada “*datos_riesgo_cv*”.

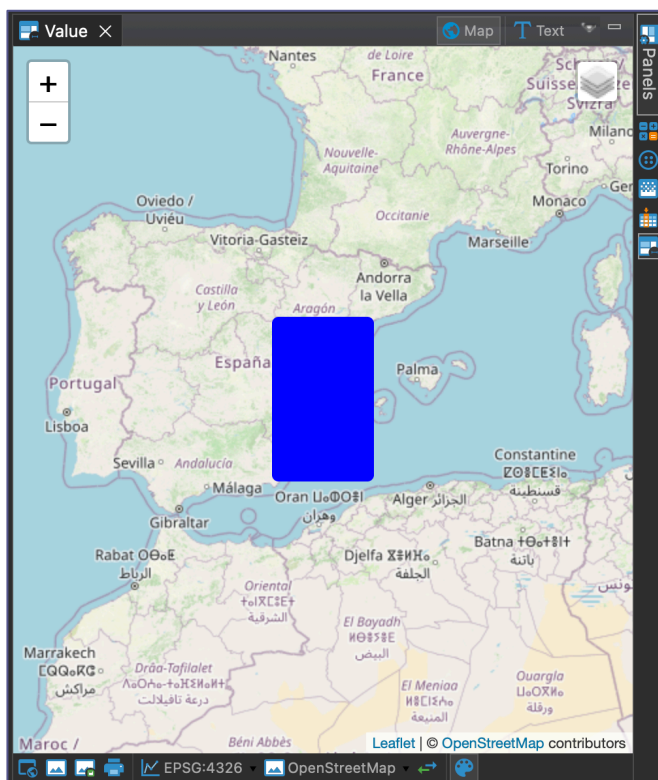


Figura 32. Representación visual de los puntos geográficos con la información sobre el riesgo de incendios de la tabla “*datos_riesgo_cv*”.

6.1.1.3. Riesgo de Incendios por Municipio y Día

A continuación, se quiere organizar y filtrar los datos de la tabla “*datos_riesgo_cv*” para facilitar la representación gráfica.

Mediante una serie de transformaciones, se crea una vista materializada, llamada “*cv_riesgo_maximo_ultimo_dia_geom_intersectada*”, que contiene el riesgo máximo de cada punto geográfico durante el último día, del que se disponga de información. A través de la vista materializada se garantiza la actualización de la información si hay una nueva entrada de datos o actualización de los datos originales. Por último, a través de operaciones espaciales de intersección de *PostGIS*, se transforman los puntos

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

geográficos de la vista en una malla de geometrías (ver figura 34) para facilitar la visualización de los datos.

Los atributos más importantes son:

- **geom.** Malla geométrica creada a partir de los puntos de riesgo.
- **prediction_date.** Fecha de la predicción de riego.
- **lat.** Latitud del punto de riesgo.
- **lon.** Longitud del punto de riesgo.
- **riesgo_maximo.** Valor del riesgo máximo.

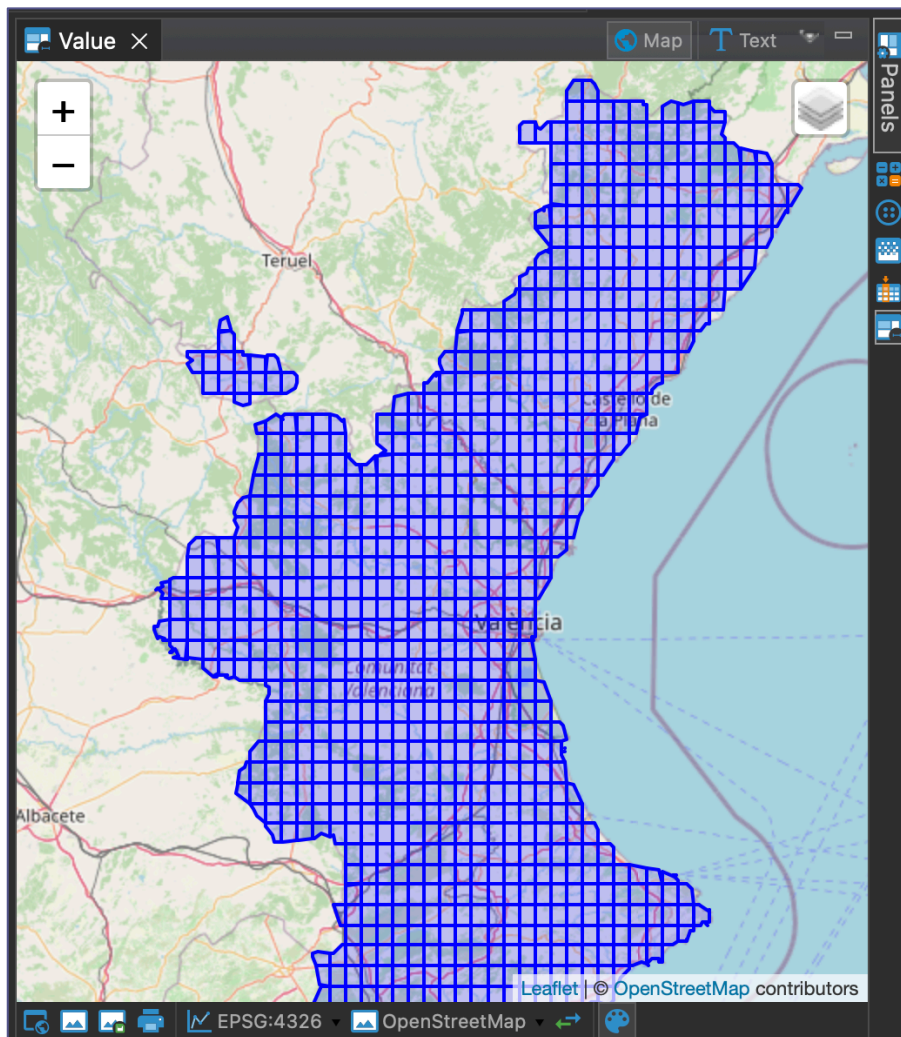


Figura 33. Malla de riesgos de incendio, contenida en la vista materializada, formada por geometrías intersectadas de las tablas "municipios_cv" y "datos_riesgo_cv".

6.1.2. Las Capas

Ahora se procede a la creación de las capas en GeoServer.

Primero, desde la sección "Datos" (ver figura 35), se configura un nuevo espacio de trabajo llamado "WORKSPACE_tfg_CV" (ver figura 36). Luego, se crea un nuevo almacén de datos vectoriales, llamado "DB_tfg_SCHEMA_com_valenciana" (ver figura

37), siendo la fuente de datos el esquema "com_valenciana" de la base de datos "tfg" del servidor PostGIS.

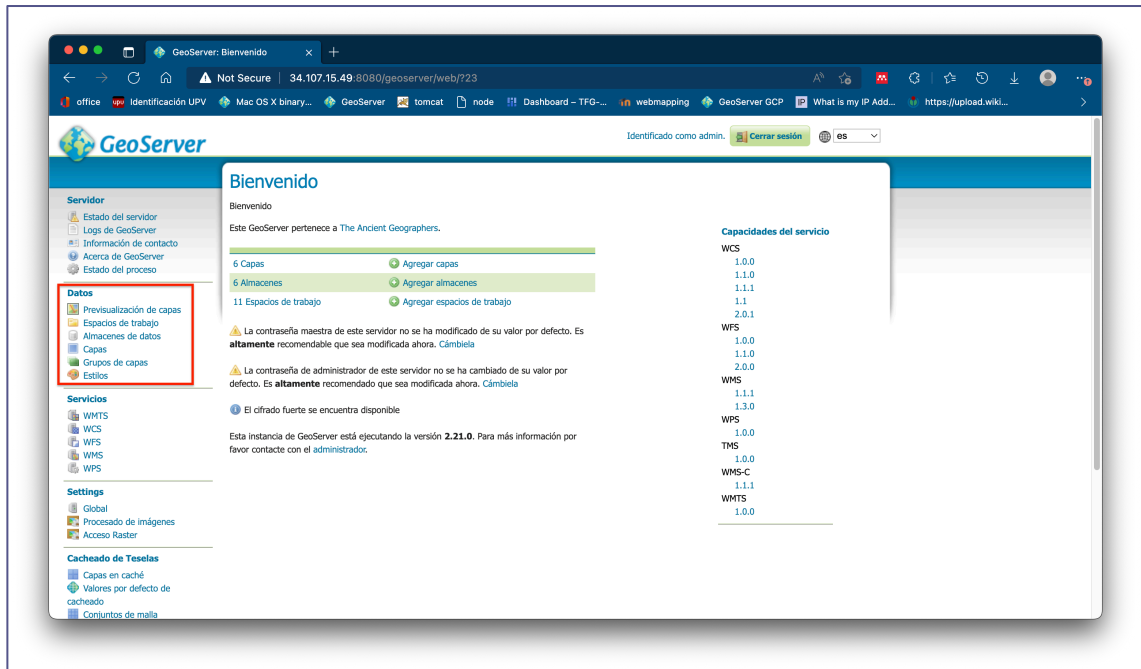


Figura 34. Ubicación de la sección "Datos".

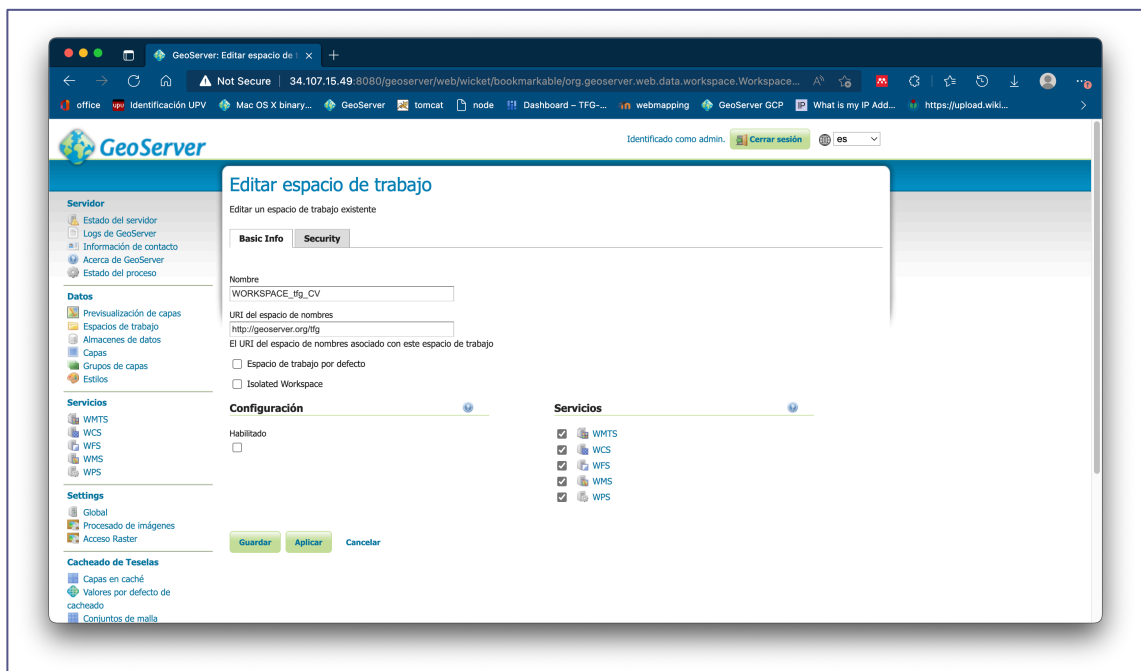
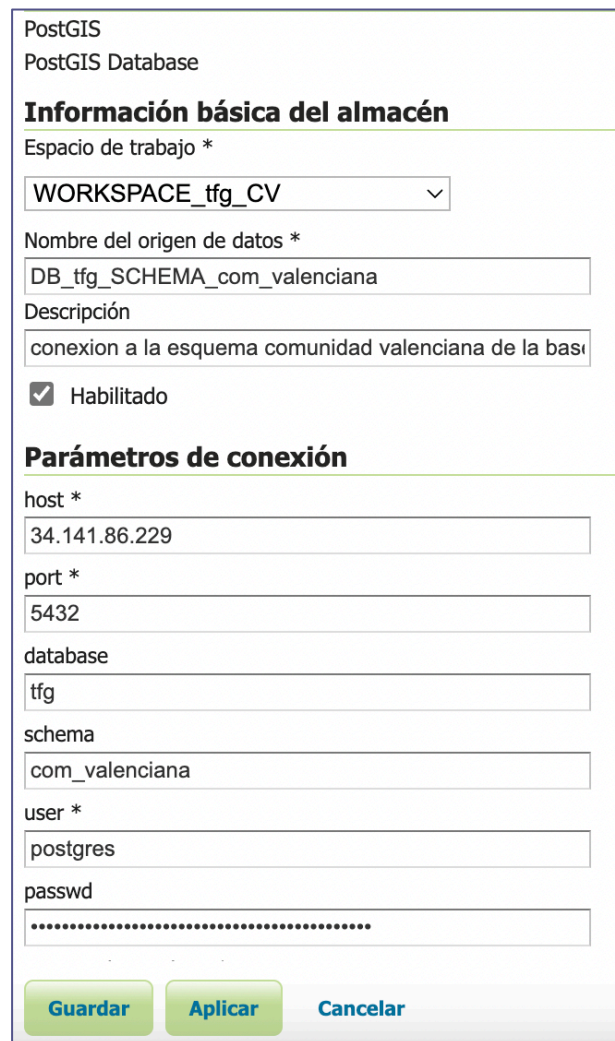


Figura 35. Espacio de trabajo "WORKSPACE_tfg_CV".

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.



The image shows a web-based configuration form for a PostGIS database. The form is titled "PostGIS Database" and is divided into two main sections: "Información básica del almacén" and "Parámetros de conexión".

Información básica del almacén

- Espacio de trabajo ***: A dropdown menu with the selected value "WORKSPACE_tfg_CV".
- Nombre del origen de datos ***: A text input field containing "DB_tfg_SCHEMA_com_valenciana".
- Descripción**: A text input field containing "conexion a la esquema comunidad valenciana de la basi".
- Habilitado**: A checked checkbox.

Parámetros de conexión

- host ***: A text input field containing "34.141.86.229".
- port ***: A text input field containing "5432".
- database**: A text input field containing "tfg".
- schema**: A text input field containing "com_valenciana".
- user ***: A text input field containing "postgres".
- passwd**: A password input field with masked characters.

At the bottom of the form, there are three buttons: "Guardar" (Save), "Aplicar" (Apply), and "Cancelar" (Cancel).

Figura 36. Configuración del almacén de datos "DB_tfg_SCHEMA_com_valenciana".

6.1.2.1. Capa de los Municipios

Primero, se va a crear una capa base con los municipios de la Comunidad Valenciana y datos con información geográfica como el nombre o el código postal de cada municipio.

Dentro de la sección "Datos", en el apartado "Capas" se configura una nueva capa llamada "municipios_cv". Se indica que el SRS es "EPSG:4326" y se deja que el GeoServer calcule los encuadres de la capa de forma automática. Se puede visualizar la capa "municipios_cv" en la figura 38. Al ser una capa de datos vectorial renderizada mediante el pre-visualizador de OpenLayers integrado en el GeoServer, también se pueden consultar los atributos de cada municipio si se seleccionan puntos de la superficie.

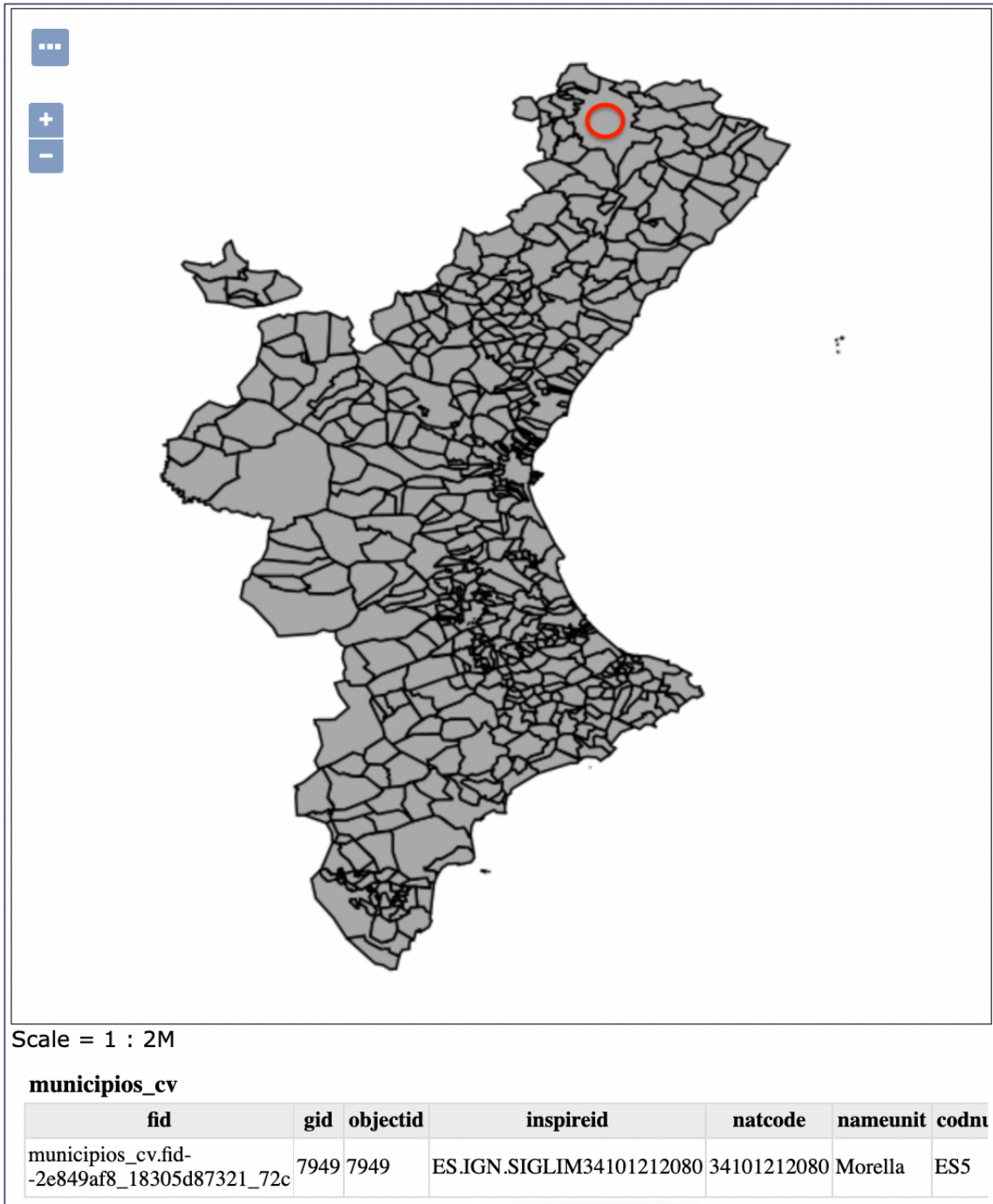


Figura 37. Capa "municipios_cv" con atributos del municipio Morella.

6.1.2.2. Capa de Puntos de Riesgos

Al igual que en la subsección anterior, se procede con la creación de una nueva capa desde el apartado "Capas" en el menú "Datos" del GeoServer. Desde el almacén de datos, se selecciona la vista materializada "cv_riesgo_maximo_ultimo_dia_geom_intersectada" como origen de datos de esta nueva capa. Se establecen el SRS y los encuadres. La nueva capa, cuyo nombre es

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

igual al de la vista materializada, “*cv_riesgo_maximo_ultimo_dia_geom_intersectada*”, es observable en la figura 39.

Dispone de las mismas características que la capa “*municipios_cv*”, que permite aproximar o alejar la vista y consultar atributos de las geometrías.

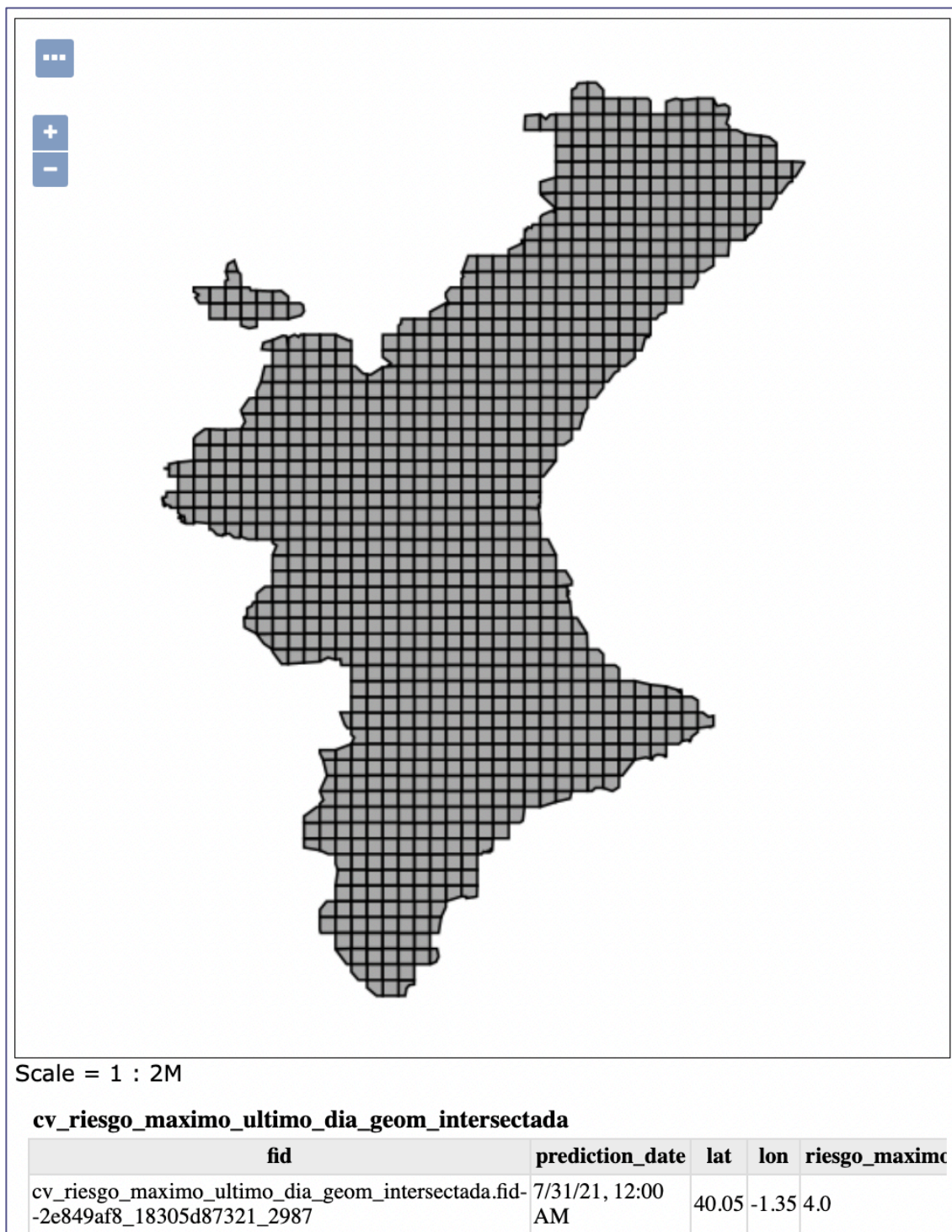


Figura 38. Capa “*cv_riesgo_maximo_ultimo_dia_geom_intersectada*” con una malla de geometrías e información sobre riesgos de incendio máximos, del último día, de la Comunidad Valenciana.

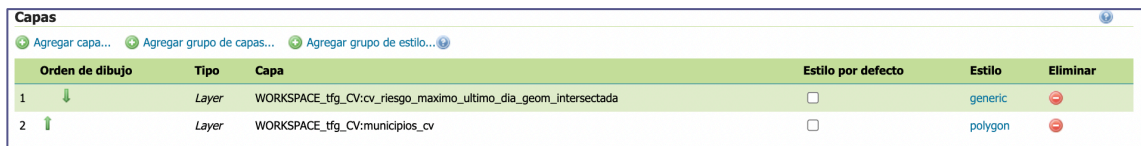
6.1.3. El Mapa

Una vez las capas han sido creadas, se van a combinar, se les va a aplicar estilos y por último publicar en forma de mapa. Esto se puede hacer desde el GeoServer a través de una estructura llamada “Grupo de Capas”.

6.1.3.1. Grupo de Capas

Desde el menú “Datos”, dentro del apartado llamado “Grupos de Capas” se va a crear una agrupación de capas:

“INCENDIOS_max_riesgo_dia_cv_geom_intersectadas”, este grupo está compuesto por la capa “cv_riesgo_maximo_ultimo_dia_geom_intersectada” superpuesta por la capa “municipios_cv”, observable en la figura 40. En la figura 41 se pueden observar atributos de las dos capas devueltos por el servidor al seleccionar un punto en el mapa.



Orden de dibujo	Tipo	Capa	Estilo por defecto	Estilo	Eliminar
1	Layer	WORKSPACE_tfg_CV:cv_riesgo_maximo_ultimo_dia_geom_intersectada	<input type="checkbox"/>	generic	
2	Layer	WORKSPACE_tfg_CV:municipios_cv	<input type="checkbox"/>	polygon	

Figura 39. Orden de dibujo de las capas del grupo “INCENDIOS_max_riesgo_dia_cv_geom_intersectadas”.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

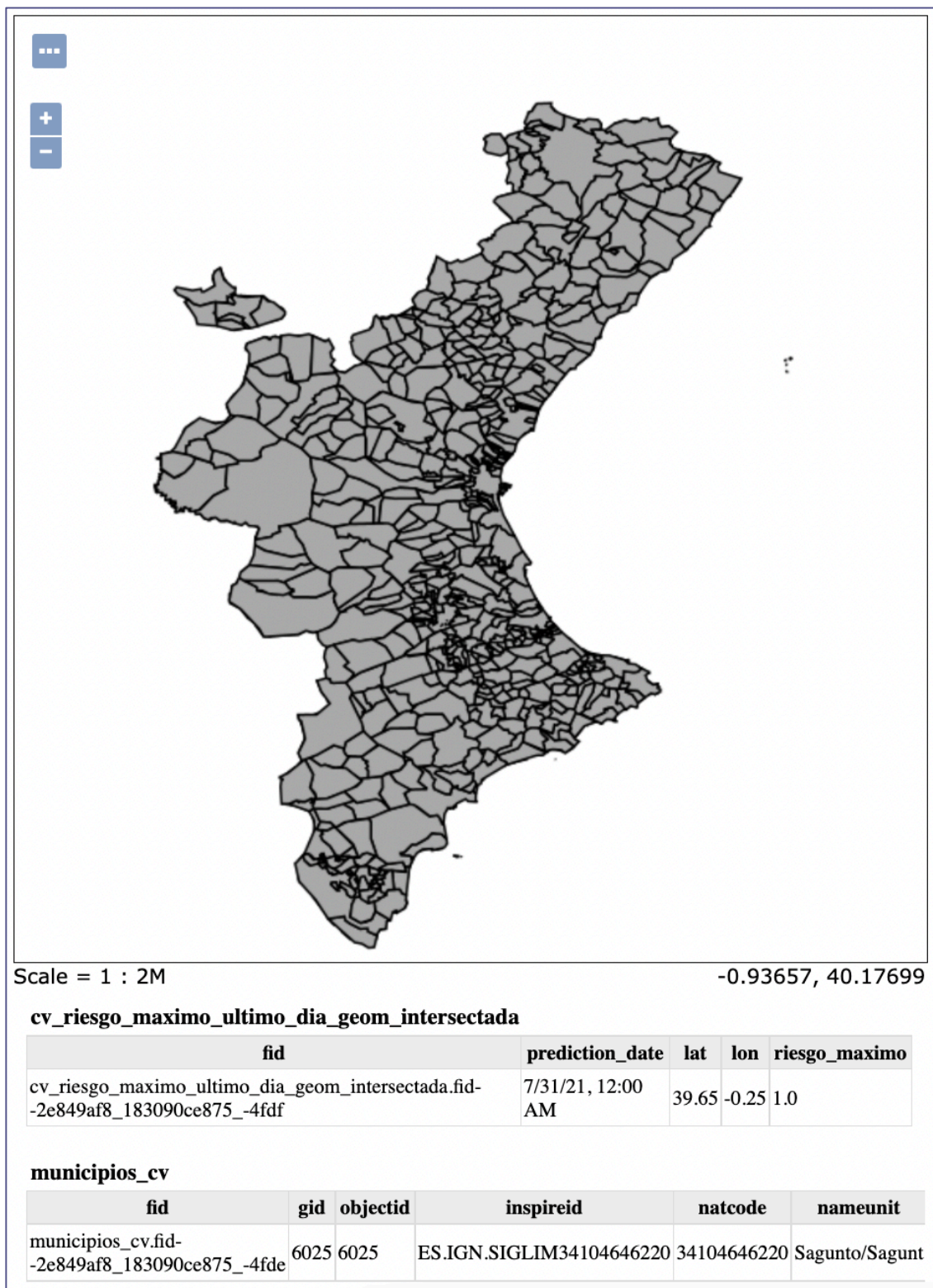


Figura 40. Atributos devueltos por el grupo de capas "INCENDIOS_max_riesgo_dia_cv_geom_intersectadas".

6.1.3.2. Estilos

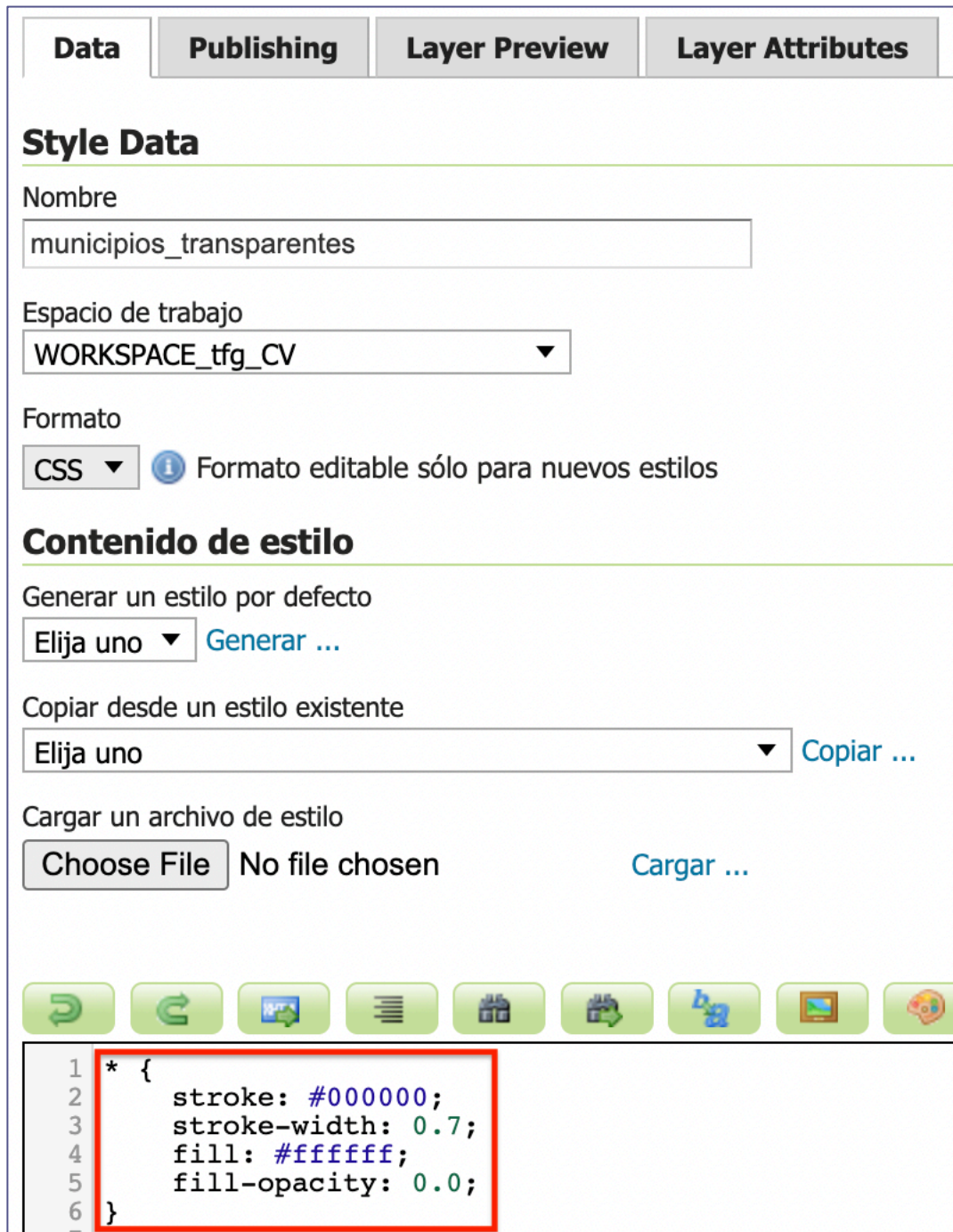
Si bien el grupo de capas “*INCENDIOS_max_riesgo_dia_cv_geom_intersectadas*” ya cumple el objetivo de interactividad perseguido, el resultado no es satisfactorio. A simple vista no hay ningún indicio visual del riesgo de incendios. El usuario tiene que seleccionar municipios de uno en uno, para averiguar el riesgo de forma individual. Este es un problema que se intenta solucionar mediante la creación de estilos en el *GeoServer*.

Se van a crear dos nuevos grupos de capas, debidamente estilizados, para mostrar lo que se puede conseguir mediante la combinación de capas y estilos.

6.1.3.2.1. Capa con Transparencia y Malla de Colores

Haciendo uso de la extensión de estilos CSS, instalada durante el desarrollo de la solución (ver apartado 5.3), se va a crear un nuevo estilo llamado “*municipios_transparentes*” (ver figura 42).

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.



Data **Publishing** **Layer Preview** **Layer Attributes**

Style Data

Nombre
municipios_transparentes

Espacio de trabajo
WORKSPACE_tfg_CV

Formato
CSS *Formato editable sólo para nuevos estilos*

Contenido de estilo

Generar un estilo por defecto
Elija uno [Generar ...](#)

Copiar desde un estilo existente
Elija uno [Copiar ...](#)

Cargar un archivo de estilo
Choose File No file chosen [Cargar ...](#)

```
1 * {  
2   stroke: #000000;  
3   stroke-width: 0.7;  
4   fill: #ffffff;  
5   fill-opacity: 0.0;  
6 }
```

Figura 41. Definición y código CSS del estilo “municipios_transparentes”.

Se va a aplicar este estilo a la capa “municipios_cv” para dar un efecto de transparencia y permitir observar capas de una jerarquía inferior dentro de un grupo de capas.

Se crea un segundo estilo llamado “colores_riesgo_cv” (ver figura 43) que asigna un color a cada geometría de la capa “cv_riesgo_maximo_ultimo_dia_geom_intersectada” según el valor del riesgo de incendio máximo.

Data Publishing Layer Preview Layer Attributes

Style Data

Nombre

Espacio de trabajo

Formato
 Formato editable sólo para nuevos estilos

Contenido de estilo

Generar un estilo por defecto
 [Generar ...](#)

Copiar desde un estilo existente
 [Copiar ...](#)

Cargar un archivo de estilo
 No file chosen [Cargar ...](#)

```

1  * {
2      stroke: #ffffff;
3      stroke-width: 0.0;
4  }
5
6  /* @title minimo */
7  [riesgo_maximo = 1] {
8      fill: #bae4b3; }
9  /* @title pequeno */
10 [riesgo_maximo = 2] {
11     fill: #fecc5c; }
12 /* @title medio */
13 [riesgo_maximo = 3] {
14     fill: #fd8d3c; }
15 /* @title maximo */
16 [riesgo_maximo = 4] {
17     fill: #d7191c; }
18

```

Figura 42. Definición y código CSS del estilo "colores_riesgo_cv".

Una vez se tienen los estilos definidos se crea un nuevo grupo de capas llamado "INCENDIOS_max_riesgo_dia_cv_geom_intersectadas_colores" con las capas estilizadas. Se puede observar en la figura 44 el orden de dibujo de las capas.



Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

Orden de dibujo	Tipo	Capa	Estilo por defecto	Estilo	Eliminar
1	Layer	WORKSPACE_tfg_CV:cv_riesgo_maximo_ultimo_dia_geom_intersectada	<input type="checkbox"/>	colores_riesgo_cv	
2	Layer	WORKSPACE_tfg_CV:municipios_cv	<input type="checkbox"/>	municipios_transparentes	

Figura 43. Grupo de capas "INCENDIOS_max_riesgo_dia_cv_geom_intersectadas_colores" con el orden de dibujo y los estilos aplicados.

El resultado final se puede ver en la figura 45. El mapa ha adquirido una nueva característica visual la cual permite identificar rápidamente las parcelas de mayor riesgo de incendios. Al igual que en el mapa sin estilo, el usuario puede seleccionar zonas del mapa y consultar información sobre el municipio o valores de riesgo.

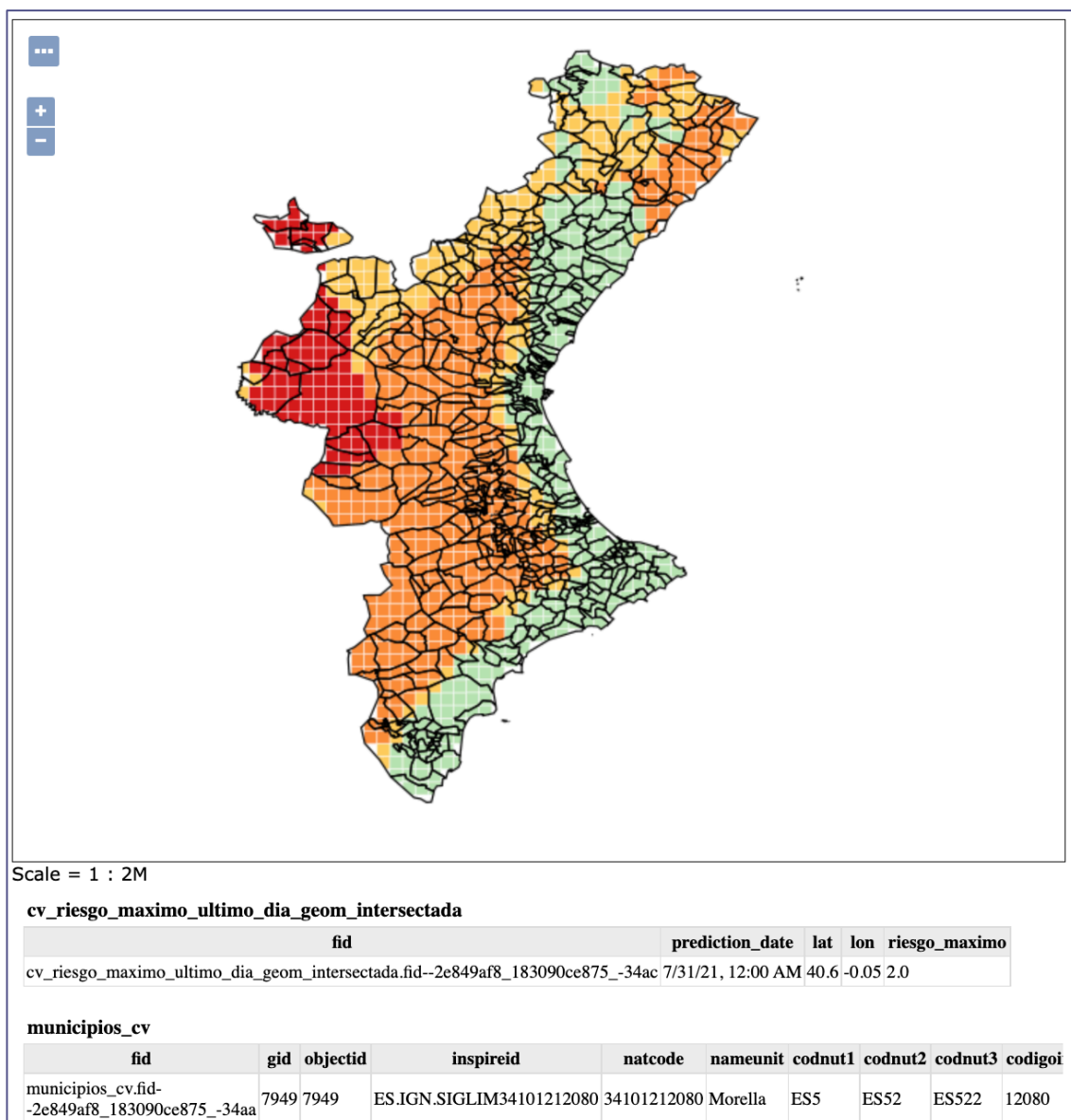


Figura 44. Mapa de la Comunidad Valenciana con el riesgo máximo de incendios representado por colores.

6.1.3.2.2. Capa con Transparencia y Mapa de Calor

A continuación, se propone un nuevo ejemplo de uso de estilos creando un “*mapa de calor*” de los riesgos. Un “*mapa de calor*” es una técnica de visualización que busca representar información en un mapa a través de la variación del tono y la intensidad de un color. Es una técnica muy utilizada en meteorología.

El primer paso es crear un nuevo estilo para la capa “*cv_riesgo_maximo_ultimo_dia_geom_intersectada*”. En esta ocasión, se usa el lenguaje de estilos nativo de GeoServer llamado SLD (ver figura 46) para definir este nuevo estilo, al que llamaremos “*SLD_heatmap*”.

```
9 <Name>Heatmap</Name>
10 <UserStyle>
11 <Title>Heatmap</Title>
12 <Abstract>Heatmap para los riesgos</Abstract>
13 <FeatureTypeStyle>
14 <Transformation>
15 <ogc:Function name="gs:Heatmap">
16 <ogc:Function name="parameter">
17 <ogc:Literal>data</ogc:Literal>
18 </ogc:Function>
19 <ogc:Function name="parameter">
20 <ogc:Literal>weightAttr</ogc:Literal>
21 <ogc:Literal>riesgo_maximo</ogc:Literal>
22 </ogc:Function>
23 <ogc:Function name="parameter">
24 <ogc:Literal>radiusPixels</ogc:Literal>
25 <ogc:Function name="env">
26 <ogc:Literal>radius</ogc:Literal>
27 <ogc:Literal>50</ogc:Literal>
28 </ogc:Function>
29 </ogc:Function>
30 <ogc:Function name="parameter">
31 <ogc:Literal>pixelsPerCell</ogc:Literal>
32 <ogc:Literal>2</ogc:Literal>
33 </ogc:Function>
34 <ogc:Function name="parameter">
35 <ogc:Literal>outputBBOX</ogc:Literal>
36 <ogc:Function name="env">
37 <ogc:Literal>wms_bbox</ogc:Literal>
38 </ogc:Function>
39 </ogc:Function>
40 <ogc:Function name="parameter">
41 <ogc:Literal>outputWidth</ogc:Literal>
42 <ogc:Function name="env">
43 <ogc:Literal>wms_width</ogc:Literal>
44 </ogc:Function>
45 </ogc:Function>
46 <ogc:Function name="parameter">
47 <ogc:Literal>outputHeight</ogc:Literal>
48 <ogc:Function name="env">
49 <ogc:Literal>wms_height</ogc:Literal>
50 </ogc:Function>
51 </ogc:Function>
52 </ogc:Function>
53 </Transformation>
54 <Rule>
55 <RasterSymbolizer>
56 <!-- specify geometry attribute of input to pass validation -->
57 <Geometry><ogc:PropertyName>riesgo_maximo</ogc:PropertyName></Geometry>
58 <Opacity>0.7</Opacity>
59 <ColorMap type="ramp" >
60 <ColorMapEntry color="#ffffff" quantity="0.25" label="sin riesgo"/>
61 <ColorMapEntry color="#fccc5c" quantity="0.5" label="bajo"/>
```

Figura 45. Código SLD del estilo “*SLD_heatmap*”.

Se procede a la creación de un nuevo grupo de capas llamado “*INCENDIOS_max_riesgo_dia_cv_geom_intersectadas_heatmap*”. Se agrupan la capa “*municipios_cv*” con la capa “*cv_riesgo_maximo_ultimo_dia_geom_intersectada*” y a esta última se le aplica el nuevo estilo “*SLD_heatmap*”. El resultado, observable en la figura 44, es un mapa de calor interactivo de los riesgos de incendio, máximos, del último día en la Comunidad Valenciana.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

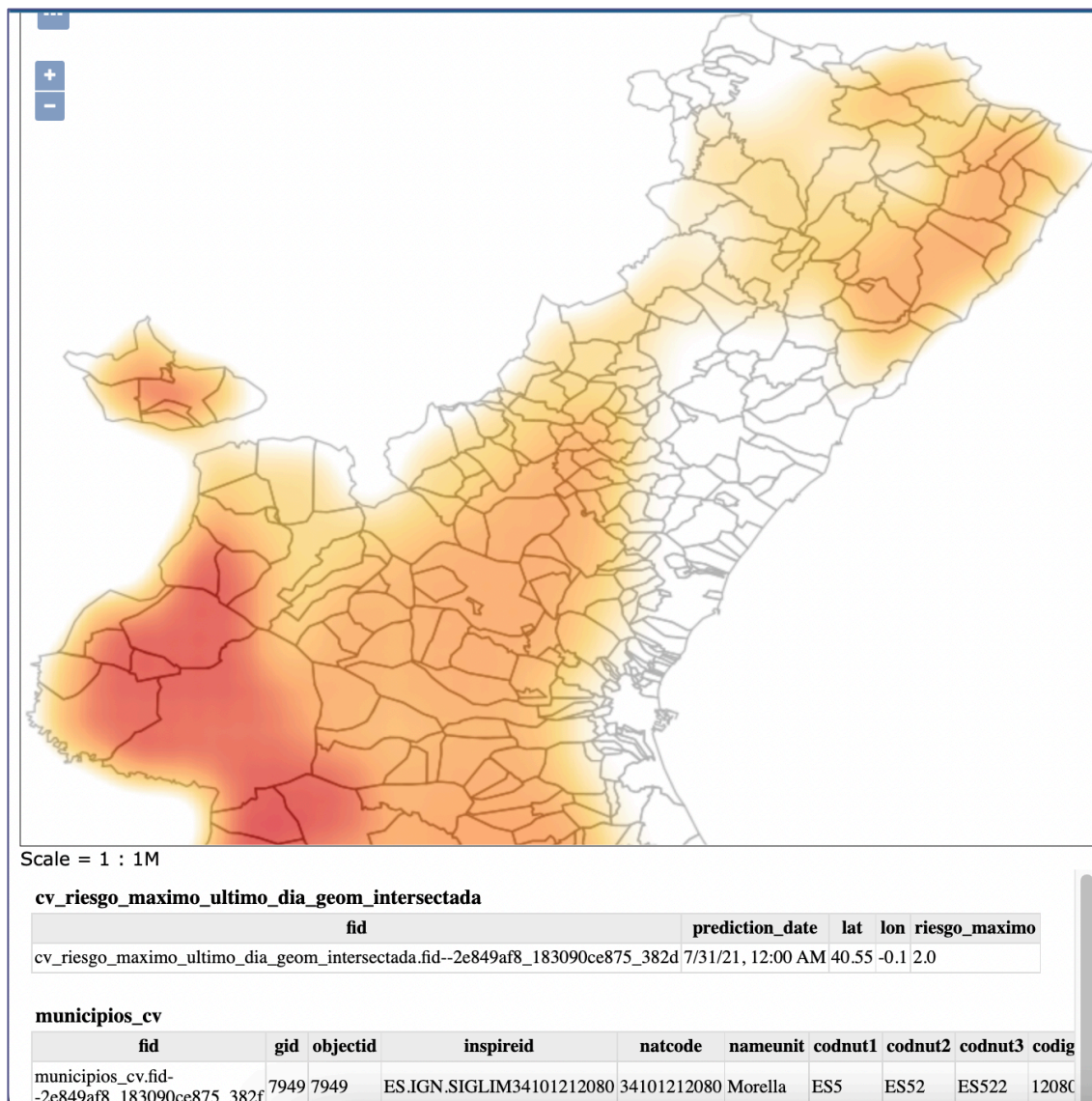


Figura 46. Grupo de capas "INCENDIOS_max_riesgo_dia_cv_geom_intersectadas_heatmap" con estilo "SLD_heatmap".

6.1.3.3. Habilitar Servicios para Clientes

Finalmente, se quiere habilitar los mapas para ser accesibles en la nube. Esto se puede conseguir a través de los servicios geospaciales soportados por GeoServer.

Dentro del menú "Datos", en "Espacios de Trabajo" se edita el espacio de trabajo "WORKSPACE_tfg_CV" (ver figura 48) para habilitar los servicios WMTS, WCS, WFS, WMS y WPS. Ahora, los mapas están disponible en la web a través de estos servicios.

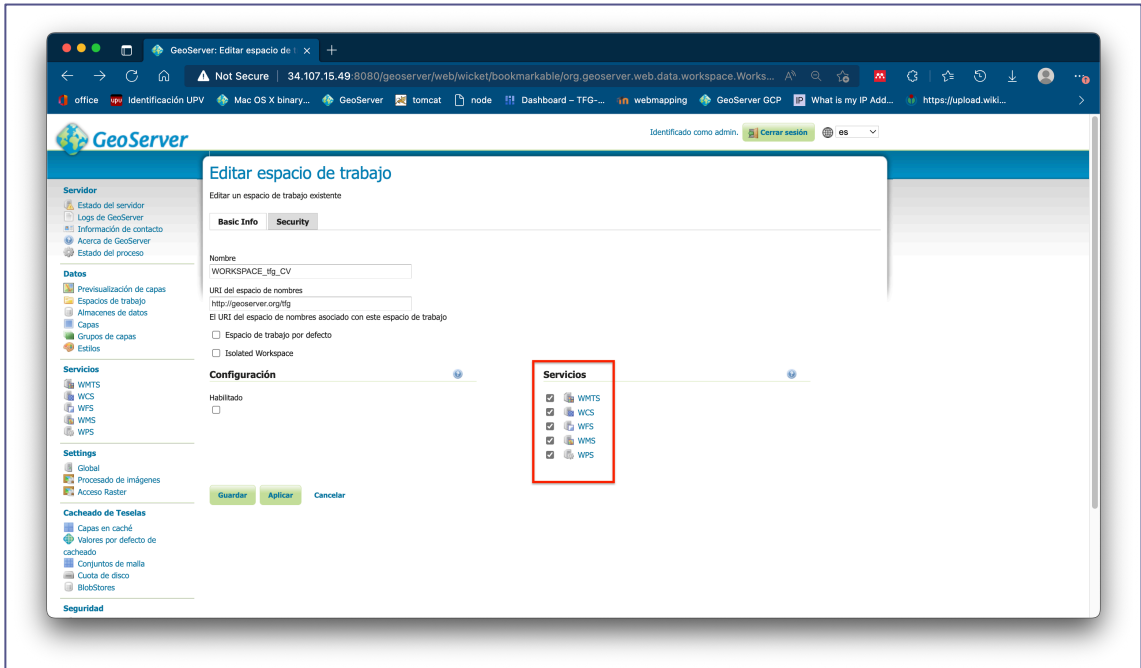


Figura 47. Habilitar servicios web de difusión de mapas del GeoServer para el espacio de trabajo "WORKSPACE_tfg_CV".

Para comprobar su correcto funcionamiento, se va a usar QGIS, un software SIG de escritorio que permite conectarse a fuentes de datos geoespaciales. Se crea un nuevo proyecto de QGIS y se añade, en forma de capa, el mapa que creado encima de una segunda capa procedente de un servicio de OpenStreetMap. Se puede observar el resultado en la figura 49 y validar la correcta publicación del mapa.

Es de interés observar el perfecto alineamiento gracias a la definición adecuada del SRS de nuestro mapa. También se puede observar en el panel derecho como GeoServer transmite correctamente la información asociada al mapa.

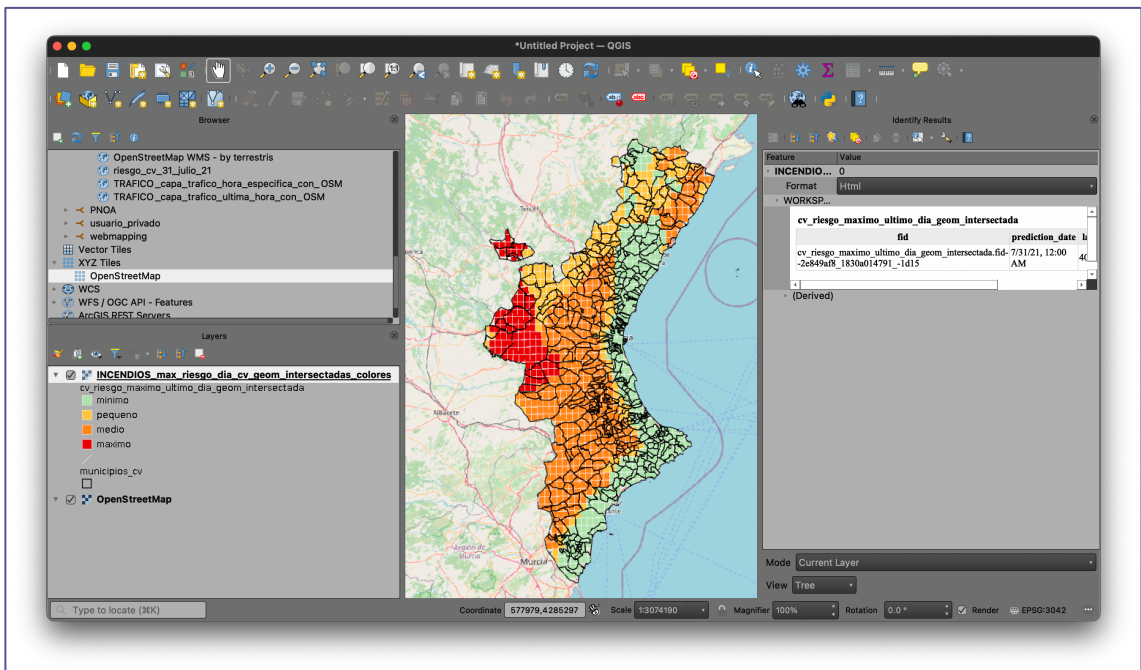


Figura 48. Proyecto en QGIS que usa la capa "cv_riesgo_maximo_ultimo_dia_geom_intersectada".



Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

6.1.4. Comentarios

Durante esta prueba se han demostrado las capacidades fundamentales de la infraestructura *SIG* para almacenar y exponer datos geoespaciales al público general.

Se ha visto cómo almacenar y manipular datos geoespaciales en la base de datos que luego pueden ser usados como fuente de datos para el servidor geoespacial. A través del servidor, se han podido crear capas, estilos y grupos de capas que han sido publicados en la nube a través de servicios *WEB*. El servidor usado, *GeoServer*, cumple con los estándares del sector, lo que ha permitido combinar de manera sencilla capas propias con capas de servicios ajenos y mostrarlos en un visor *SIG*, como *QGIS*.

6.2. Tráfico en la localidad de Valencia

Para esta segunda prueba, se quiere desarrollar y publicar un mapa que muestre el tráfico de Valencia. En concreto, se quiere mostrar el tráfico de la última hora de la que se tiene información. El mapa estará compuesto por 2 capas:

- Una capa ráster proporcionada por un servicio de mapas.
- Una capa vectorial con geometrías e información de tránsito sobre las vías de Valencia.

6.2.1. Los Datos

Se proporcionan dos conjuntos de datos. Un primer conjunto que contiene información sobre el nombre, el sentido y las coordenadas de las vías. El segundo conjunto contiene información de tránsito asociada a estas vías. El origen de la información de este segundo conjunto son unas espiras magnéticas, situadas debajo de las calles de Valencia, que monitorizan el caudal de tráfico.

El primer conjunto de datos es un fichero CSV llamado "*definición_tramos.csv*". Contiene la siguiente información:

- Un identificador único para cada tramo denominado "*IdTA*".
- El nombre del tramo.
- Las coordenadas de origen del tramo.
- Las coordenadas destino del tramo.

El segundo conjunto de datos es otro fichero CSV llamado "*informacion_trafico_por_tramos.csv*". Este segundo fichero contiene:

- El identificador "*IdTA*" para emparejar la información con los tramos.
- La fecha y la hora de creación de la entrada.
- La intensidad, ocupación y velocidad en un tramo.
- Fiabilidad probabilística de la intensidad, ocupación y velocidad.

6.2.2. Carga y Limpieza de datos

Como en la prueba anterior, se va a crear un nuevo esquema para organizar nuestra información. El nombre de este esquema será "*trafico_valencia*".

Dentro de este esquema se crean dos tablas para almacenar la información proporcionada: "*tramos_geom_info*" (ver figura 50), para almacenar información del fichero "*definición_tramos.csv*", y "*primitiva_info_trafico*" (ver figura 51), para almacenar información proporcionada por el fichero "*informacion_trafico_por_tramos.csv*".

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

The screenshot shows the PostgreSQL Properties window for the table 'tramos_geom_info'. The table is located in the 'pg_default' tablespace, owned by 'postgres', and has Object ID 37459. The table structure is as follows:

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
geom_tramo	1	geometry			[]		
idta	2	int4			[]		
origen_id	3	int8			[]		
destino_id	4	int8			[]		
coeficiente	5	int4			[]		
nombre	6	varchar(50)		default	[]		
descripcion	7	varchar(128)		default	[]		
sentido	8	varchar(50)		default	[]		

Figura 49. Estructura de la tabla "tramos_geom_info".

The screenshot shows the PostgreSQL Properties window for the table 'primitiva_info_trafico'. The table is located in the 'pg_default' tablespace, owned by 'postgres', and has Object ID 37391. The table structure is as follows:

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
fecha_por_corregir	2	timestamptz			[]		
hora	3	timetz			[]		
intensidad	4	float8			[]		
ocupacion	6	float8			[]		
velocidad	8	float8			[]		
idta	10	int4			[]		
fiabilidad_intensidad	11	int4			[]		
fiabilidad_ocupacion	12	int4			[]		
fiabilidad_velocidad	13	int4			[]		
fecha	14	date			[]		

Figura 50. Estructura de la tabla "primitiva_info_trafico".

La tabla "tramos_geom_info" ha sido expandida, añadiendo el atributo "geom_tramo", para contener las geometrías correspondientes a las calles de Valencia, generadas mediante transformaciones geoespaciales. Para lograrlo se han utilizado las coordenadas de inicio y final de los tramos.

Finalmente, haciendo uso de las dos tablas creadas, se crea una vista materializada llamada "capa_geom_tramos_con_info_trafico_ultimo_dia_hora_mv" (ver figura 52), para contener las geometrías de las vías y la información de tránsito asociada a estas. Se filtran los datos para mostrar solo los de la última hora del último día. Esta vista será utilizada como fuente de datos en GeoServer.

```

-- trafico_valencia.capa_geom_tramos_con_info_trafico_ultimo_dia_hora_mv source
CREATE MATERIALIZED VIEW trafico_valencia.capa_geom_tramos_con_info_trafico_ultimo_dia_hora_mv
TABLESPACE pg_default
AS SELECT tgi.geom_tramo,
tgi.idta,
tgi.origen_id,
tgi.destino_id,
tgi.coeficiente,
tgi.nombre,
tgi.descripcion,
tgi.sentido,
pit.fecha + pit.hora AS fecha_hora,
pit.intensidad,
pit.fiabilidad_intensidad,
pit.ocupacion,
pit.fiabilidad_ocupacion,
pit.velocidad,
pit.fiabilidad_velocidad
FROM trafico_valencia.tramos_geom_info tgi
LEFT JOIN trafico_valencia.primitiva_info_trafico pit ON tgi.idta = pit.idta
WHERE (pit.fecha + pit.hora IN ( SELECT max(pit2.fecha + pit2.hora) AS max
FROM trafico_valencia.primitiva_info_trafico pit2))
WITH DATA;

```

Figura 51 Código SQL para la creación de la vista materializada "capa_geom_tramos_con_info_trafico_ultimo_dia_hora_mv".

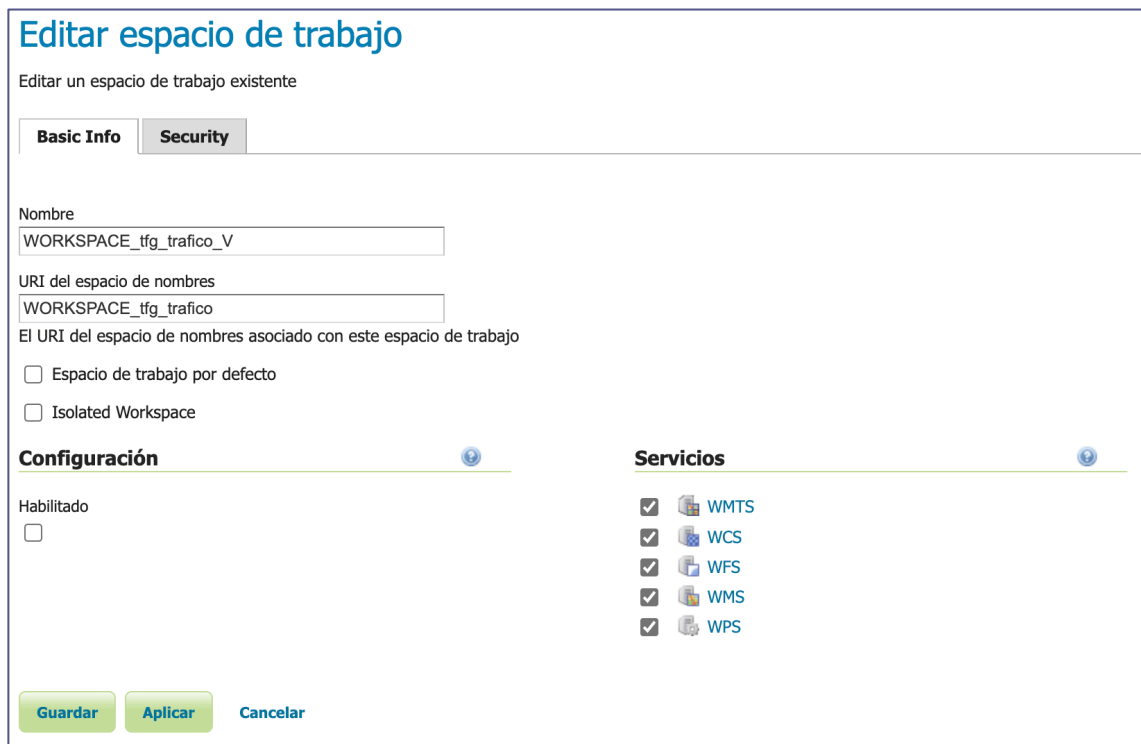
Cabe mencionar que si bien el proceso de limpieza y carga de datos ha sido manual, las estructuras en la base de datos están preparadas para una carga automatizada con, por ejemplo, un proceso *ETL* (extraer, transformar y cargar).

6.2.3. Las Capas

Con los datos debidamente filtrados y estructurados, se pueden crear las capas en *GeoServer*.

Para una mejor organización, se crea un nuevo espacio de trabajo llamado "*WORKSPACE_tfg_trafico_V*" (ver figura 53).

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.



Editar espacio de trabajo

Editar un espacio de trabajo existente

Basic Info **Security**

Nombre
WORKSPACE_tfg_trafico_V

URI del espacio de nombres
WORKSPACE_tfg_trafico

El URI del espacio de nombres asociado con este espacio de trabajo

Espacio de trabajo por defecto

Isolated Workspace

Configuración

Habilitado

Servicios

- WMTS
- WCS
- WFS
- WMS
- WPS

Guardar **Aplicar** **Cancelar**

Figura 52. Espacio de trabajo "WORKSPACE_tfg_trafico_V" y sus características.

6.2.3.1. Capa con Tramos e Información de Transito

Para empezar, se configura una nueva fuente de datos llamada "DB_tfg_SCHEMA_trafico_valencia" para leer del esquema "trafico_valencia" (ver figura 54) de la base de datos.

Editar un origen de datos vectoriales

Editar un origen de datos vectorial existente

PostGIS
PostGIS Database

Información básica del almacén

Espacio de trabajo *

WORKSPACE_tfg_trafico_V

Nombre del origen de datos *

DB_tfg_SCHEMA_trafico_valencia

Descripción

Habilitado

Parámetros de conexión

host *

34.141.86.229

port *

5432

database

tfg

schema

trafico_valencia

user *

postgres

passwd

.....

Figura 53. Configuración fuente de datos "DB_tfg_SCHEMA_trafico_valencia".

Utilizando la vista "capa_geom_tramos_con_info_trafico_ultimo_dia_hora_mv" como fuente, se crea una nueva capa en el GeoServer denominada "tramos_con_info_trafico_ultimo_dia_hora" (ver figura 55). Se puede observar en la figura 56 las geometrías de la capa, que fueron generadas durante el subapartado 6.2.2. También están disponibles las funciones de consulta de atributos, observables también en la figura 56.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

Editar capa

Editar los datos de la capa y la información de publicación

WORKSPACE_tfg_ trafico_V:tramos_con_info_ trafico_ ultimo_dia_hora

Configure el recurso y la información de publicación para esta capa

Datos | **Publicación** | **Dimensiones** | **Cacheado de Teselas** | **Seguridad**

Editar capa

Información básica del recurso

Nombre del almacén: DB_tfg_SCHEMA_ trafico_valencia

Native Name: capa_geom_tramos_con_info_ trafico_ ultimo_dia_hora_mv

Nombre

Habilitado
 Anunciado

Título i18n

Resumen i18n

Figura 54. Información de la capa "tramos_con_info_ trafico_ ultimo_dia_hora".

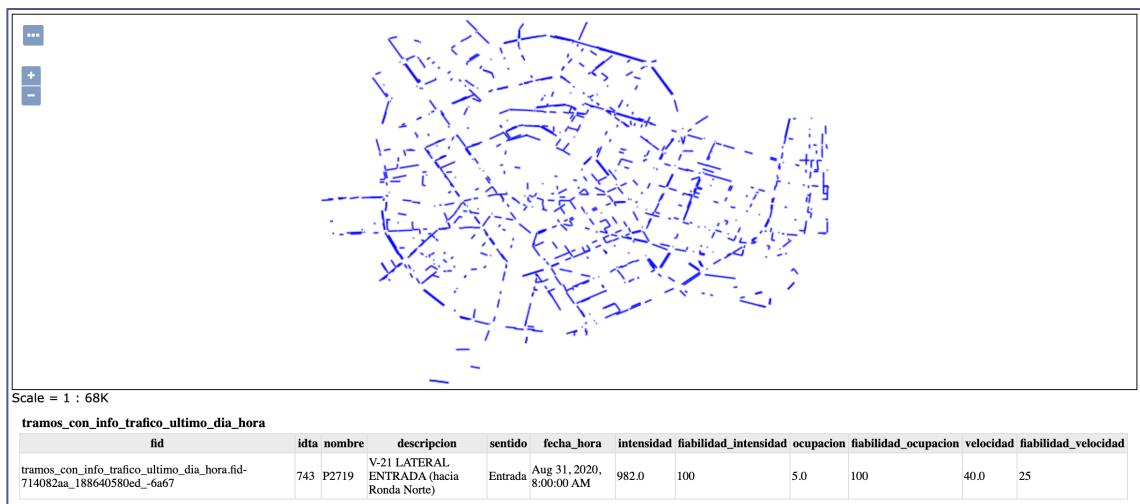
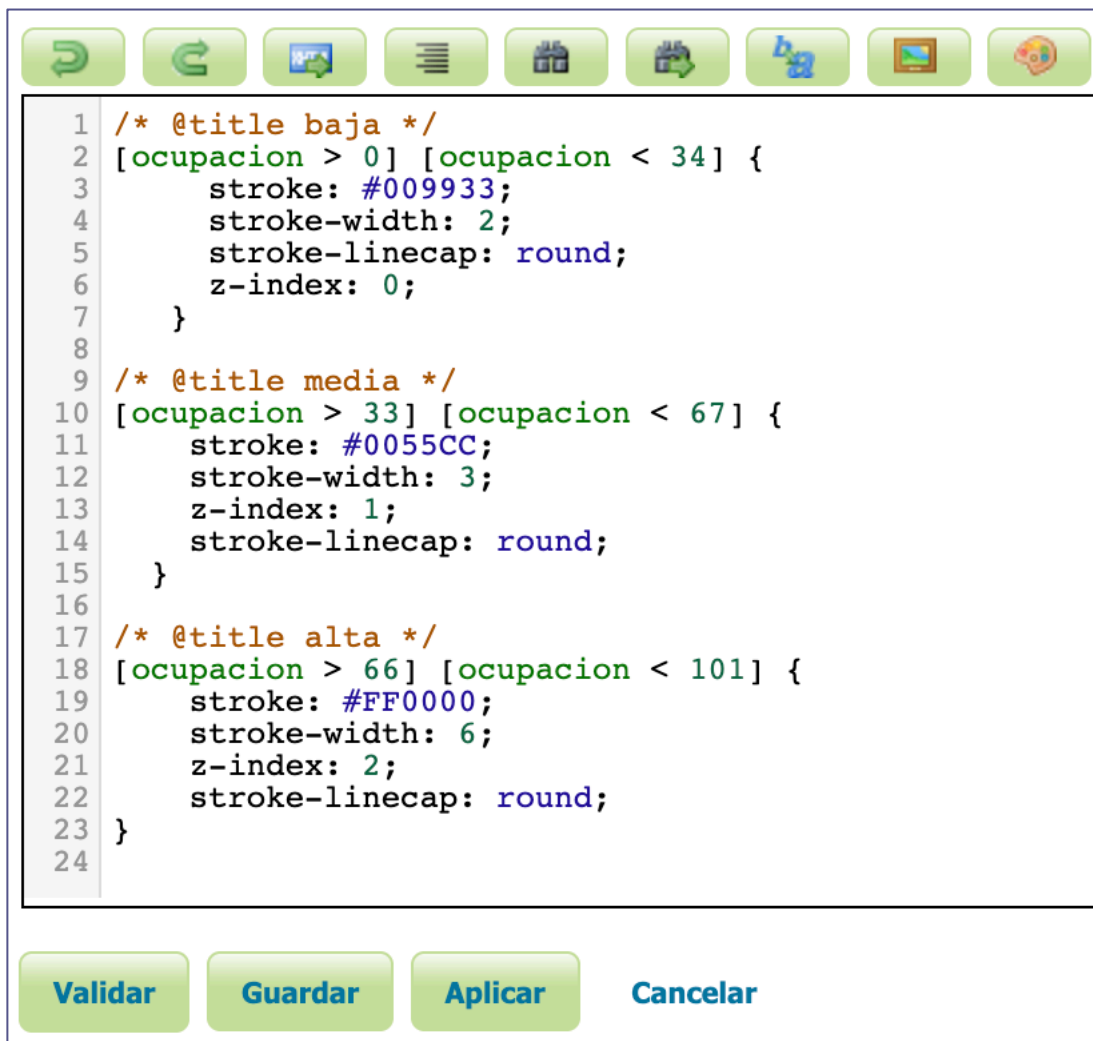


Figura 55. Visualización de la capa "tramos_con_info_ trafico_ ultimo_dia_hora".

Ahora, se quiere estilizar los tramos para representar su ocupación de forma visual. Como en la prueba anterior, esto se puede hacer desde el GeoServer creando un estilo.

En el apartado "Estilos" en la sección "Datos", se crea un nuevo estilo llamado "estilos_ trafico" y se asigna al espacio de trabajo "WORKSPACE_tfg_ trafico_V". A través de CSS y el atributo "ocupacion" se crea el código observable en la figura 57.



```
1 /* @title baja */
2 [ocupacion > 0] [ocupacion < 34] {
3     stroke: #009933;
4     stroke-width: 2;
5     stroke-linecap: round;
6     z-index: 0;
7 }
8
9 /* @title media */
10 [ocupacion > 33] [ocupacion < 67] {
11     stroke: #0055CC;
12     stroke-width: 3;
13     z-index: 1;
14     stroke-linecap: round;
15 }
16
17 /* @title alta */
18 [ocupacion > 66] [ocupacion < 101] {
19     stroke: #FF0000;
20     stroke-width: 6;
21     z-index: 2;
22     stroke-linecap: round;
23 }
24
```

Validar Guardar Aplicar Cancelar

Figura 56. Código CSS para el estilo "estilos_ trafico".

Este código permite alterar el color y tamaño de las geometrías de los tramos. En la figura 58 se puede observar el resultado de aplicar este estilo a la capa "tramos_con_info_ trafico_ ultimo_dia_hora".

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

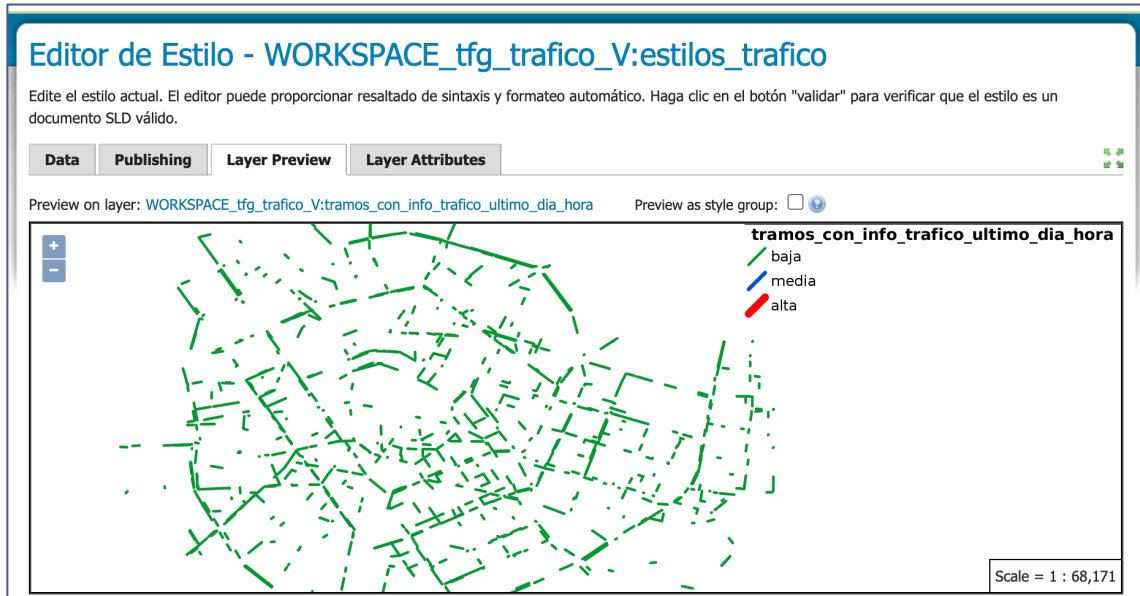


Figura 57. Capa "tramos_con_info_trafico_ultimo_dia_hora" con el estilo "estilos_trafico" aplicado.

Como en esta capa la ocupación de todos los tramos está dentro del umbral "bajo", no se pueden apreciar los distintos colores y grosores modificados por el estilo. Se ha decidido preparar otra capa llamada "tramos_con_info_trafico_dia_y_hora_especifico" como ejemplo para mostrar la aplicación del estilo a tramos con distintos niveles de ocupación. Es una capa parecida a la anterior, con la única diferencia siendo el origen de los datos seleccionados, correspondientes a una hora elegida de forma manual, donde la ocupación de los tramos es variada. Se puede observar de esta manera, en la figura 59, la representación de diversos tramos con una ocupación "media" y "alta". El estilo modifica de forma adecuada el grosor y el color de los tramos.

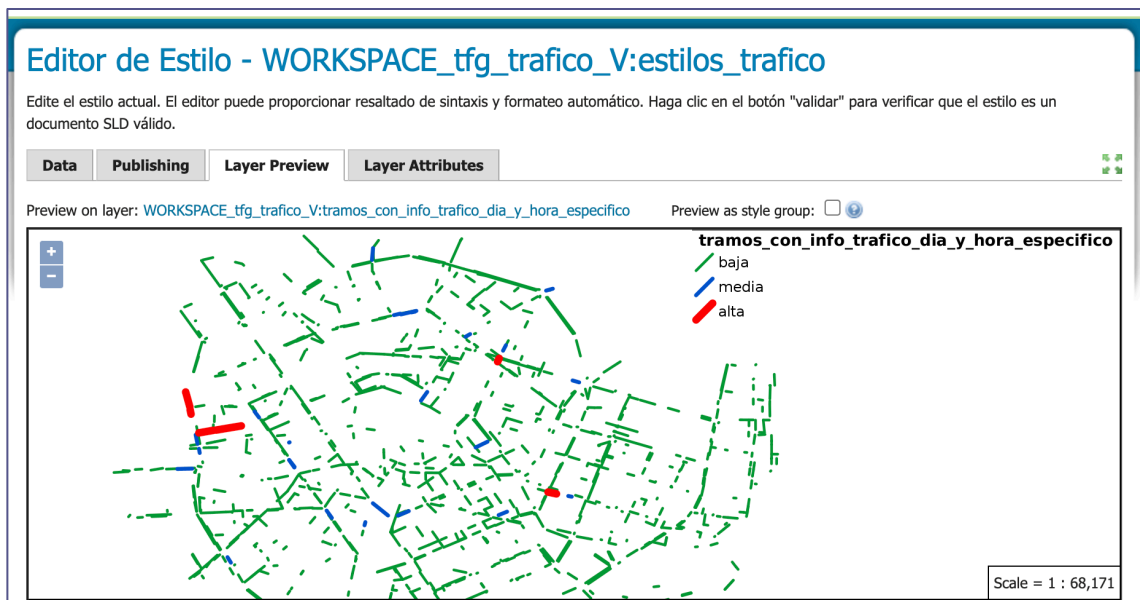


Figura 58. Capa alternativa "tramos_con_info_trafico_dia_y_hora_especificos" con el estilo "estilos_trafico" aplicado.

6.2.3.2. Capa de *OpenStreetMap* proporcionado por *Terrestris*

En esta ocasión se quiere crear una capa que sirva de fondo para el mapa interactivo. Se ha decidido usar un servicio *WMS*.

Se quiere usar el mapa de *OSM*, dada su naturaleza *de código abierto*, sin embargo, *OSM* no proporciona servicios de transmisión de datos espaciales. Lo que si permite *OSM* es descargar sus mapas, que subsecuentemente se pueden usar para crear servicios o productos propios.

*Terrestris*²¹ se sirve de mapas de *OSM* para crear sus propios servicios *WMS* personalizados, que se pueden consumir en *GeoServer*.

Para la capa que se quiere crear se ha decidido usar el servicio “*osm-gray*” de *Terrestris*, que ofrece varios mapas con una paleta de colores en escala de grises. De esta manera, se quiere hacer contraste con la capa de los tramos para ofrecer un visionado más intuitivo.

En primer lugar, hay que crear un nuevo almacén de datos para poder enlazar al servicio *WMS* de *Terrestris*. Desde el apartado “*Almacenes de Datos*”, se agrega un nuevo almacén seleccionando *WMS* como origen de datos (ver figura 60).



Figura 59. Selección de origen de datos en *GeoServer*.

Se usa la dirección de acceso al servicio²² para configurar la conexión (ver figura 61) y se la nombra “*TERRESTRIS_GRAY*”. Se puede observar en la figura 62 el nuevo almacén de datos.

²¹ <https://www.terrestris.de/en/openstreetmap-wms/>

²² <https://ows.terrestris.de/osm-gray/service?>

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

Editar conexión WMS

Editar los parámetros de conexión a un WMS remoto

Información básica del almacén

Espacio de trabajo *

WORKSPACE_tfg_trafico_V

Nombre del origen WMS *

TERRESTRIS_GRAY

Almacén WMS en cascada

Habilitado

Información de conexión

URL del documento Capabilities *

https://ows.terrestris.de/osm-gray/service?

Usuario

Contraseña

Habilitar pool de conexiones HTTP

Máximo número de conexiones simultáneas *

6

Tiempo de espera para obtener una conexión en segundos *

30

Tiempo máximo de lectura, en segundos *

60

[Guardar](#) [Cancelar](#)

Servidor

- Estado del servidor
- Logs de GeoServer
- Información de contacto
- Acerca de GeoServer
- Estado del proceso

Datos

- Previsualización de capas
- Espacios de trabajo
- Almacenes de datos
- Capas
- Grupos de capas
- Estilos

Servicios

- WMTS
- WCS
- WFS
- WMS
- WPS

Settings

- Global
- Procesado de imágenes
- Acceso Raster

Cacheado de Teselas

- Capas en caché
- Valores por defecto de cacheado

Figura 60. Detalles de conexión a servicio WMS de Terrestris.

Identificado como admin. [Cerrar sesión](#) es

Almacenes de datos

Gestionar los almacenes que proveen datos a GeoServer

+ Agregar nuevo almacén - Eliminar los almacenes seleccionados

Resultados 1 a 6 (de un total de 6 ítems)

<input type="checkbox"/>	Tipo de datos	Espacio de trabajo	Nombre del almacén	Tipo	¿Habilitado?
<input type="checkbox"/>		WORKSPACE_curso_webrmapping	DB_curso_webrmapping_SCHEMA_public	PostGIS	✓
<input type="checkbox"/>		WORKSPACE_tfg_CV	DB_tfg_SCHEMA_com_valenciana	PostGIS	✓
<input type="checkbox"/>		WORKSPACE_tfg_public	DB_tfg_SCHEMA_public	PostGIS	✓
<input type="checkbox"/>		WORKSPACE_tfg_trafico_V	DB_tfg_SCHEMA_trafico_valencia	PostGIS	✓
<input type="checkbox"/>		WORKSPACE_tfg_trafico_V	TERRESTRIS	WMS	✓
<input checked="" type="checkbox"/>		WORKSPACE_tfg_trafico_V	TERRESTRIS_GRAY	WMS	✓

Resultados 1 a 6 (de un total de 6 ítems)

Figura 61. Almacén de datos "TERRESTRIS_GRAY".

El almacén “TERRESTRIS_GRAY” es básicamente una puerta de enlace al servicio WMS de TERRESTRIS que ofrece múltiples capas seleccionables, observables en la figura 63. Para la capa que se quiere crear en GeoServer se va a hacer uso de la capa “OSM-WMS” del servicio de TERRESTRIS. El nombre de esta capa será “capa_OSM_terrestris_gris” (ver figura 64).

Nueva capa

Agregar nueva capa

Agregar capa de

Puede importar todas las capas WMS en cascada de una sola vez del almacén de datos seleccionado usando **importación en lote**

Esta es una lista de los recursos contenidos en el almacén 'TERRESTRIS_GRAY'. Haga click sobre la capa que desea configurar

<< < 1 > >> Resultados 0 a 0 (de un total de 0 ítems)

Publicada	Capa con espacio de nombres y prefijo	Acción
✓	OSM-WMS	Publicar de nuevo
	Dark	Publicación
	OSM-Overlay-WMS	Publicación
	OSM-WMS-no-labels	Publicación
	SRTM30-Colored	Publicación
	SRTM30-Colored-Hillshade	Publicación
	SRTM30-Contour	Publicación
	SRTM30-Hillshade	Publicación
	TOPO-OSM-WMS	Publicación
	TOPO-WMS	Publicación

<< < 1 > >> Resultados 0 a 0 (de un total de 0 ítems)

Figura 62. Capas proporcionadas por el servicio WMS de Terrestris.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

Editar capa

Editar los datos de la capa y la información de publicación

WORKSPACE_tfg_trafico_V:capa_OSM_terrestris_gris

Configure el recurso y la información de publicación para esta capa

Datos	Publicación	Dimensiones	Cacheado de Teselas	Seguridad
-------	-------------	-------------	---------------------	-----------

Editar capa

Información básica del recurso

Nombre del almacén: TERRESTRIS_GRAY

Native Name: OSM-WMS

Nombre

Habilitado

Anunciado

Título i18n

Resumen i18n

Figura 63. Detalles de la capa "capa_OSM_terrestris_gris".

En la figura 65 se puede visualizar la capa "capa_OSM_terrestris_gris" enfocada sobre la localidad de Valencia.

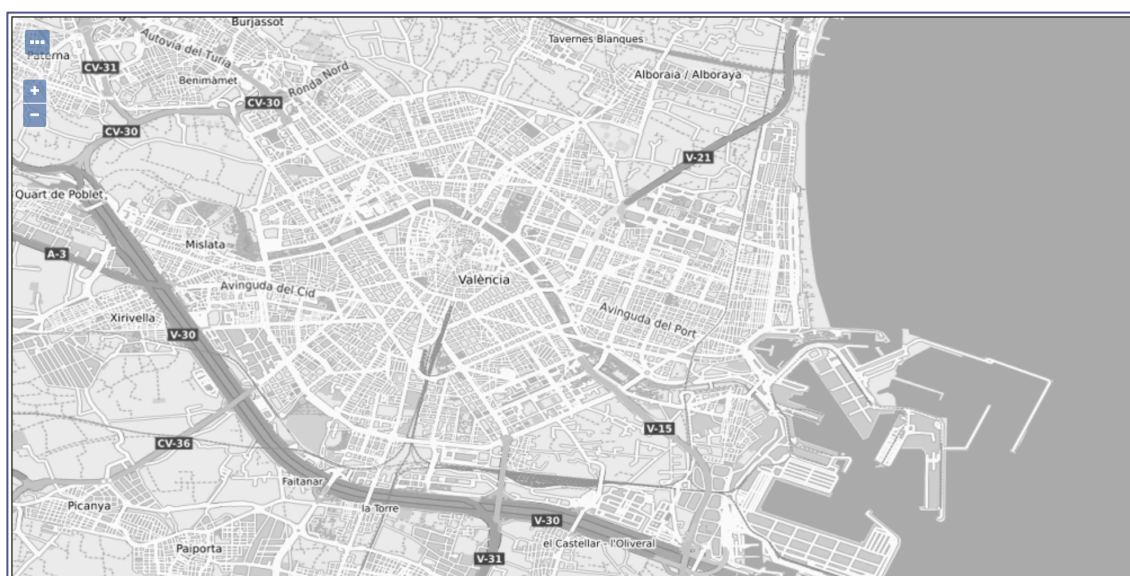


Figura 64. Previsualización de la capa "capa_OSM_terrestris_gris".

6.2.4. El Mapa

En este subapartado se van a usar las capas anteriormente definidas para crear un “Grupo de Capas” que combine las dos capas creadas durante la prueba.

El primer paso será acceder a la sección “Grupo de capas” y agregar un nuevo grupo de capas llamado “TRAFICO_ultima_hora_con_OSM” que será asignado al espacio de trabajo “WORKSPACE_tfg_trafico_V” (ver Figura 66).

Grupo de capas

Editar los contenidos de un grupo de capas

WORKSPACE_tfg_trafico_V:TRAFICO_ultima_hora_con_OSM

Configure las capas y la información de publicación para este grupo de capas

Datos | Publicación | Cacheado de Teselas | Seguridad

Nombre
TRAFICO_ultima_hora_con_OSM

Habilitado

Anunciado

Título i18n
TRAFICO_ultima_hora_con_OSM

Resumen i18n

Espacio de trabajo
WORKSPACE_tfg_trafico_V

Figura 65. Nombre y Espacio de trabajo del grupo de capas.

A continuación, se agregan las capas creadas y se establece el orden de dibujo (ver figura 67) de tal manera que el mapa proporcionado por el servicio de *Terrestris* sirva de fondo. A la capa con geometrías de los tramos de Valencia se le aplica el estilo “estilos_trafico”.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

Espacio de trabajo
 WORKSPACE_tfg_trafico_V

Límites
 Min X Min Y Máx X Máx Y
 -0.4179745912551 39.435417175292t -0.3241274058818 39.502529144287t

Sistema de coordenadas de referencia
 EPSG:4326 Buscar... EPSG:WGS 84...

Modo
 Single

Interrogable

Capas

Orden de dibujo	Tipo	Capa	Estilo por defecto	Estilo	Eliminar
1	Layer	WORKSPACE_tfg_trafico_V:capa_OSM_terrestris_gris	<input checked="" type="checkbox"/>		<input type="button" value="Eliminar"/>
2	Layer	WORKSPACE_tfg_trafico_V:tramos_con_info_trafico_ultimo_dia_hora	<input type="checkbox"/>	estilos_trafico	<input type="button" value="Eliminar"/>

Layer Group Styles

Copy from
 Elija uno Copiar ...

Vínculos a metadatos

No hay vínculos de metadatos hasta el momento

Tenga en cuenta que en las capabilities de WMS 1.1.1 sólomente se muestran los enlaces a metadatos de tipo FGDC y TC211.

Figura 66. Límites, capas, estilos y orden de dibujo del grupo de capas.

Finalmente, se puede usar el visualizador *OpenLayers* de *GeoServer* para observar el mapa de Valencia (ver figura 68) con información visual del tráfico de la última hora.

Scale = 1 : 34K

tramos_con_info_trafico_ultimo_dia_hora	fid	idta nombre	descripcion	sentido	fecha_hora	intensidad	fiabilidad_intensidad	ocupacion	fiabilidad_ocupacion	velocidad	fiabilidad_velocidad
tramos_con_info_trafico_ultimo_dia_hora.fid-714082aa_188640580ed_6ef1	287	P1036	Pl. MAURO GULLEM (de Llano de la Zaidia a Av. Menéndez Pidal)	Pte. Campanar	Aug 31, 2020 8:00:00 AM	1256.0	100	3.0	100	59.0	23
tramos_con_info_trafico_ultimo_dia_hora.fid-714082aa_188640580ed_6ef0	277	P1023	MAURO GULLEM, N° 1 (de Pte. de las Artes a Av. Burjassot)	Pte. San José	Aug 31, 2020 8:00:00 AM	123.0	100	1.0	100	33.0	25

Figura 67. Mapa dibujado por *OpenLayers*, centrado sobre Valencia y con información de tráfico de la última hora disponible.

6.2.5. Visualizador *Leaflet*

En este subapartado se quiere simular el consumo del mapa de tráfico desde un servidor externo. Se dispone de un *GeoServer* en una máquina local que incluye un servidor *Jetty* como se ha mencionado en el capítulo 4. Se ha elegido la librería *Leaflet* porque es una tecnología muy extendida dentro de los *SIG* además de ser muy accesible y de código abierto.

Para empezar, se habilitan los servicios del espacio de trabajo “*WORKSPACE_tfg_trafico_V*” (ver figura 69) que serán utilizados para consumir las capas desde el servidor local con *Leaflet*.

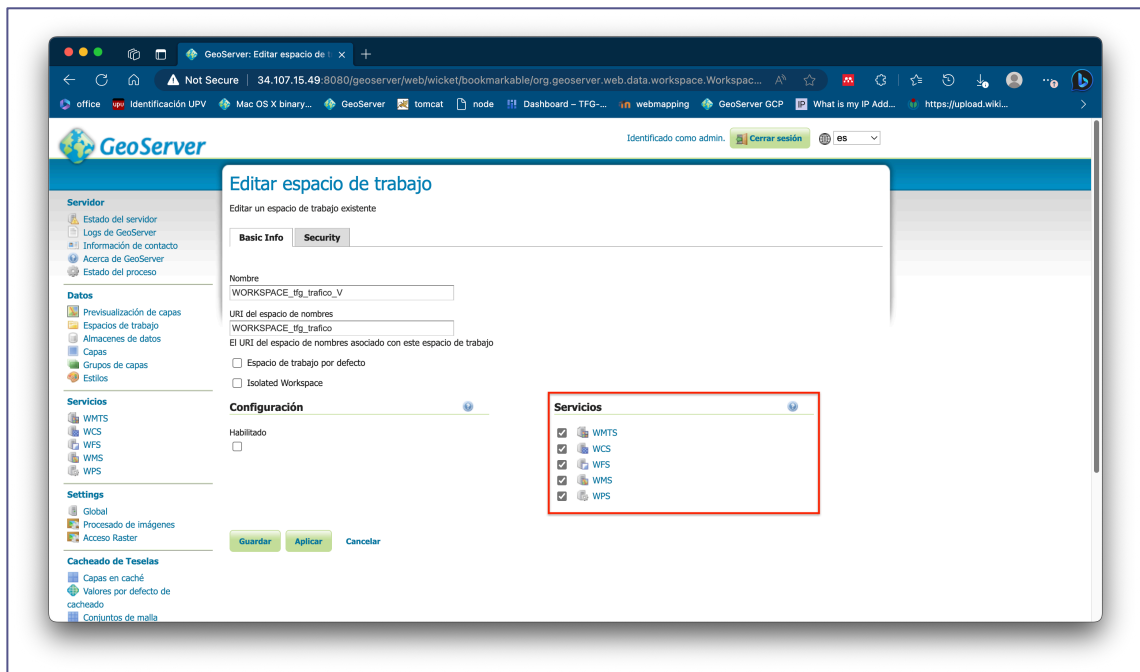


Figura 68. Opciones del espacio de trabajo “*WORKSPACE_tfg_trafico_V*”.

A continuación, se crea un fichero *HTML* llamado “*tfg_trafico_valencia_ejemplo_encuadre.html*” dentro del directorio local “*../geoserver/data_dir/www*” para poder ser ejecutado por *Jetty*.

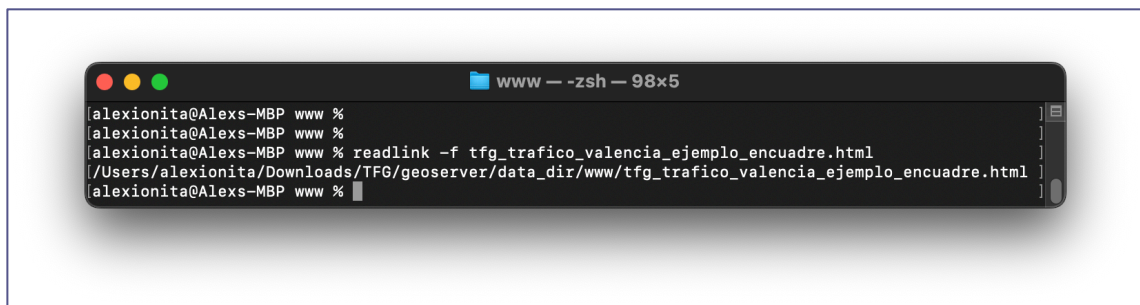
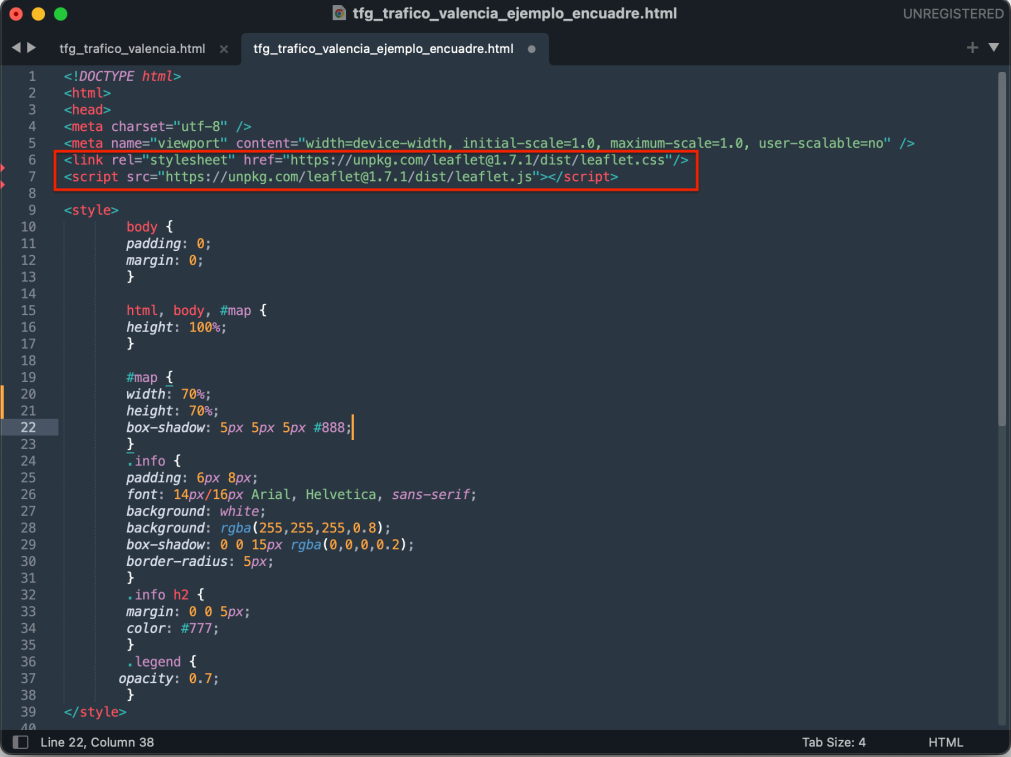


Figura 69. Localización de “*tfg_trafico_valencia_ejemplo_encuadre.html*”.

El contenido del fichero será un código sencillo que hace uso de la librería *JavaScript* de *Leaflet* instanciada a través de la URL “*https://unpkg.com/leaflet@1.7.1/dist/leaflet.js*” (ver figura 71) que permite crear un mapa compuesto por una capa (ver figura 72). Esta capa, de nombre “*tramos*”, es un objeto específico de *Leaflet* llamado “*tileLayer*” y

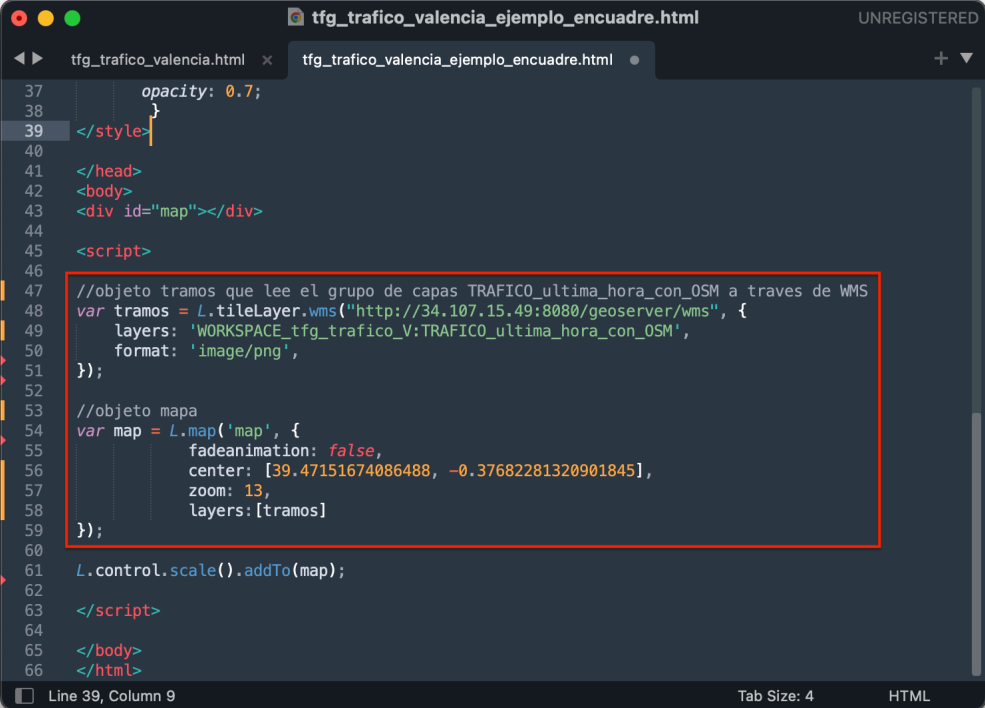
Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

contiene la llamada al servicio WMS, la cual, a través del enlace “<http://34.107.15.49:8080/geoserver/wms>”, devuelve el grupo de capas “*TRAFICO_ultima_hora_con_OSM*”. El objeto “*map*” contiene la función para representar gráficamente las capas pasadas como argumento, que en este caso será la capa “*tramos*”.



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8" />
5 <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
6 <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css"/>
7 <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
8
9 <style>
10   body {
11     padding: 0;
12     margin: 0;
13   }
14
15   html, body, #map {
16     height: 100%;
17   }
18
19   #map {
20     width: 70%;
21     height: 70%;
22     box-shadow: 5px 5px 5px #888;
23   }
24   .info {
25     padding: 6px 8px;
26     font: 14px/16px Arial, Helvetica, sans-serif;
27     background: white;
28     background: rgba(255,255,255,0.8);
29     box-shadow: 0 0 15px rgba(0,0,0,0.2);
30     border-radius: 5px;
31   }
32   .info h2 {
33     margin: 0 0 5px;
34     color: #777;
35   }
36   .legend {
37     opacity: 0.7;
38   }
39 </style>
40
```

Figura 70. Referencia hacia la librería Leaflet.



```
37     opacity: 0.7;
38   }
39 </style>
40
41 </head>
42 <body>
43 <div id="map"></div>
44
45 <script>
46
47 //objeto tramos que lee el grupo de capas TRAFICO_ultima_hora_con_OSM a través de WMS
48 var tramos = L.tileLayer.wms("http://34.107.15.49:8080/geoserver/wms", {
49   layers: 'WORKSPACE_tfg_trafico_V:TRAFICO_ultima_hora_con_OSM',
50   format: 'image/png',
51 });
52
53 //objeto mapa
54 var map = L.map('map', {
55   fadeanimation: false,
56   center: [39.47151674086488, -0.37682281320901845],
57   zoom: 13,
58   layers: [tramos]
59 });
60
61 L.control.scale().addTo(map);
62
63 </script>
64
65 </body>
66 </html>
```

Figura 71. Capas y mapa mediante objetos “map” y “tileLayer” de Leaflet.

Para comprobar la correcta ejecución del código, se puede acceder a “localhost:8080/geoserver/www/tfg_trafico_valencia_ejemplo_encuadre.html”. Se puede observar en la figura 73 el mapa del tráfico de Valencia servido por el GeoServer al servidor local Jetty a través de la librería Leaflet.

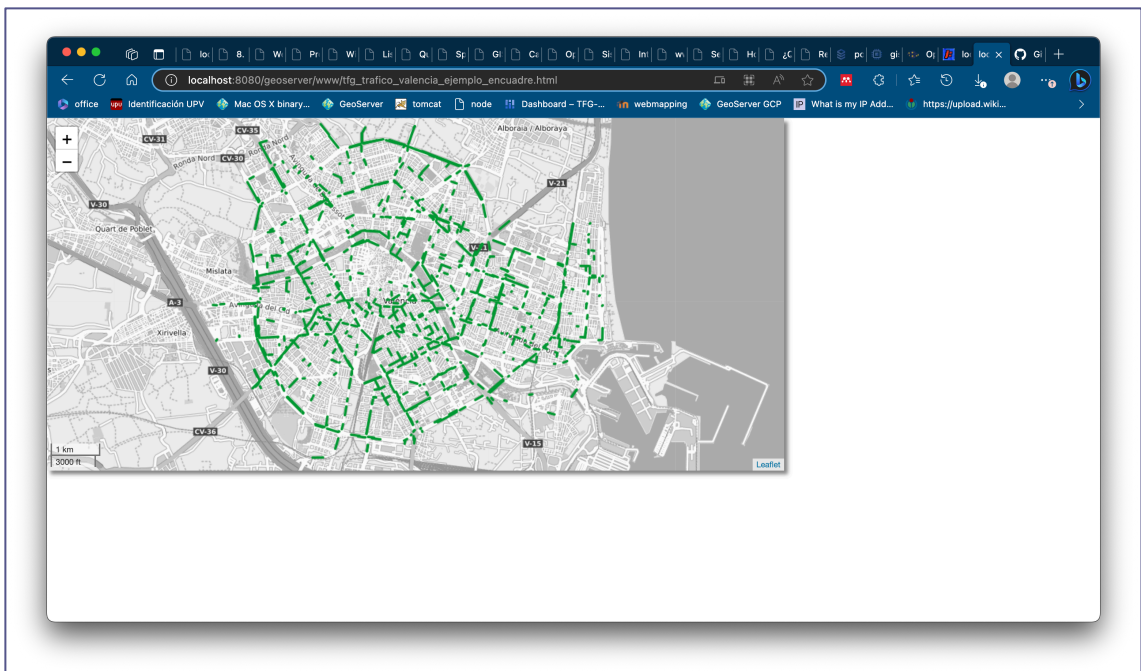


Figura 72. Web con el mapa del tráfico de Valencia usando Leaflet.

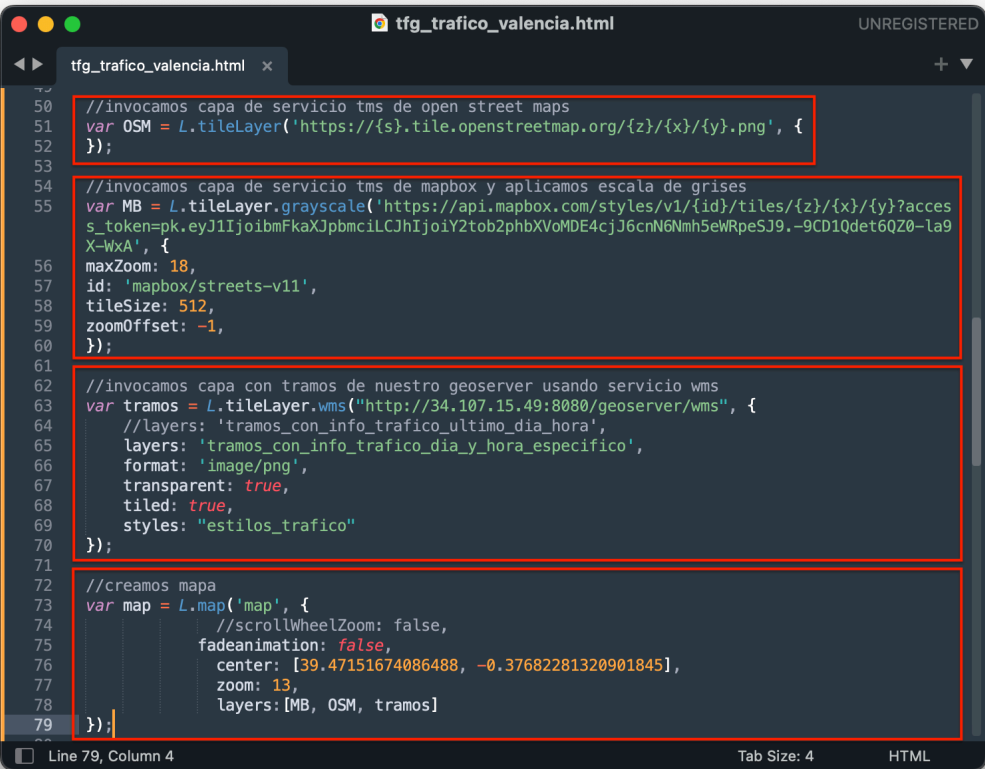
Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

6.2.6. Visualizador Leaflet con funcionalidad extendida

A lo largo de este ejemplo, se va a reemplazar el grupo de capas “*TRAFICO_ultima_hora_con_OSM*” por una capa de geometrías proporcionada por el *GeoServer* y dos capas base proporcionadas por servicios *TMS* disponibles en la web. Se va a desarrollar un caso de uso donde un servidor de datos geográficos, como *GeoServer*, proporciona capas sencillas con información muy específica, que luego serán combinadas con capas de servicios disponibles en la *WEB* para crear mapas interactivos.

La capa con los tramos y la información de tráfico que se va a usar en esta ocasión será “*tramos_con_info_trafico_dia_y_hora_especifico*” mencionada en el subapartado 6.2.3.1 en la figura 56. Se va a usar esta capa porque durante la fecha preseleccionada el tráfico de Valencia ha sido más diverso, lo que permite visualizar mejor las capacidades del mapa.

Las dos capas base provienen de *OpenStreetMap* y *MapBox* (ver figura 74).



```
50 //invocamos capa de servicio tms de open street maps
51 var OSM = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
52 });
53
54 //invocamos capa de servicio tms de mapbox y aplicamos escala de grises
55 var MB = L.tileLayer.grayscale('https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?acces
56 s_token=pk.eyJ1IjoibmFkaXJpbmciLCJhIjoiY2tob2phbXVoMDE4cjJ6cnN6Nmh5eWwRpeSj9,-9CD1Qdet6QZ0-la9
57 X-WxA', {
58 maxZoom: 18,
59 id: 'mapbox/streets-v11',
60 tileSize: 512,
61 zoomOffset: -1,
62 });
63
64 //invocamos capa con tramos de nuestro geoserver usando servicio wms
65 var tramos = L.tileLayer.wms("http://34.107.15.49:8080/geoserver/wms", {
66 //layers: 'tramos_con_info_trafico_ultimo_dia_hora',
67 layers: 'tramos_con_info_trafico_dia_y_hora_especifico',
68 format: 'image/png',
69 transparent: true,
70 tiled: true,
71 styles: "estilos_trafico"
72 });
73
74 //creamos mapa
75 var map = L.map('map', {
76 //scrollWheelZoom: false,
77 fadeAnimation: false,
78 center: [39.47151674086488, -0.37682281320901845],
79 zoom: 13,
80 layers: [MB, OSM, tramos]
81 });
```

Figura 73. Mapa en Leaflet compuesto por dos capas de *OpenStreetMap* y *Mapbox*, y una capa con información de tráfico del SIG del proyecto.

Para crear un mapa más interactivo, se va a usar *Leaflet* para añadir controladores de escala y capas, y una leyenda (ver figura 75). Además, también se ha creado una función que consulta la propiedad “*ocupacion*” de la capa “*tramos_con_info_trafico_dia_y_hora_especifico*” (ver figura 72).

```

tfg_ trafico_valencia.html x
UNREGISTERED
81 //añadimos control de escala
82 L.control.scale().addTo(map);
83
84 var baseMaps = {
85   "OpenStreetMap": OSM,
86   "GreyMapBox" : MB,
87 };
88
89 var overlayMaps = {
90   "Tramos": tramos
91 };
92
93 //añadimos control de capas
94 L.control.layers(baseMaps, overlayMaps).addTo(map);
95
96
97 //añadimos leyenda
98 var legend = L.control({position: 'bottomright'});
99 legend.onAdd = function (map) {
100   var div = L.DomUtil.create('div', 'info legend');
101   div.innerHTML += '';
102
103   return div;
104 };
105 legend.addTo(map);
106
Line 79, Column 4 Tab Size: 4 HTML

```

Figura 74. Código JavaScript con Leaflet para añadir escala, control de capas y leyenda al mapa.

```

tfg_ trafico_valencia.html x
UNREGISTERED
108 //añadimos función de consulta de propiedades
109 var owsrootUrl = 'http://34.107.15.49:8080/geoserver/ows';
110 var defaultParameters = {
111   service: 'WFS',
112   version: '1.0.0',
113   request: 'GetFeature',
114   typeName: 'WORKSPACE_tfg_trafico_V:tramos_con_info_trafico_dia_y_hora_especifico',
115   outputFormat: 'application/json'
116 };
117 var parameters = L.Util.extend(defaultParameters);
118 var URL = owsrootUrl + L.Util.getParamString(parameters);
119 $.ajax({
120   url: URL,
121   success: function (data) {
122     var geojson = new L.geoJson(data,
123     {style: {"color": "#2ECCFA", "opacity": "0", "weight": 2},
124     onEachFeature: function(feature, layer){
125       layer.bindPopup("La ocupacion del tramo " + feature.
properties.descripcion + " es del " + feature.
properties.ocupacion + "%." (" + feature.
properties.fecha_hora + ")");
126     }).addTo(map);
127   }
128 });
129
Line 79, Column 4 Tab Size: 4 HTML

```

Figura 75. Código JavaScript con Leaflet para añadir una función de consulta de propiedades al mapa.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

Se puede ver en las figuras 77 y 78 el nuevo mapa con tráfico de Valencia procesado por *Leaflet*. Para la base, el usuario puede elegir entre un mapa en escala de grises proporcionado por *MapBox* o el mapa base de *OpenStreetMap*, a través del controlador de la esquina superior derecha. Tiene visible una leyenda en la parte inferior derecha y un controlador de escala en la parte superior izquierda. Además, dispone de una funcionalidad de consulta de la ocupación de un tramo seleccionando con el cursor. Un pequeño recuadro mostrara el nombre, la ocupación y la hora.

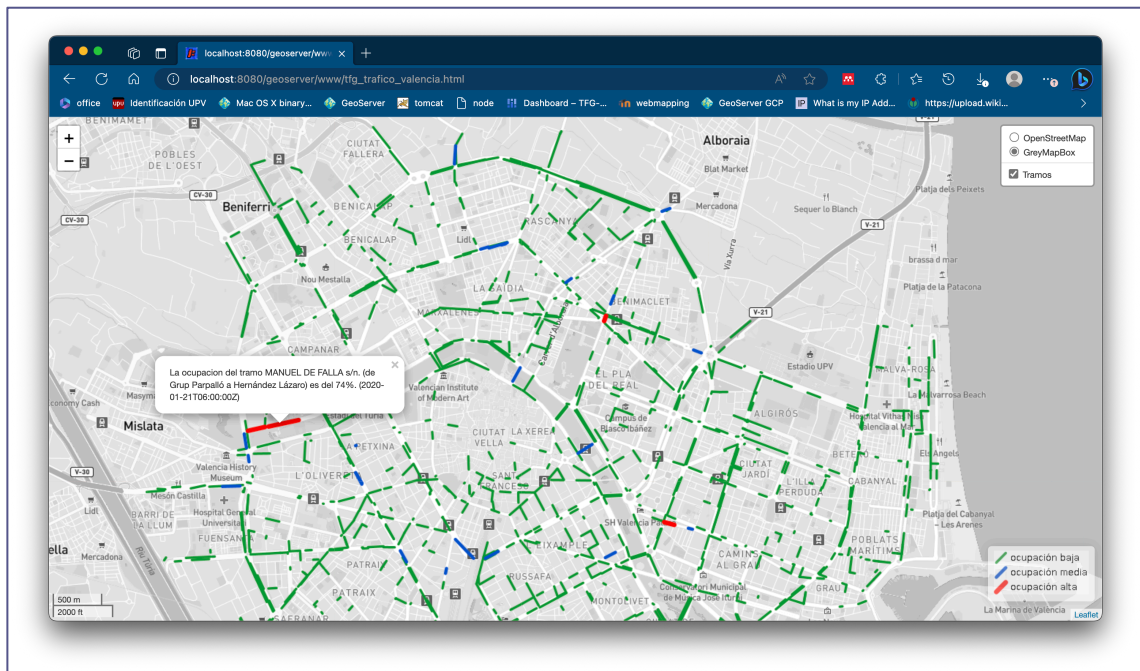


Figura 76. Mapa Leaflet con tráfico de Valencia, selector de escala, leyenda, función de consulta y capa base de MapBox seleccionada.

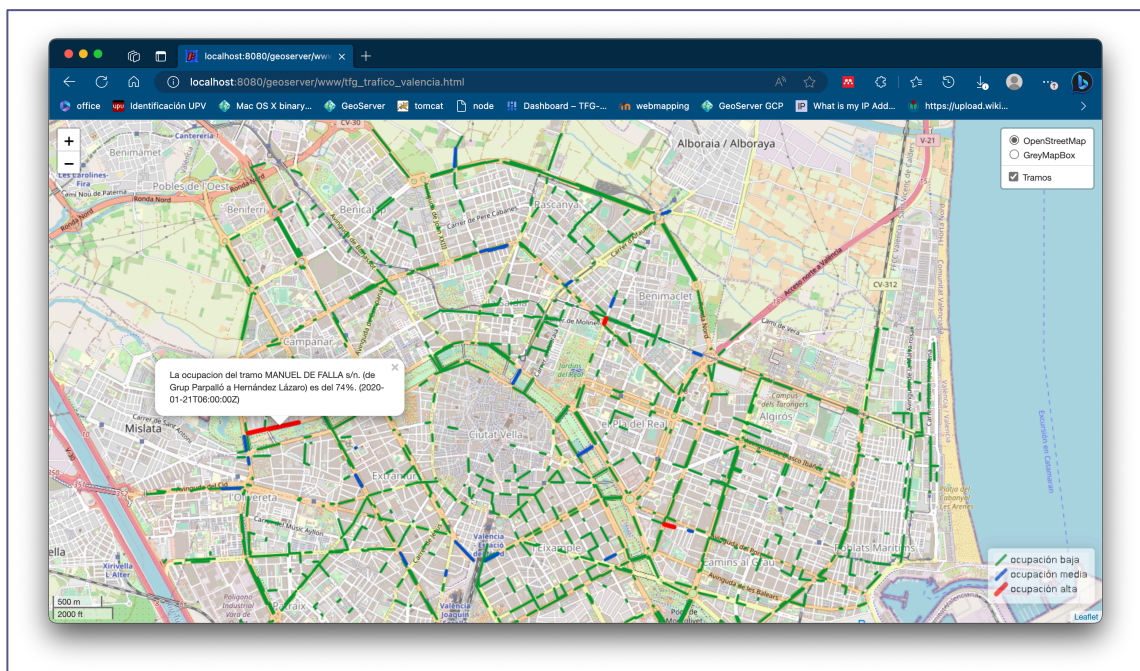


Figura 77. Mapa Leaflet con tráfico de Valencia, selector de escala, leyenda, función de consulta y capa base de OpenStreetMap seleccionada.

6.3. Comentarios sobre la Validación

Como apunte final, para este proyecto, se ha decidido no aplicar políticas de acceso a los servicios del servidor al escapar de los objetivos y no tener interés por la naturaleza de los datos utilizados. Los riesgos de incendios y un informe de tráfico son información de interés general y se quiere un acceso abierto para todo el público.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

7. Conclusiones

En este proyecto se ha diseñado, implementado y estructurado un sistema de información geográfica (*SIG*) utilizando soluciones geoespaciales de código abierto y tecnologías en la nube. Se ha tenido que investigar y adquirir conocimientos de cartografía, un área del que no se tenía experiencia previa.

Durante el proceso de desarrollo, se realizaron pruebas exhaustivas que validaron el correcto funcionamiento del sistema. Estas pruebas abarcaron desde la ejecución en los servicios en la nube, como *Google Cloud*, hasta la verificación de la precisión y eficacia del *SIG* en la gestión de datos geográficos y su representación.

Los resultados obtenidos cumplen los objetivos descritos en el primer capítulo. En la primera prueba de validación se hace una representación a nivel municipal de los datos del riesgo de incendio, con un estilo aplicado que facilita la percepción de patrones, relaciones y situaciones. En la segunda prueba se ha creado un mapa de tráfico local que puede ser alimentado con datos en tiempo real para, por ejemplo, facilitar la toma de decisiones de un ente administrativo. Además, los dos mapas tienen funciones de interactividad que permiten al usuario buscar información con distinto nivel de especificación y detalle.

En definitiva, el desarrollo del *SIG* se ha logrado mezclando conocimientos de sistemas informáticos, arquitectura de datos y cartografía. Se ha conseguido ampliar el conocimiento sobre los servicios en la nube, en particular *Google Cloud*, y comprender las ventajas que ofrecen en términos de escalabilidad, disponibilidad o seguridad. Se ha adquirido experiencia práctica en la configuración y administración de esta plataforma, así como en la instalación de sistemas de información geográfica en este entorno.

7.1. Relación del Trabajo Desarrollado con los Estudios Cursados y las Competencias Transversales

Este proyecto ha sido realizado gracias a los conocimientos adquiridos en varias asignaturas del Grado en Ingeniería Informática de la Universidad Politécnica de Valencia. Estas asignaturas han influido en el desarrollo del sistema de información geográfica de diferentes maneras.

Por ejemplo, “*Ingeniería de Software*” (*ISW*) y “*Gestión de Proyectos*” (*GPR*) han sido fundamentales para estructurar un plan de desarrollo viable y acorde a los recursos disponibles.

Las asignaturas de programación y de bases de datos, han proporcionado las técnicas necesarias para manipular los datos espaciales y crear el código que facilite la visualización e interacción del usuario. Se destacan en este aspecto asignaturas como “*Programación*” (*PRG*) y “*Bases de Datos*” (*PRG*).

No menos importantes han sido todos los conceptos de redes y sistemas operativos para poder instalar y administrar el proyecto en la plataforma de computación en la nube de *Google*, *GCP*.

Se deja a continuación detalles de la realización de este proyecto con respecto a las competencias transversales:

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

- **CT-01. Comprensión e integración:** al diseñar el sistema ha sido necesario adquirir conocimientos ajenos a los estudiados durante el grado.
- **CT-02. Aplicación y pensamiento práctico:** en el desarrollo de la solución que se explica durante el capítulo 5.
- **CT-03. Análisis y resolución de problemas:** en el análisis que se realiza del estado del arte de los SIG y la síntesis que se realiza durante el capítulo 2 del trabajo.
- **CT-04. Innovación, creatividad y emprendimiento:** durante las pruebas realizadas se ha tenido que pensar y diseñar de manera creativa para elegir y presentar ejemplos que puedan demostrar la capacidad y el valor social de un SIG.
- **CT-05. Diseño y proyecto:** en el diseño de la solución realizado en el capítulo 4.
- **CT-06. Trabajo en equipo y liderazgo:** al realizar diversas reuniones con el tutor y personas expertas para validar conocimientos.
- **CT-07. Responsabilidad ética, medioambiental y profesional:** al analizar los aspectos legales y éticos en el apartado 3.1.
- **CT-08. Comunicación efectiva:** en la elaboración de esta memoria se ha seguido el formato requerido y se ha puesto atención en organizar la información de manera clara y comprensible.
- **CT-09. Pensamiento crítico:** el análisis del problema durante el capítulo 3 ha obligado a identificar, analizar y proponer una solución que cumpla los objetivos expuestos en el apartado 1.1.
- **CT-10. Conocimiento de problemas contemporáneos:** el uso de servicios de computación en la nube.
- **CT-11. Aprendizaje permanente:** al investigar y comprender tecnologías ajenas a los ámbitos educativos y laborales de los que se tiene experiencia. Ha supuesto el autoaprendizaje de varias herramientas para poder desarrollar el proyecto final.
- **CT-12. Planificación y gestión del tiempo:** al realizar el plan de trabajo en el apartado 3.4.
- **CT-13. Instrumental específico:** al utilizar tecnologías especializadas como lo son, por ejemplo, *GeoServer* y *PostGIS*.

8. Trabajos Futuros

A continuación, se presenta una lista de posibles ampliaciones que podrían considerarse para mejorar o ampliar el proyecto:

- La creación de procesos *ETL* para carga de datos en tiempo real en *PostgreSQL*. Se puede implementar un sistema de extracción, transformación y carga, *ETL*, para realizar la carga automática de datos en tiempo real en la base de datos *PostgreSQL*. Esto permitiría mantener actualizada la información geoespacial y mejorar la gestión del *backend*.
- La optimización y limpieza de datos en la base de datos: Sería beneficioso realizar una optimización de los datos almacenados en la base de datos. Esto implica realizar tareas de limpieza, normalización y estandarización de datos, así como la creación de índices dinámicos para agilizar la consulta de datos. Además, se podría considerar la implementación de un sistema de archivado de datos antiguos en un entorno dedicado, con el objetivo de mantener el sistema de producción ligero y económico.
- Utilizar *OpenLayers* o *MapBox* para una representación más avanzada de los datos espaciales. Estas herramientas ofrecen una amplia gama de capacidades y funcionalidades para mejorar la visualización y el análisis de datos geoespaciales.
- La configuración de políticas de acceso en *GeoServer*. En proyectos en los que se manejan datos sensibles o que no deben hacerse públicos, es recomendable configurar políticas de acceso. Esto implica establecer usuarios y roles con diferentes niveles de privilegios para controlar el acceso a los datos. De esta manera, se garantiza la seguridad y la privacidad de la información.

Estas son solo algunas ideas de posibles ampliaciones para el proyecto. La implementación de estas mejoras podría brindar un mayor nivel de funcionalidad, rendimiento y seguridad al sistema de información geográfica desarrollado.



Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

9. Anexo

9.1. Relación del Trabajo con los Objetivos de Desarrollo Sostenible de la Agenda 2030 de Naciones Unidas

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.		X		
ODS 2. Hambre cero.		X		
ODS 3. Salud y bienestar.		X		
ODS 4. Educación de calidad.		X		
ODS 5. Igualdad de género.		X		
ODS 6. Agua limpia y saneamiento.		X		
ODS 7. Energía asequible y no contaminante.		X		
ODS 8. Trabajo decente y crecimiento económico.		X		
ODS 9. Industria, innovación e infraestructuras.		X		
ODS 10. Reducción de las desigualdades.		X		
ODS 11. Ciudades y comunidades sostenibles.		X		
ODS 12. Producción y consumo responsables.		X		
ODS 13. Acción por el clima.		X		
ODS 14. Vida submarina.		X		
ODS 15. Vida de ecosistemas terrestres.		X		
ODS 16. Paz, justicia e instituciones sólidas.		X		
ODS 17. Alianzas para lograr objetivos.		X		

El desarrollo de un sistema de información geográfica (SIG) tiene un potencial significativo para contribuir a los Objetivos de Desarrollo Sostenible (ODS) establecidos en la agenda 2030. Los SIG son herramientas que permiten capturar, almacenar, analizar y visualizar datos espaciales, lo que brinda oportunidades para abordar desafíos ambientales, sociales y económicos.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

En primer lugar, los *SIG* son fundamentales para el *ODS 11, ciudades y comunidades sostenibles*. Estas tecnologías pueden ayudar a planificar y gestionar el crecimiento urbano de manera sostenible, identificando áreas adecuadas para la construcción de infraestructuras, gestionando los recursos naturales de manera eficiente y mejorando la calidad de vida de los habitantes de las ciudades.

Además, los *SIG* también desempeñan un papel importante en el *ODS 13, acción por el clima*. Permiten la monitorización y evaluación de los efectos del cambio climático, así como la planificación de medidas de mitigación y adaptación. Los datos espaciales pueden ayudar a identificar áreas vulnerables, evaluar los impactos de eventos climáticos extremos y facilitar la toma de decisiones informadas para reducir las emisiones de gases de efecto invernadero.

Otro objetivo relevante es el *ODS 15, vida de ecosistemas terrestres*. Los *SIG* son herramientas esenciales para la gestión y conservación de los recursos naturales, como los bosques, las reservas naturales y las áreas protegidas. Permiten monitorizar y evaluar la biodiversidad, identificar patrones de deforestación o degradación del suelo, y apoyar la toma de decisiones para promover la conservación y el uso sostenible de los ecosistemas terrestres.

En el ámbito social, los *SIG* pueden contribuir al *ODS 3, salud y bienestar*. Estas tecnologías pueden ayudar a mapear y analizar la distribución de enfermedades, identificar áreas con mayor necesidad de servicios de salud y apoyar la planificación de infraestructuras sanitarias. Además, los *SIG* también pueden ser útiles en la respuesta a emergencias y desastres, facilitando la coordinación de recursos y la toma de decisiones en situaciones críticas.

En términos económicos, los *SIG* son relevantes para el *ODS 8, trabajo decente y crecimiento económico*. Estas herramientas pueden ayudar a identificar áreas con potencial económico, facilitar la planificación de infraestructuras y servicios, y apoyar la toma de decisiones en el ámbito empresarial. Los *SIG* también pueden contribuir a la reducción de la desigualdad, al facilitar el acceso a información geoespacial para comunidades desfavorecidas o rurales.

En conclusión, el desarrollo de un sistema de información geográfica puede tener una estrecha relación con los Objetivos de Desarrollo Sostenible.

10. Glosario

- **SIG** sistema de información geográfica.
- **Web-Mapping** es una técnica que consiste en mostrar y visualizar datos geográficos en un entorno *WEB*.
- **Frontend** es la parte de una aplicación que interactúa con los usuarios.
- **Backend** es la parte de una aplicación *WEB* que contiene la lógica y la capa de acceso a datos.
- **Shapefile** formato de archivo utilizado en los *SIG* para almacenar datos geoespaciales, como puntos, líneas y polígonos.
- **GeoJSON** formato de datos utilizado para representar y almacenar información geoespacial.
- **GeoTIFF** formato de archivo utilizado para almacenar imágenes ráster georreferenciadas.
- **SRC** sistema de referencia de coordenadas.
- **SRID** identificador de referencia espacial.
- **HTTP** protocolo utilizado para la comunicación y transferencia de datos en la *WEB*.
- **HTTPS** versión segura del protocolo *HTTP* que utiliza cifrado.
- **GPS** sistema de navegación por satélite que permite determinar la ubicación precisa en la superficie terrestre.
- **VPN** red privada virtual.
- **EDT** estructura de desglose del trabajo.
- **IP** protocolo de comunicación utilizado para identificar y enviar datos entre dispositivos en una red.
- **SSH** protocolo de red que permite la conexión segura y el acceso remoto a sistemas informáticos a través de una conexión cifrada.
- **CSS** lenguaje de hojas de estilo utilizado para controlar la apariencia y el diseño de documentos *HTML*.
- **URL** dirección que se utiliza para acceder a un recurso específico en internet, como una página *WEB* o un archivo.
- **CSV** formato de archivo utilizado para almacenar datos en formato tabular, donde los valores están separados por comas.
- **OSM** OpenStreetMap.

Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.



11. Referencias Bibliográficas

- [1] «¿Qué son los SIG? | Tecnología de representación cartográfica con sistemas de información geográfica». <https://www.esri.com/es-es/what-is-gis/overview> (accedido 22 de junio de 2023).
- [2] «ArcGIS for Server vs Open Source - MappingGIS». <https://mappinggis.com/2012/07/arcgis-for-server-vs-open-source/> (accedido 22 de junio de 2023).
- [3] «3. Vector Data — QGIS Documentation documentation». https://docs.qgis.org/3.28/en/docs/gentle_gis_introduction/vector_data.html (accedido 22 de junio de 2023).
- [4] «6. Datos Raster — documentación de QGIS Documentation -». https://docs.qgis.org/3.28/es/docs/gentle_gis_introduction/raster_data.html (accedido 22 de junio de 2023).
- [5] «Winkel III—ArcGIS Pro | Documentación». <https://pro.arcgis.com/es/pro-app/latest/help/mapping/properties/winkel-tripel.htm> (accedido 22 de junio de 2023).
- [6] «8. Sistema de Referencia de Coordenadas — documentación de QGIS Documentation -». https://docs.qgis.org/3.28/es/docs/gentle_gis_introduction/coordinate_reference_systems.html (accedido 22 de junio de 2023).



Infraestructura basada en tecnologías de cartografía en la web para crear mapas interactivos que faciliten la visualización de información a nivel de municipio, código postal y barrio.

12. Bibliografía

- «Vector vs. Raster Images: Choosing the Right Format > DINFOS Pavilion > Article». <https://pavilion.dinfos.edu/Article/Article/2223089/vector-vs-raster-images-choosing-the-right-format/> (accedido 27 de junio de 2023).
- «Introduction to PostGIS — Introduction to PostGIS». <http://postgis.net/workshops/postgis-intro/index.html> (accedido 27 de junio de 2023).
- H. Fernandes *et al.*, «Design of geographic information systems to promote accessibility and universal access», *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10279 LNCS, pp. 291-299, 2017, doi: 10.1007/978-3-319-58700-4_24.
- «Qué son los códigos EPSG / SRID y su vinculación con PostGIS». <https://mappinggis.com/2016/04/los-codigos-epsg-srid-vinculacion-postgis/> (accedido 25 de junio de 2023).
- «Sistemas de coordenadas, proyecciones de mapas y transformaciones—ArcGIS Pro | Documentación». <https://pro.arcgis.com/es/pro-app/latest/help/mapping/properties/coordinate-systems-and-projections.htm> (accedido 25 de junio de 2023).
- «Una Introducción fácil a GIS — documentación de QGIS Documentation ->». https://docs.qgis.org/3.28/es/docs/gentle_gis_introduction/index.html (accedido 25 de junio de 2023).