



Trabajo de Fin de Grado

Simulador para la optimización de despliegues IoT
(LoRaWAN)

Autor

Pablo Sanz Alguacil

Directores

Dr. D. Julio Alberto Sangüesa Escorihuela
D. Pablo Doñate Navarro

Escuela Universitaria Politécnica de Teruel.

2024

TABLA DE CONTENIDOS

1. INTRODUCCIÓN, MOTIVACIÓN Y OBJETIVOS	1
2. CONTEXTUALIZACIÓN	3
2.1. ¿Qué son LoRa y LoRaWAN?	3
2.2. ¿Qué es la zona de Fresnel?.....	4
3. ESTADO DEL ARTE	7
4. INTEGRACIÓN DEL SIMULADOR EN FRESSIM.....	10
5. SIMULADOR DE COBERTURA	11
5.1. Generación de mapas de cobertura.....	11
5.2. Mitigación de las limitaciones.....	14
5.2.1. Migración de los cálculos al servidor.....	14
5.2.2. Reducción de peticiones.....	15
5.2.3. Resultados	18
5.2.4. Fichero de configuración.....	19
6. OPTIMIZACIÓN DEL RENDIMIENTO	20
6.1. Reaprovechamiento de datos	20
6.2. Migración de la API de elevaciones	22
6.3. Mapas	24
6.3.1. Diferentes mapas base.....	24
6.3.2. Filtros del mapa.....	26
7. RESULTADOS	28
7.1. Rendimiento	28
7.1.1. Tiempos.....	28
7.1.2. Número de peticiones.....	29
7.1.3. Fluctuaciones entre las versiones sin optimizar y optimizada.....	30
7.2. APIs de elevaciones.....	33
7.2.1. Selector de modo.....	33
7.2.2. APIs de elevaciones self-hosted	34
7.2.3. Selector de API	36
7.2.4. Estimación de peticiones.....	37
7.3. Despliegue en el servidor	38

8. PROBLEMAS ENCONTRADOS.....	40
8.1. Fallos de petición.....	40
8.2. Rendimiento en cuadrículas grandes	41
8.3. La API de elevaciones de Bing Maps dejó de funcionar.....	41
8.4. Perdida de precisión con el modo velocidad	42
8.5. Espacio en disco necesario	44
8.6. Peticiones pesadas	45
9. CONCLUSIÓN.....	46
10. REFERENCIAS	48
ANEXO I. CARACTERÍSTICAS DE LORAWAN.....	I
ANEXO II. TECNOLOGIAS UTILIZADAS	III
ANEXO III. DESPLIEGUE DE OPEN TOPO.....	V
1. Clonar repositorio oficial.....	V
2. Descargar dataset.....	V
3. Configuración de Open Topo Data.....	VIII
4. Compilar y ejecutar la API con Docker.....	VIII
5. Acceso a la API	IX
ANEXO IV. MANUAL DE USUARIO.....	X
1. Selección de la API de elevaciones	X
2. Selección de modo de análisis.....	XI
3. Carga de datos de fuentes externas.....	XII
4. Manejo de los nodos y gateways	XII
4.1 Añadir elementos	XII
4.2 Eliminar elementos	XIII
4.3 Modificar la posición de un gateway	XIV
4.4 Modificar la altura de un elemento	XV
4.5 Cambiar nombre de un elemento	XVI
4.6 Modificar RSSI y SNR de un nodo.....	XVI
5. Generación de gráficas punto a punto	XVIII

6.	Mapas y filtros.....	XX
7.	Exportar datos.....	XXI
8.	Generar mapas de cobertura	XXII

ÍNDICE DE ILUSTRACIONES

Ilustración 1 - Funcionalidad principal de FresSim.....	10
Ilustración 2 - Primer mapa de cobertura generado	13
Ilustración 3 - Área de 500km ² con puntos de control	17
Ilustración 4 – Área generada con sectores sin valores aproximados.....	21
Ilustración 5 - Resultados de cobertura sobre mapa satelital.....	25
Ilustración 6 - Resultados de cobertura sobre mapa topográfico	25
Ilustración 7 - Resultados de cobertura sobre mapa en escala de grises.....	25
Ilustración 8 - Mapa de cobertura con filtrado de sectores activo	26
Ilustración 9 - Cobertura de Teruel con nodos de comprobación (original).....	33
Ilustración 10 - Cobertura de Teruel (optimizada)	33
Ilustración 11 - Selector de modo	34
Ilustración 12 - Selector de API de elevaciones	37
Ilustración 13 - Earthdata EULAs	VI
Ilustración 14 - Filtro de Earthdata	VI
Ilustración 15 - Descarga de Earthdata (parte 1)	VII
Ilustración 16 - Descarga de Earthdata (parte 2)	VII
Ilustración 17 - Fichero de configuración de Open Topo.....	VIII
Ilustración 18 - Respuesta de Open Topo	IX
Ilustración 19 - Botón del selector de API.....	X
Ilustración 20 - Panel del selector de API.....	X

Ilustración 21 - Botón del selector de modo	XI
Ilustración 22 - Panel del selector de modo	XI
Ilustración 23 - Botón de carga de datos.....	XII
Ilustración 24 - Nodos y gateways en el mapa	XIII
Ilustración 25 - Confirmación añadir gateway	XIII
Ilustración 26 - Botón para eliminar nodos y gateway	XIV
Ilustración 27 - Botón para modificar posición de nodo y gateway	XIV
Ilustración 28 - Panel de modificación de posición	XV
Ilustración 29 - Botón de modificación de la altura de un gateway	XV
Ilustración 30 - Botón de edición del nombre.....	XVI
Ilustración 31 - Botón de modificación del RSSI y SNR	XVII
Ilustración 32 - Panel de modificación del RSSI y SNR	XVII
Ilustración 33 - Botón de generación de gráficas	XVIII
Ilustración 34 - Panel de generación de gráficas	XIX
Ilustración 35 - Gráfica de la zona de Fresnel y elevaciones.....	XIX
Ilustración 36 - Botón de capas del mapa	XX
Ilustración 37 - Panel de capas del mapa.....	XXI
Ilustración 38 - Botón para exportar	XXI
Ilustración 39 - Botón para generar mapas de cobertura	XXII
Ilustración 40 - Panel de generación de mapas de cobertura.....	XXIII

ÍNDICE DE FIGURAS

Figura 1 - Elipses de la zona de Fresnel entre un emisor y receptor	4
Figura 2 - Montañas interfiriendo en la zona de Fresnel	5
Figura 3 - Zona de Fresnel y curvatura de la Tierra	6
Figura 4 – Arquitectura de procesamiento individual	12
Figura 5 - Peticiones individuales.....	15
Figura 6 - Peticiones por lotes	16
Figura 7 - Arquitectura de procesamiento en lotes	17
Figura 8 - Petición de sectores externos	20
Figura 9 - Ejemplo de área de análisis por sectores.....	43
Figura 10 - Número de puntos intermedios en función de la distancia	43

ÍNDICE DE TABLAS

Tabla 1 - Métricas de la versión original	14
Tabla 2 - Métricas de la versión original mejorada	18
Tabla 3 - Métricas de la versión optimizada.....	22
Tabla 4 - Comparación de las APIs de elevaciones.....	23
Tabla 5- Comparación de tiempos entre versiones	29
Tabla 6- Comparación de número de peticiones entre versiones	30
Tabla 7 - Variaciones de la versión optimizada en áreas de 1x1 km.....	31
Tabla 8 – Variaciones de la versión optimizada en áreas de 5x5 km	31
Tabla 9 - Variaciones de la versión optimizada en áreas de 10x10 km.....	32
Tabla 10 - Variaciones de la versión optimizada en áreas de 20x20 km.....	32
Tabla 11 - Métricas de tiempo de Open Topo en función del modo	35
Tabla 12 - Comparación de tiempos en modo velocidad entre servidor y local.....	38
Tabla 13 - Comparación de tiempos en modo precisión entre servidor y local.....	39

RESUMEN

Este TFG propone el desarrollo de un simulador para la optimización de despliegues de redes LoRaWAN. Este sistema está diseñado para mejorar la planificación de redes mediante análisis avanzados de cobertura basados en el cálculo de la zona de Fresnel, integrando datos topográficos y geoespaciales.

La herramienta desarrollada permite generar mapas de cobertura que pueden cubrir grandes áreas y permiten previsualizar el impacto de la topografía en la calidad de la señal, facilitando la toma de decisiones sobre la ubicación idónea de nodos y gateways. Además, cuenta con un sistema flexible para gestionar recursos de APIs de elevación públicas o privadas y diversas herramientas que ayudan al usuario a utilizar la herramienta y visualizar los resultados obtenidos.

El resultado es una solución escalable y económica para el despliegue de redes LoRaWAN en escenarios complejos.

Palabras clave

LoRaWAN, Internet de las Cosas (IoT), Redes inalámbricas, Zona de Fresnel, Simulación de Cobertura.

ABSTRACT

This Final Degree Project proposes the development of a simulator for optimizing LoRaWAN network deployments. This system is designed to enhance network planning through advanced coverage analysis based on the Fresnel Zone calculus, integrating topographic and geospatial data.

The developed tool generates coverage maps spanning large terrain areas, allowing users to previsualize the topography's impact on signal quality, thereby facilitating decision-making regarding the optimal location of nodes and gateways. Additionally, it features a flexible system for managing public or private elevation API resources, along with various tools to help users use the application and interpret the results obtained.

The result is a scalable and cost-effective solution for deploying LoRaWAN networks in complex scenarios.

Palabras clave

LoRaWAN, Internet of Things (IoT), Wireless Networks, Fresnel Zone, Coverage Simulation.

1. INTRODUCCIÓN, MOTIVACIÓN Y OBJETIVOS

La evolución del Internet de las Cosas (IoT) ha revolucionado múltiples sectores, permitiendo la creación de redes para la monitorización y gestión inteligente de recursos. En este contexto, han emergido tecnologías como LoRaWAN (Long Range Wide Area Network), que ha ganado relevancia por ofrecer comunicación inalámbrica de largo alcance con bajo consumo energético. Estas características han convertido a LoRaWAN en una solución idónea para aplicaciones en monitorización ambiental, agricultura y gestión de infraestructuras en entornos urbanos y rurales.

Sin embargo, en áreas con orografía compleja se presentan ciertas complicaciones, especialmente donde las características del terreno pueden influir en la calidad y cobertura de la señal. Para garantizar una cobertura óptima, y por tanto, un buen uso de los recursos de red es indispensable elegir correctamente la ubicación de gateways y nodos. Para ello, el análisis de la Zona de Fresnel resulta una herramienta muy útil, al permitir evaluar los obstáculos orográficos a lo largo del camino de la señal y optimizar el posicionamiento de los dispositivos teniendo en cuenta el terreno.

El objetivo de este Trabajo Fin de Grado (TFG) se enfoca en el diseño e implementación de una herramienta para la planificación de despliegues LoRaWAN y su incorporación a una aplicación ya existente. Esta herramienta se basa en los cálculos de la zona de Fresnel junto con los datos de varios DEMs (Digital Elevation Model) obtenidos mediante APIs de elevación para generar mapas de cobertura que pueden cubrir zonas extensas y que permitan simular y evaluar de forma eficiente la cobertura que tendrá un nodo en el área circundante de un gateway. El proyecto busca además proporcionar una interfaz intuitiva que permita a los usuarios analizar estas grandes áreas sin necesidad de comprobar puntos de forma individual. Para ello, los mapas que se generan están compuestos por sectores, para los cuales se pueden configurar distintas resoluciones y ajustar así la precisión a las necesidades del usuario.

Adicionalmente, en la implementación se incluye Docker para un despliegue sencillo y rápido de la aplicación sin necesidad de tener en cuenta el entorno en el que se vaya a alojar y aumentando así la portabilidad y compatibilidad de la aplicación.

Los objetivos específicos son:

- Implementar la visualización de mapas de cobertura, integrando resultados de análisis topográficos y geoespaciales para facilitar la evaluación.
- Crear una interfaz de usuario clara y funcional, que permita a los usuarios seleccionar parámetros, visualizar resultados y tomar decisiones informadas sobre el posicionamiento de nodos y gateways.
- Optimizar la herramienta existente, mejorando la eficiencia y reduciendo el tiempo de procesamiento, haciendo posible el análisis de grandes áreas en tiempos mínimos.
- Desarrollar un modelo de cobertura avanzado que incorpore el análisis de la Zona de Fresnel por sectores, mejorando la precisión en la estimación de la señal.

En conclusión, este proyecto no solo busca superar las limitaciones de otras herramientas existentes, sino también ofrecer una solución escalable, flexible y precisa para la planificación de despliegues IoT. Al abordar tanto los desafíos técnicos como las necesidades prácticas de los usuarios, esta herramienta contribuirá significativamente a la adopción de redes LoRaWAN en escenarios reales y complejos.

2. CONTEXTUALIZACIÓN

En este capítulo se establece el marco teórico necesario para comprender las tecnologías LoRa y LoRaWAN, destacando sus características clave, como el bajo consumo energético, la comunicación a largas distancias y su topología de red en estrella, junto con sus principales ventajas y limitaciones en aplicaciones IoT.

También, se aborda el concepto de la zona de Fresnel, necesario para evaluar la propagación de la señal y el impacto de los obstáculos en la calidad de esta. Estos elementos teóricos son fundamentales para entender los desafíos y enfoques utilizados en el análisis de cobertura desarrollado en este trabajo.

2.1. ¿Qué son LoRa y LoRaWAN?

LoRa es una tecnología de modulación de radiofrecuencia patentada por Semtech[10] en 2014. Utiliza una técnica de espectro ensanchado llamada Chirp Spread Spectrum (CSS), que aporta robustez frente a interferencias y ruido en el entorno. Este método modula las señales utilizando un cambio gradual de frecuencia (chirp) y ofrece una alta sensibilidad en los receptores, lo que se traduce en un alcance extendido incluso con niveles bajos de potencia.

Las principales características de LoRa son las bajas tasas de datos utilizadas, la comunicación a largas distancias (más reducido en entornos urbanos) y el uso de bandas de frecuencia libre.

LoRaWAN es el protocolo que complementa la capa física LoRa, proporcionando una arquitectura de red y un modelo de comunicación que permite la interoperabilidad entre dispositivos y aplicaciones. Es un estándar abierto desarrollado por la LoRa Alliance [11], lo que facilita la implementación y la integración en redes privadas y públicas. Es ideal para aplicaciones que requieren una comunicación eficiente en términos de energía y costes en áreas geográficamente amplias, con poca necesidad de transmisión en tiempo real: ciudades inteligentes, agricultura inteligente, rastreo de activos y mediciones remotas. Las características principales se detallan de forma más extensa en el [Anexo I](#).

En contraposición, también tienen dos aspectos negativos que pueden no hacerlas aptas para determinados proyectos. El primero es la velocidad limitada que tiene esta red, la

cual no la hace apta para implementaciones que requieran envío de volúmenes grandes de datos necesarios en poco tiempo (como video por ejemplo). El otro punto es la latencia, que debido a las ventanas de escucha limitadas que tienen los dispositivos de la red, pueden aumentar los tiempos de respuesta.

En definitiva, ambas son tecnologías destacables para el despliegue de redes IoT, permitiendo conectar dispositivos en entornos remotos y urbanos de manera eficiente. Su diseño robusto y adaptable las posiciona como soluciones clave para aplicaciones donde el bajo consumo energético y la larga distancia son primordiales. Sin embargo, es necesario abordar desafíos como la velocidad y la latencia para expandir aún más su adopción en diversas industrias.

2.2. ¿Qué es la zona de Fresnel?

La zona de Fresnel es un concepto fundamental en el análisis de la propagación de ondas electromagnéticas, especialmente en sistemas de comunicación inalámbrica. En términos simples, se trata de una serie de elipsoides concéntricos imaginarios que se generan entre un transmisor y un receptor de señal (Figura 1). Estas zonas determinan cómo las ondas se propagan en un entorno, particularmente en términos de difracción y los efectos de los obstáculos en la trayectoria de la señal.

Cada zona de Fresnel tiene la forma de un elipsoide con el transmisor y el receptor en sus dos focos. Cuanto más se aumente la distancia entre emisor y receptor, más anchas se harán las elipses. Las ondas que viajan dentro de estas zonas contribuyen constructiva o destrutivamente al total de la señal recibida y se numeran a partir del centro (primera zona) hacia afuera. Dentro de la primera zona de Fresnel las ondas suelen contribuir positivamente a la señal, mientras que las de las zonas exteriores pueden causar interferencia destructiva.

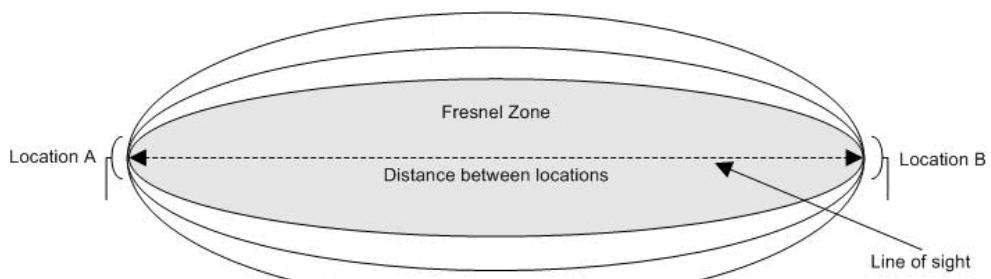


Figura 1 - Elipses de la zona de Fresnel. Obtenida de Air-Stream Wireless. (n.d.). *Fresnel zone*.

La mayoría de las señales que viajan desde un transmisor de radio hasta el receptor lo hacen a través de un camino directo entre los dos puntos, sin embargo, una cantidad significativa de las señales viaja fuera del eje. Las señales que viajan fuera del eje pueden desviarse por obstáculos, lo que puede hacer que lleguen desfasadas y causen interferencia destructiva.

De todas las zonas de Fresnel, la primera es la más relevante para la propagación eficiente de las señales porque contiene las trayectorias más directas y efectivas de las ondas hacia el receptor. En esta zona, las ondas llegan con la menor cantidad de difracción y mayor intensidad, lo que asegura una señal fuerte y estable. Además, las ondas se suman constructivamente, mejorando la intensidad de la señal. Cuando hay obstáculos (edificios, árboles, montañas, etc.) dentro de la primera zona de Fresnel, estos pueden causar bloqueo directo, difracción e interferencias (Figura 2).

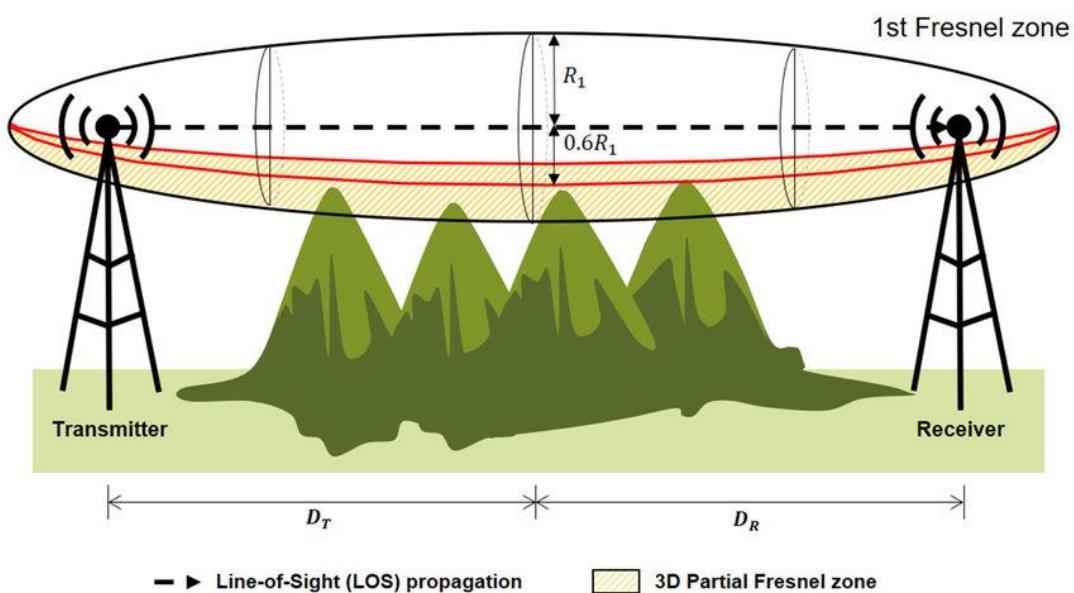


Figura 2 - Montañas interfiriendo en la zona de Fresnel. Obtenida de Baek, Jieun & Choi, Yosoon. (2018). *A New GIS-Based Algorithm to Support Initial Transmitter Layout Design in Open-Pit Mines*

Cuando un objeto obstruye completamente la línea de vista, la señal experimenta una pérdida significativa o total debido a que las ondas electromagnéticas no pueden pasar a través del obstáculo de forma efectiva, salvo por pequeñas porciones difractadas alrededor del mismo. Si un objeto invade parcial o completamente la primera zona de Fresnel, las ondas pueden doblarse alrededor de sus bordes y aunque la señal puede llegar al receptor, lo hace con menor intensidad debido a la pérdida de contribución directa. Los

obstáculos también pueden causar la reflexión de las ondas hacia el receptor. Estas ondas reflejadas pueden interferir con las ondas directas, creando efectos de aumento temporal de la intensidad de la señal o de reducción de la señal debido al desfasamiento entre las ondas directas y reflejadas.

Además, medida que aumenta la distancia entre el transmisor y el receptor, la curvatura de la Tierra también se convierte en un factor. Incluso si el área entre los dispositivos está despejada, el terreno en sí podría convertirse en una obstrucción debido al tamaño de la Zona de Fresnel (Figura 3).

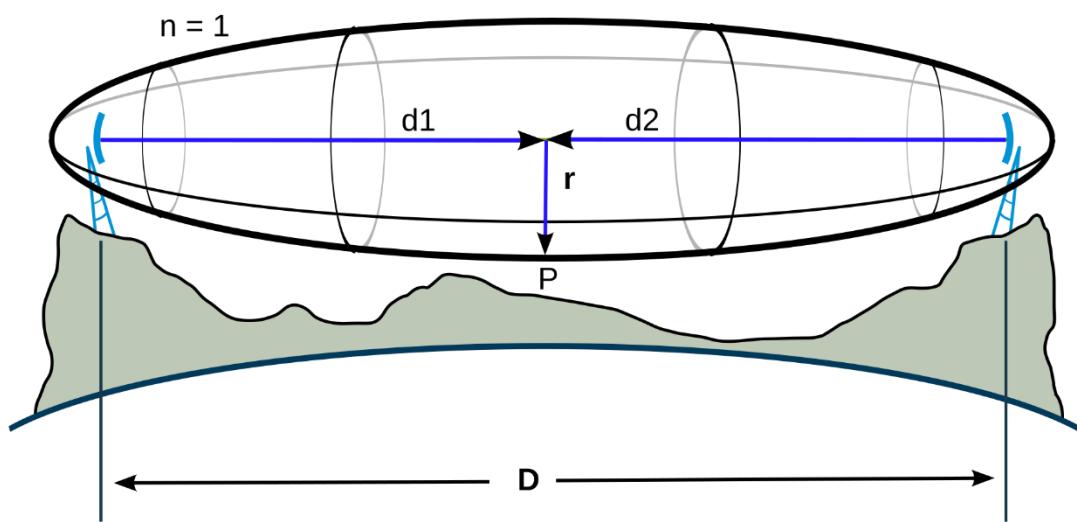


Figura 3 - Zona de Fresnel y curvatura de la Tierra. Obtenida de National Training Center. (n.d.). *The Fresnel Zone*. National Training Center

La severidad de estos efectos depende de varios factores:

- Porcentaje de obstrucción: para mantener una calidad aceptable, se recomienda que al menos el 60% de la primera zona de Fresnel esté libre de obstáculos.
- Distancia entre el transmisor y el receptor: a mayor distancia, la zona de Fresnel se amplía, aumentando la posibilidad de interferencias.
- Materiales y naturaleza del obstáculo: las superficies metálicas tienden a reflejar las ondas, mientras que materiales absorbentes, como roca, hormigón o árboles, pueden atenuarlas.

3. ESTADO DEL ARTE

Tal y como se ha explicado en el apartado anterior, los despliegues de redes LoRaWAN presentan desafíos específicos relacionados con la optimización de la cobertura y la eficiencia energética. La correcta planificación de la ubicación de nodos y gateways resulta crítica para maximizar la cobertura de la red y optimizar el consumo energético. A este respecto, el análisis de la zona de Fresnel es un concepto clave en el despliegue de redes de comunicación inalámbrica, ya que evalúa cómo los obstáculos entre el transmisor y el receptor afectan la propagación de la señal. Este análisis es fundamental en redes LoRaWAN para asegurar que los dispositivos se coloquen en ubicaciones donde la obstrucción sea mínima, lo que mejora la calidad de la señal y reduce el número de dispositivos necesarios para cubrir un área determinada.

En el ámbito de la planificación de redes de IoT, existen herramientas y modelos de simulación dedicados al análisis de cobertura en redes LoRaWAN. Estas herramientas permiten evaluar diferentes configuraciones de nodos y gateways para optimizar la eficiencia y cobertura de la red.

- **Simuladores de Cobertura:** herramientas como NS-3[14] y OMNeT++[15] han sido ampliamente utilizadas para la simulación de redes de comunicación, y algunos módulos permiten incorporar modelos de propagación específicos de LoRaWAN. Estos simuladores pueden analizar variables como la interferencia, la pérdida de señal y el alcance de los nodos, proporcionando datos precisos sobre la cobertura teórica de la red. Sin embargo, la mayoría de estos simuladores no contemplan la integración de datos de elevación y geográficos detallados, lo que limita su aplicación en terrenos complejos.
- **Herramientas topográficas:** como ArcGIS Pro[16], que permite analizar la elevación y trazar rutas, pero no simula la propagación de señales ni considera la Zona de Fresnel.
- **Herramientas de comunicación:** las herramientas de comunicación permiten evaluar la calidad y cobertura de redes ya desplegadas. Algunas de las principales herramientas en esta categoría son:

- **TTN Mapper**[17]: evalúa la cobertura de red TTN con despliegues reales, aunque no ofrece información de zonas sin dispositivos previos ni analiza la Zona de Fresnel.
 - **HeyWhatsThat**[18]: muestra visibilidad de montañas, pero no garantiza precisión en conectividad ni considera la Zona de Fresnel.
 - **Solwise**[19] y **SCADACore RF Line of Sight**[20]: permiten evaluar la línea de visión y el perfil del terreno entre dos puntos, útiles para planificación de enlaces, aunque sin análisis de ocupación de la Zona de Fresnel.
- **Simuladores de propagación de señales:** Los simuladores de propagación de señales se utilizan para modelar redes de comunicación y analizar el comportamiento de las señales en función de la configuración de la red y el entorno. Entre los simuladores más destacados se encuentran:
 - **Matlab**[21]: permite simular redes LoRaWAN, pero no incorpora análisis de terreno ni de ocupación de la Zona de Fresnel.
 - **Radio Mobile**[22]: combina datos de elevación con parámetros de transceptores para evaluar enlaces, representando la Zona de Fresnel, aunque sin calcular su ocupación.
 - **FloRa**[23] y **LoRaSim**[24]: modelan redes LoRa, sin incluir el análisis de línea de visión o de la Zona de Fresnel.

Para obtener datos de elevación precisos en simulaciones de cobertura, diversas APIs proporcionan modelos de elevación digital (DEM). Las principales API utilizadas son:

- **Google Maps Elevation API:** Proporciona datos de elevación detallados, pero tiene costes asociados y límites de peticiones, lo que la hace menos viable para simulaciones intensivas sin una estrategia de gestión de peticiones.
- **Bing Maps Elevation API:** Hasta su discontinuación el próximo 2025, esta API permitió el acceso gratuito a datos de elevación. Actualmente, Azure Maps está destinado a sustituirla en aplicaciones comerciales, aunque carece de una API específica de elevación.

- **Open Topo Data y Open-Meteo:** Estas soluciones permiten despliegues locales y auto-alojados, proporcionando una opción escalable y sin limitaciones de peticiones, aunque requieren almacenamiento significativo.

La investigación sobre las herramientas y simuladores disponibles revela que no existe una solución que combine análisis topográfico y evaluación de la Zona de Fresnel en redes IoT. La herramienta que se plantea en este proyecto busca generar mapas de cobertura mediante la obtención de datos de elevación y el análisis detallado de la Zona de Fresnel. Esta herramienta permitirá evaluar la ocupación de la Zona de Fresnel, visualizando las áreas de cobertura en función de la posición de los gateways y las características geográficas del entorno, superando así las limitaciones en herramientas existentes de planificación de redes IoT.

4. INTEGRACIÓN DEL SIMULADOR EN FRESSIM

El simulador desarrollado en este Trabajo de Fin de Grado será incorporado a una aplicación web desarrollada para la evaluación de cobertura en redes de LoRaWAN mediante el análisis de la zona de Fresnel entre nodos y gateways llamada FresSim [5]. Esta aplicación permite al usuario agregar y configurar tanto gateways como nodos sobre un mapa interactivo, de manera que se pueda seleccionar cualquier nodo y gateway para evaluar si existe cobertura directa entre ellos. El objetivo de esta aplicación es proporcionar una visualización gráfica, facilitando la interpretación de la cobertura y las posibles obstrucciones en el terreno tal y como se puede observar en la Ilustración 1.

La aplicación ha sido desarrollada utilizando una arquitectura basada en tecnologías web, combinando React y Next.js para la interfaz y el manejo de la página, y Node.js para la lógica de la aplicación y la comunicación con la API. La visualización del mapa se realiza mediante la biblioteca Leaflet, que permite generar mapas interactivos y superponer elementos como gateways y nodos en las coordenadas especificadas. Para obtener información sobre la elevación del terreno en cada punto, la aplicación utiliza Bing Maps REST Services, una API que recibe coordenadas geográficas y devuelve los datos de altitud correspondientes, lo que permite construir el perfil del terreno y evaluar la zona de Fresnel.

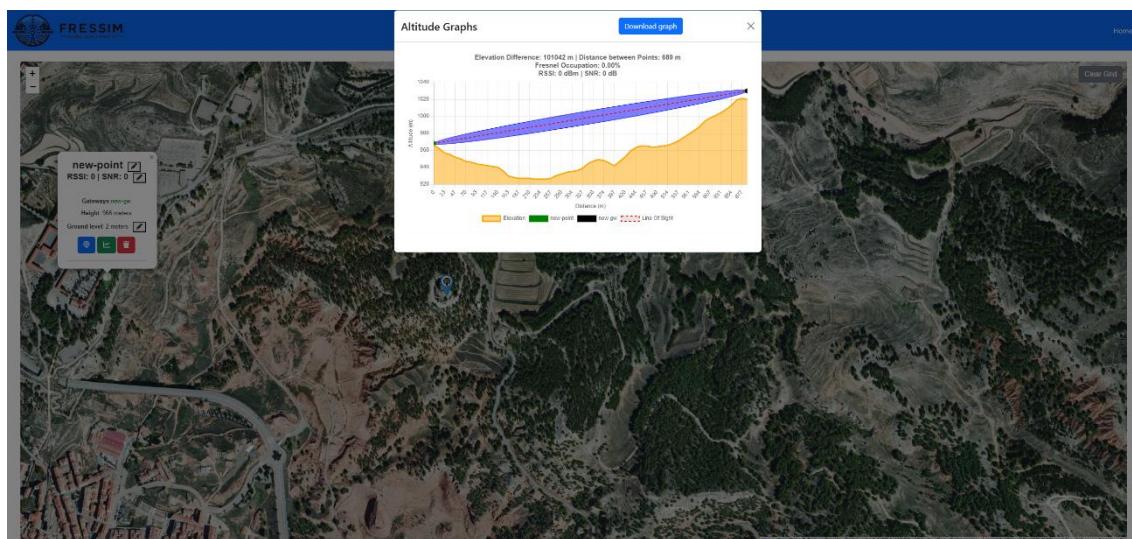


Ilustración 1 - Funcionalidad principal de FresSim.

5. SIMULADOR DE COBERTURA

En este capítulo, se detalla el funcionamiento del simulador, incluyendo el proceso de generación de mapas y la evaluación de sectores en función de los datos de elevación y la zona de Fresnel. Además, se analizan las limitaciones observadas en esta primera versión, como el alto tiempo de procesamiento, las restricciones impuestas por la API de elevaciones y la dependencia del rendimiento del cliente. Finalmente, se describen las medidas implementadas para mitigar estas limitaciones.

5.1. Generación de mapas de cobertura

El simulador de cobertura permite evaluar la calidad de la señal en redes LoRaWAN mediante un análisis visual de la cobertura en el área circundante a uno o varios gateways. A través de mapas interactivos, esta herramienta representa gráficamente las áreas con cobertura buena, pobre o inexistente, facilitando el análisis de la cobertura por parte del usuario.

El generador de mapas se ha diseñado de tal forma que el usuario solo tenga que seleccionar entre el diámetro y la resolución deseados de unas opciones de ratio preconfiguradas y la altura de los nodos. Estos valores determinan el tamaño del área de lado a lado, la densidad de sectores en los que se evaluará la cobertura y la altura sobre el nivel del suelo en la que se posicionarán todos los nodos para ese análisis en particular, respectivamente.

La Figura 4 muestra la arquitectura del generador, en la que el cliente se encargará de realizar las operaciones más ligeras, como es el cálculo de la matriz de sectores, y el servidor se encargue del cálculo del nivel de ocupación, para el que tendrá que solicitar datos de elevación a una API externa.

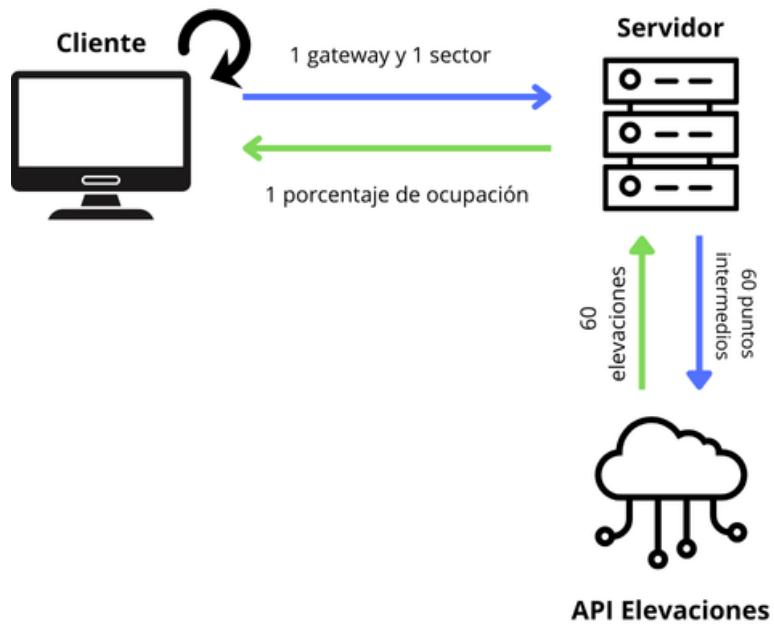


Figura 4 – Arquitectura de procesamiento individual

Una vez escogidos los valores, el cliente comienza el proceso de generación de coordenadas de los sectores que cubrirán el área seleccionada. Para la representación de cada sector en Leaflet, es necesario contar con dos puntos diagonales de cada rectángulo (sector); en este caso, se utiliza la esquina superior izquierda y la esquina inferior derecha de cada uno. Además, se calcula la coordenada central de cada sector, que servirá como el punto de referencia para medir la cobertura en relación al gateway seleccionado. Este punto central será clave en la evaluación, ya que es el lugar que definirá el valor de cobertura (buena, pobre o mala) de cada sector según los datos de elevación.

Con las coordenadas generadas, el cliente envía la ubicación del punto central de cada sector junto con el gateway seleccionado al servidor mediante una solicitud POST. En el servidor, se utiliza el mecanismo de análisis de cobertura para calcular el nivel de señal en cada sector en función de los datos de elevación y la zona de Fresnel. La respuesta obtenida se procesa en el cliente y, según el valor de cobertura calculado, el sector correspondiente en el mapa se colorea: verde para una buena cobertura, amarillo para una cobertura débil, y rojo cuando no hay cobertura. Este proceso se repite de forma individual para cada sector, hasta que todos los sectores del área de análisis han sido evaluados y coloreados (Figura 4), reflejando los resultados en el mapa de manera visual (Ilustración 2).



Ilustración 2 - Primer mapa de cobertura generado

Una limitación significativa es el tiempo requerido para la generación de los mapas de cobertura. Dado que el análisis requiere una solicitud a la API de elevaciones de Bing para cada sector, el proceso puede ser lento, especialmente en áreas de gran tamaño o cuando la resolución es alta. Además, debido a las limitaciones impuestas por la API de Bing, que permite un máximo de 50.000 peticiones diarias, el área de análisis está limitada a un máximo de 497,3 km² (que equivale a 22,37 km de ancho). Esto restringe las posibilidades de cobertura, especialmente en entornos donde se necesita analizar zonas más amplias o de mayor resolución.

Además de las limitaciones de tiempo y área de análisis, dado que el procesamiento de las coordenadas de cada sector se realiza en el cliente, se ha observado que la carga de trabajo impacta directamente en el rendimiento del dispositivo del usuario. Este procesamiento puede sobrecargar el navegador, especialmente en dispositivos con menor capacidad de procesamiento, resultando en una experiencia de usuario lenta y, en algunos casos, incluso provocando bloqueos o ralentizaciones en el navegador.

Esta dependencia del rendimiento del cliente limita la efectividad de la herramienta y su capacidad de manejar análisis de gran escala o alta resolución en todos los dispositivos. Por tanto, la experiencia de usuario puede variar significativamente en función de las

especificaciones del dispositivo, lo cual representa un reto importante para la escalabilidad y usabilidad de esta primera versión de la aplicación.

La Tabla 1 muestra el tiempo requerido para el procesamiento y el número de sectores que componen el área en función del tamaño.

Tabla 1 - Métricas de la versión original

BING API	1 km	5 km	10 km	20 km
Tiempo	0,23 min	8,90 min	36,07 min	153,23 min
Nº de sectores	100	2.500	10.000	40.000
Nº de peticiones	100	2.500	10.000	40.000

5.2. Mitigación de las limitaciones

Para solventar las carencias mencionadas se han implementado una serie de mejoras con el objetivo de mejorar el tiempo de respuesta y el uso de la API de elevaciones. Esta nueva versión incorpora mejoras significativas que optimizan el rendimiento, simplifican la configuración y amplían el área de análisis.

5.2.1. Migración de los cálculos al servidor

La lógica de generación de coordenadas y el proceso de coloreado de sectores se ha trasladado del cliente al servidor. Con esta modificación, el cliente solo recibe un vector con la información que tiene que representar, sin necesidad de realizar cálculos adicionales. Esto reduce drásticamente el tiempo de respuesta, ya que el navegador no necesita procesar el mapa en tiempo real, y permite que la herramienta dependa del rendimiento del servidor en lugar del dispositivo del usuario. Así, el cliente únicamente se ocupa de la representación visual, haciendo que la experiencia sea mucho más fluida y ágil, incluso en dispositivos con menor capacidad de procesamiento.

5.2.2. Reducción de peticiones

La principal optimización ha sido rediseñar el sistema de generación de la cuadrícula para reducir el abusivo número de llamadas a la API de elevaciones. En lugar de hacer una solicitud por cada sector del mapa (Figura 5), el servidor agrupa todas las coordenadas intermedias en un vector único. Este vector se divide en lotes de 1024 puntos, que es el número máximo permitido por la API de Bing en una sola petición (Figura 6). Los lotes se envían secuencialmente, y las respuestas se almacenan hasta completar todas las solicitudes de elevación. Además, para cumplir con la restricción de Bing de 5 peticiones por segundo, se ha establecido un intervalo de 210 milisegundos entre cada llamada, contando un margen de seguridad de 10 ms para asegurar que no se sobrepase el límite impuesto. Pese a estos ajustes, ocasionalmente se producen errores en la respuesta de la API; por ello, se ha implementado un mecanismo de detección y reinicio automático para gestionar estos fallos y garantizar la obtención de todos los datos de elevación necesarios.

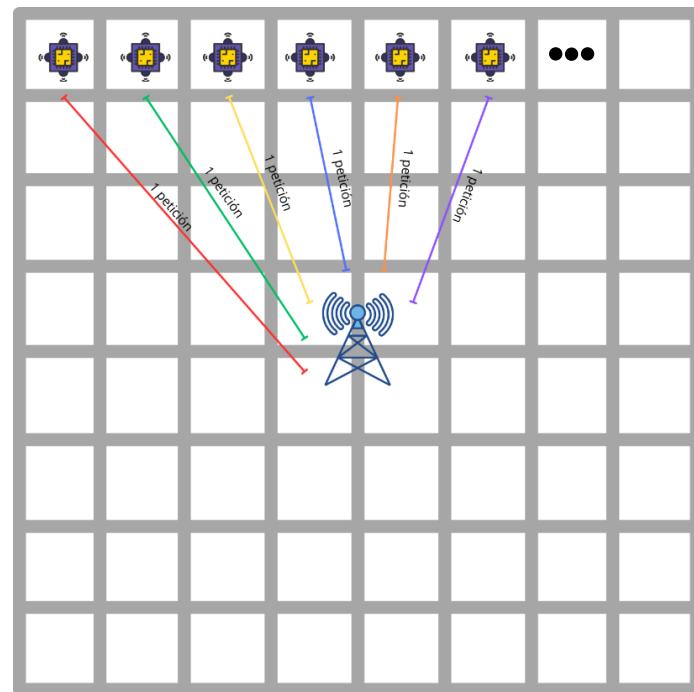


Figura 5 - Peticiones individuales

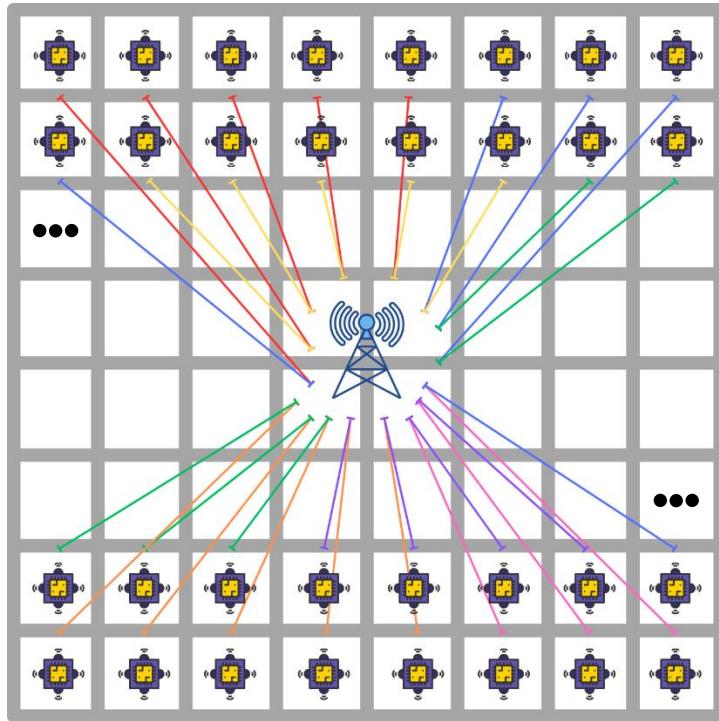


Figura 6 - Peticiones por lotes

Una vez completada la obtención de datos de elevación, el sistema asigna las elevaciones a sus sectores correspondientes y realiza los cálculos necesarios para determinar la cobertura de cada uno. En función de estos resultados, se asigna un color a cada sector, indicando visualmente la calidad de la cobertura: verde para buena cobertura, amarillo para cobertura baja y rojo cuando no hay cobertura. Finalmente, se genera un vector de respuesta, que contiene únicamente la información esencial para la representación en el cliente: las coordenadas dos esquinas de cada sector, el punto central y el color asignado.

En la Figura 7 se muestra la nueva arquitectura, destacando en rojo los cambios respecto al diseño anterior.

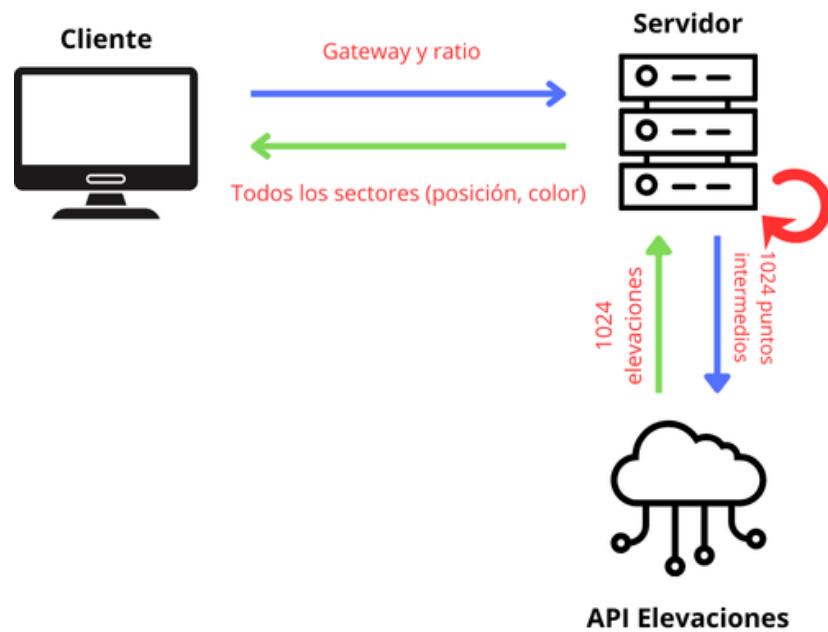


Figura 7 - Arquitectura de procesamiento en lotes

La Ilustración 3 muestra el área generada y una serie de puntos de control creados para evaluar si el algoritmo que procesa los lotes los asigna correctamente generando un mapa consistente.



5.2.3. Resultados

La Tabla 2 muestra el tiempo requerido para el procesamiento y el número de peticiones y de sectores del área en función del tamaño después de aplicar los cambios.

Tabla 2 - Métricas de la versión original mejorada

BING API	1 km	5 km	10 km	20 km
Tiempo	0,0435 min	2,81 min	4,68 min	17,44 min
Nº de sectores	100	2.500	10.000	40.000
Nº de peticiones	6	147	586	2.344

Gracias a estos cambios, ha sido posible incrementar significativamente el área máxima de análisis, alcanzando hasta 8500 km² sin exceder el límite diario de peticiones de la API. Esto implica que ahora pueden analizarse áreas hasta 17 veces mayores en comparación con la versión anterior, lo que representa una expansión notable del área de análisis.

Además, la mejora en eficiencia no solo se refleja en el área de análisis, sino también en el tiempo de procesamiento. Para áreas de 20x20 km², el tiempo requerido para completar el análisis ha disminuido drásticamente, pasando de aproximadamente 3 horas en la versión inicial a menos de 18 minutos en esta segunda versión. Esto significa que el análisis de una misma extensión de terreno ahora requiere solo el 11,4% del tiempo previamente necesario.

A pesar de las mejoras significativas de esta versión en cuanto a la eficiencia de las llamadas y el rendimiento del servidor, persiste un problema relacionado con la representación gráfica en el cliente. Para áreas de análisis extensas, el alto número de elementos que deben renderizarse en el navegador puede afectar a su rendimiento, especialmente en dispositivos de menor capacidad¹.

¹ Este problema se detalla en el punto 2 de la sección *Problemas Encontrados*

5.2.4. Fichero de configuración

Para facilitar la gestión de parámetros y configuraciones de la aplicación, se ha añadido un archivo de configuración centralizado, que contiene todos los valores modificables de la API. Estos parámetros estaban distribuidos en múltiples archivos de la aplicación, por lo que cualquier ajuste requería buscar y modificar el valor deseado en varias ubicaciones del código. Con esta nueva estructura, se centralizan todas las configuraciones, facilitando su actualización, reduciendo el tiempo de mantenimiento y minimizando el riesgo de errores por configuraciones inconsistentes.

6. OPTIMIZACIÓN DEL RENDIMIENTO

Con el fin de optimizar el simulador al máximo, se ha rediseñado la lógica del cálculo de la cobertura de los sectores.

Como se puede observar en la Figura 8, los trazados entre los sectores exteriores y el gateway cruzan sobre los otros sectores. Esto significa que la mayor parte de las peticiones a la API de elevaciones se pueden evitar realizando solo las peticiones de los sectores exteriores y reutilizando las elevaciones obtenidas.

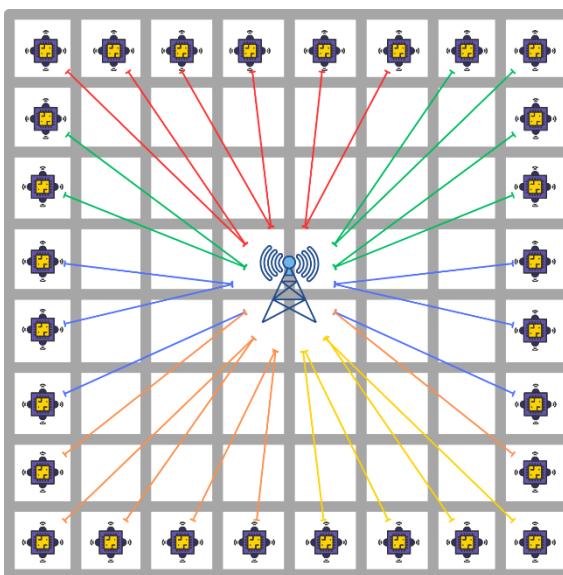


Figura 8 - Petición de sectores externos

Esta optimización ha permitido mejorar significativamente la velocidad de generación de cuadrículas y reducir aún más el número de peticiones a la API de elevaciones. Además, se han incorporado filtros y dos nuevas vistas de mapa para mejorar la visualización e interpretación de los resultados por parte del usuario.

6.1. Reaprovechamiento de datos

Una vez calculadas las coordenadas de todos los sectores, se optimiza el proceso al filtrar solo los sectores ubicados en las filas y columnas exteriores de la cuadrícula y solicitando únicamente las elevaciones de estos sectores. Después de obtener y asignar las elevaciones a los sectores exteriores, se verifica si estas pueden reutilizarse para los sectores intermedios. Esta verificación se realiza comprobando si alguno de los puntos de

los sectores exteriores está a una distancia determinada (definida en el archivo de configuración) del centro de los sectores intermedios. Cuando se cumple esta condición, las elevaciones del sector exterior (desde el punto de la coincidencia hasta el gateway) se copian para el sector intermedio.

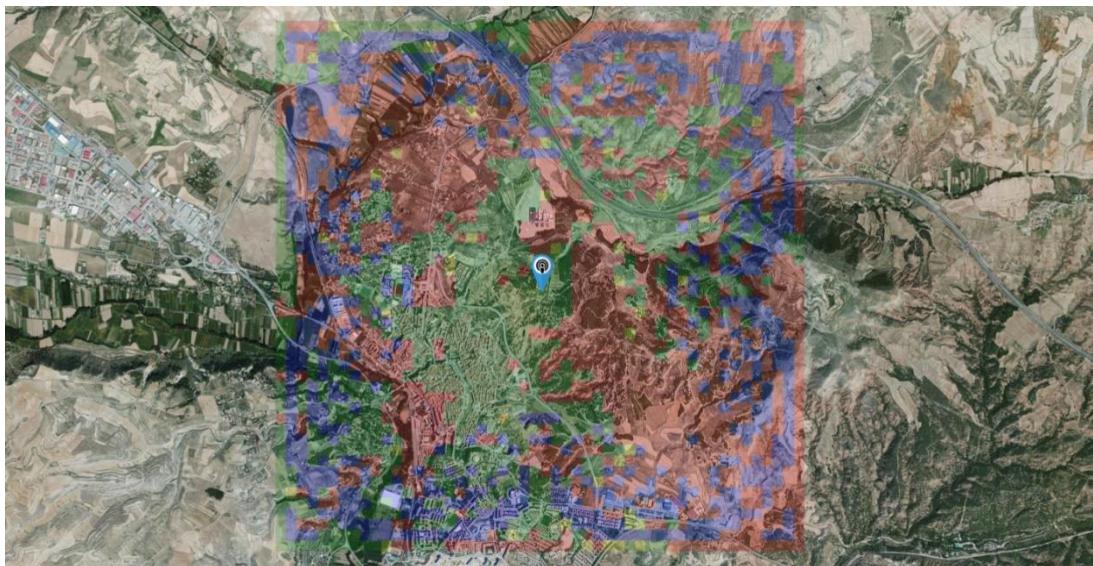


Ilustración 4 – Área generada con sectores sin valores aproximados

Una vez que todos los sectores intermedios han sido procesados, se identifica si alguno de ellos sigue sin contar con los datos de elevación necesarios (sectores azules de la Ilustración 4), y se agrupan los puntos de estos sectores en un vector único que se envía en lotes para completar el análisis. Con todas las elevaciones asignadas, se procede al cálculo de ocupación y asignación de color correspondiente para cada sector.

La Tabla 3 muestra el tiempo requerido para el procesamiento y el número de sectores que componen el área y el número de sectores que no se han encontrado mediante el método de aproximación en función del tamaño del área.

Tabla 3 - Métricas de la versión optimizada

GOOGLE API	1 km	5 km	10 km	20 km
Tiempo	0,01 min	0,06 min	0,23 min	1,09 min
Nº de sectores	100	2.500	10.000	40.000
Sectores no encontrados	0	0	0	12

Esta implementación permite reducir significativamente el tiempo de procesamiento, ya que resulta mucho más eficiente realizar un mayor procesamiento de datos en el servidor que realizar el número más elevado de peticiones requerido en la versión anterior. Esta mejora, no obstante, implica una pérdida de precisión, ya que los sectores intermedios cuentan con menos puntos de elevación y estos están basados en coordenadas aproximadas a su centro. Para maximizar la precisión, se ajusta el radio de comprobación de proximidad entre los sectores intermedios y los puntos vecinos según el tamaño del área de análisis; de esta manera, para áreas grandes se aumenta el radio de comprobación, mientras que para áreas pequeñas se reduce.

6.2. Migración de la API de elevaciones

Durante el desarrollo de la optimización desarrollada en el punto anterior, Bing REST Services dejó de funcionar debido a que las credenciales utilizadas para obtener la API key se bloquearon impidiendo su uso², lo que ha obligado a migrar hacia una API alternativa para obtener los datos de elevación del terreno. Dado que cada API presenta limitaciones y características de uso propias, resulta necesario investigar varias opciones para seleccionar la más adecuada.

Tras evaluar las opciones mostradas en la Tabla 4, se ha optado por Google Maps Platform. Aunque es un servicio de pago, ofrece un crédito mensual gratuito de 200 USD para Google Cloud, lo cual permite realizar hasta 200.000 peticiones mensuales sin coste.

² Este problema se detalla en el punto 3 de la sección *Problemas Encontrados*

Tabla 4 - Comparación de las APIs de elevaciones

	Google Maps Platform	Open-Meteo (no comercial)	Open-Meteo (comercial)	Open-Elevation	Open-Topo
Puntos por petición	512	100	100	100	100
Peticiones por segundo	Sin límite	Sin límite	Sin límite	1	1
Peticiones por minuto	6.000	600	Sin límite	Sin límite	Sin límite
Peticiones diarias	Sin límite	10.000	Sin límite	1.000	1.000
Peticiones al mes	Sin límite	Sin límite	1.000.000 – 5.000.000	Sin límite	Sin límite
Resolución	10 m	90 m	90m	30 m	A elección
Coste	0,005 USD por petición	Ninguno	29-99 EUR al mes	Ninguno	Ninguno

Para llevar a cabo esta migración de Bing a Google, se ha reconstruido el sistema de petición de elevaciones y se han ajustado varios parámetros internos. Hasta ahora, cada vez que se añadía un gateway o un nodo, el cliente hacía una petición directa a la API para obtener la elevación de ese punto. Sin embargo, debido a restricciones de seguridad en la librería de Google para Node.js³, no es posible realizar consultas de elevación directamente desde el cliente, ya que la respuesta de la API, cuando se usa sin librería, incluye un código que el cliente no tiene permiso para interpretar. Para resolver esto, se traslada este proceso al servidor, de manera que el cliente solicita la elevación al servidor y este gestiona la petición a la API de Google.

³ <https://github.com/googlemaps/google-maps-services-js>

Además, la lógica de obtención de elevaciones se simplifica gracias a la librería de Google, que gestiona los reintentos automáticamente en caso de fallo, conservando los mismos parámetros de entrada y salida que se utilizaban anteriormente. La principal optimización se logra mediante la reducción de peticiones, algo fundamental para reducir tanto los tiempos de procesamiento como los costes económicos asociados al uso de la API de Google.

6.3. Mapas

Se han introducido nuevas herramientas de personalización y análisis para mejorar la visualización de los resultados de cobertura en el mapa. Estas incluyen la posibilidad de alternar entre diferentes mapas base y la incorporación de filtros para gestionar de forma más precisa la visualización de sectores y gateways, adaptándose a las necesidades específicas de cada usuario.

6.3.1. Diferentes mapas base

Para ofrecer una visualización más completa y flexible de los resultados, se han incorporado dos nuevos tipos de mapas que pueden seleccionarse desde un botón ubicado en una esquina de la pantalla principal. Esto permite al usuario alternar entre tres opciones de visualización: una vista de imagen satelital, un mapa topográfico y un mapa en escala de grises.

Con esta funcionalidad, el usuario puede analizar los resultados de cobertura en diversos contextos, aprovechando la imagen satelital (Ilustración 5) para una selección más realista de las ubicaciones de los gateways y nodos, el mapa topográfico (Ilustración 6) para observar variaciones de elevación, y el mapa en escala de grises (Ilustración 7) para un análisis simplificado sin distracciones visuales, adaptándose así a diferentes necesidades de evaluación y optimizando la interpretación de los datos de cobertura.

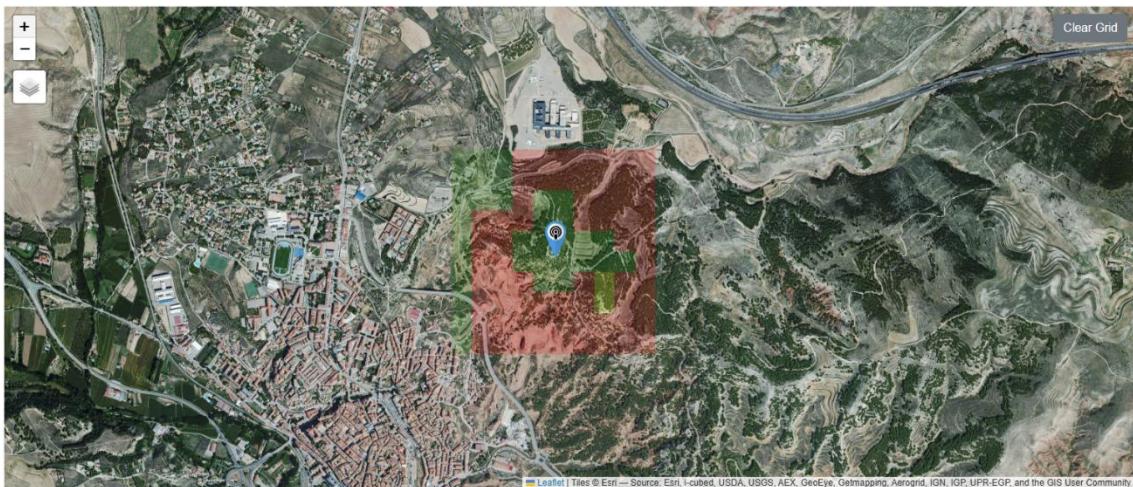


Ilustración 5 - Resultados de cobertura sobre mapa satelital

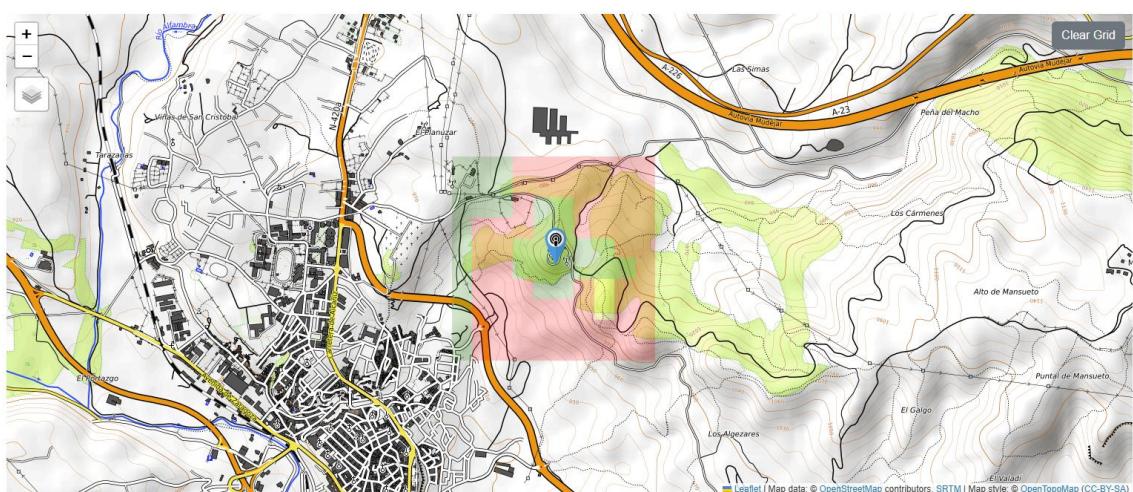


Ilustración 6 - Resultados de cobertura sobre mapa topográfico



Ilustración 7 - Resultados de cobertura sobre mapa en escala de grises

6.3.2. Filtros del mapa

Para mejorar la visualización y el análisis de los resultados, se han añadido dos filtros en el control de capas: uno para los sectores y otro para los gateways.

El filtro de sectores permite seleccionar individualmente qué sectores se desean visualizar en el mapa, lo cual es especialmente útil cuando se trabajan con múltiples áreas de análisis que pueden superponerse y complicar la interpretación de la cobertura. Con este filtro, el usuario puede activar o desactivar sectores según convenga, evitando la saturación visual y centrándose solo en los sectores relevantes.

En la Ilustración 8 se muestra como con varios gateways, la visualización de las áreas con cobertura se hace más fácil de distinguir al no mostrar los sectores sin cobertura.

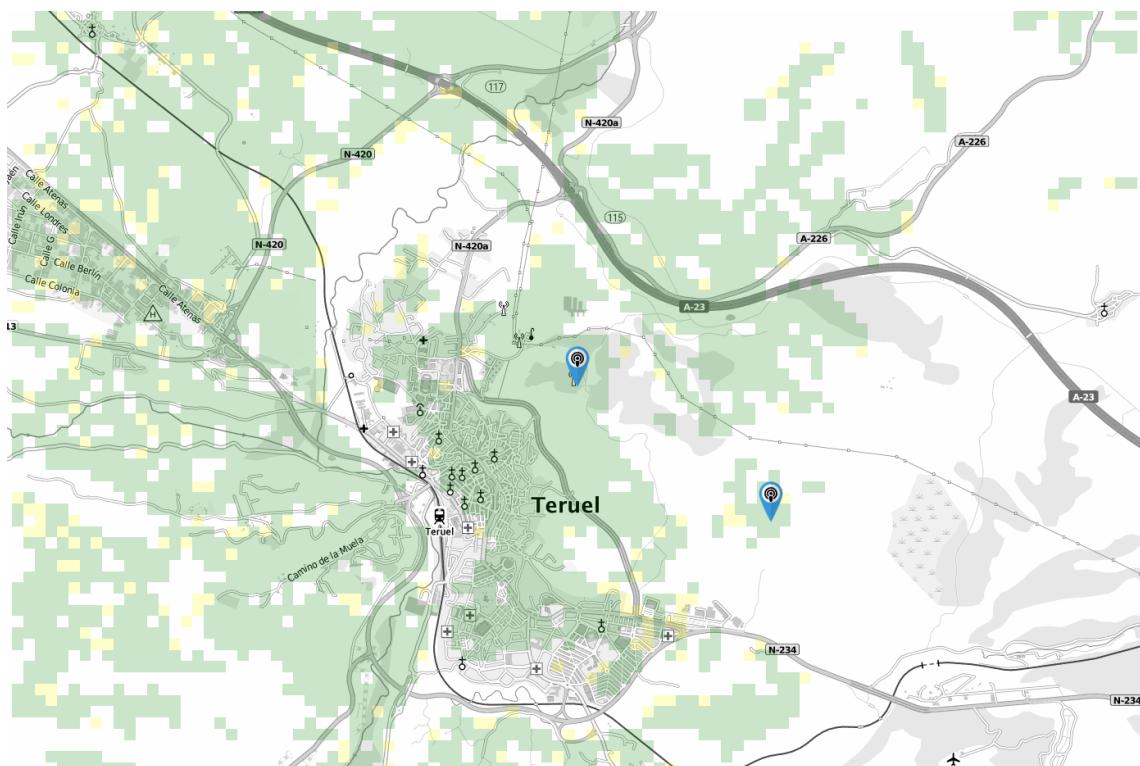


Ilustración 8 - Mapa de cobertura con filtrado de sectores activo

Por otro lado, el filtro de gateways permite escoger qué gateways se muestran en el mapa junto con sus respectivas áreas de cobertura. Esta opción resulta fundamental cuando no se desea trabajar con un gateway específico pero se quiere mantener su área

de cobertura para futuras referencias sin necesidad de eliminarla y volver a recalcularla. Al ocultar gateways y sus áreas asociadas de manera temporal, se logra una visualización más limpia y eficiente del mapa, sin perder información importante que pueda ser útil en análisis posteriores.

Ambos filtros, tanto el de sectores como el de gateways, ofrecen un nivel adicional de control y personalización, permitiendo ajustar la visualización a las necesidades específicas de cada análisis y facilitando una navegación más fluida y enfocada en la interfaz.

7. RESULTADOS

En esta sección se analizan los resultados obtenidos en las pruebas de rendimiento del simulador antes y después de las optimizaciones, se presentan las principales mejoras en la gestión de APIs dentro de la aplicación y, por último, se detalla cómo se ha desplegado la herramienta en el servidor que la va a alojar definitivamente.

Estas mejoras incluyen la implementación de un selector de modos, que permite ajustar la herramienta según la prioridad entre velocidad y precisión; el despliegue de APIs de elevaciones en un entorno local para eliminar restricciones en consultas; y un selector dinámico de APIs. Estas mejoras aumentan la flexibilidad y la escalabilidad de la herramienta, permitiendo al usuario adaptar el análisis a sus necesidades específicas.

7.1. Rendimiento

Las dos versiones de la herramienta de análisis de cobertura representan una mejora gradual en términos de eficiencia y velocidad. A medida que se implementan mejoras en cada iteración, se logran superar limitaciones técnicas significativas, optimizando la funcionalidad y reduciendo los tiempos de procesamiento.

7.1.1. Tiempos

La evolución de los tiempos mostrada en la Tabla 5 refleja una mejora que resuelve los problemas de eficiencia de la primera implementación. Las optimizaciones llevadas a cabo permiten a la herramienta manejar áreas mayores en tiempos considerablemente más cortos:

- Sin optimización: con su estructura inicial, muestra tiempos prolongados en comparación, particularmente en áreas mayores, debido al número de peticiones a la API. Dada la diferencia entre el número de puntos por petición y tiempo entre peticiones de Bing y Google, el segundo se ve perjudicado en áreas grandes.

- Optimizada: aprovechando una lógica de reutilización de datos, esta versión representa gran avance en la optimización del tiempo, permitiendo analizar áreas hasta 400 km² en tiempos menores a 1,1 minutos.

La versión optimizada no solo mejora drásticamente los tiempos, sino que también resuelve las limitaciones de la API de Bing en favor de una solución de pago (Google Maps), manteniendo un equilibrio entre el costo de operación y el rendimiento.

Tabla 5- Comparación de tiempos entre versiones

TIEMPOS	1 km	5 km	10 km	20 km
Bing (sin optimizar)	0,0435 min	2,81 min	4,68 min	17,44 min
Google (sin optimizar)	0,0347 min	0,62 min	2,67 min	11,63 min
Bing (con optimización)	-	-	-	-
Google (con optimización)	0,01 min	0,06 min	0,23 min	1,09 min

7.1.2. Número de peticiones

La optimización de las peticiones API mostrada en la Tabla 6 es uno de los mayores avances logrados en la herramienta:

- Sin optimizar: la implementación de lotes permite una mejora drástica en la eficiencia de las peticiones respecto a su primera implementación, pero aún así el número de peticiones sigue siendo alto para áreas grandes.
- Tras la optimización: la reestructuración de la lógica para utilizar solo las elevaciones externas y reutilizar los datos para los sectores intermedios lleva la optimización a su punto máximo, reduciendo las peticiones en un 98,3% para áreas grandes (98% en Bing), sacrificando precisión.

La evolución de la herramienta muestra cómo una mejora continua en la lógica de peticiones y reutilización de datos impacta directamente en el rendimiento. Con las optimizaciones, se logra minimizar al máximo el uso de peticiones, permitiendo la

cobertura de grandes áreas de hasta 400 km² con solo 96 peticiones, algo inimaginable en la primera versión. Esta optimización no solo beneficia el tiempo de procesamiento, sino que también reduce significativamente los costes asociados a las peticiones en la API de Google, haciendo la herramienta más escalable y viable en términos económicos y de rendimiento.

Tabla 6- Comparación de número de peticiones entre versiones

PETICIONES	1 km	5 km	10 km	20 km
Bing (sin optimizar)	6	147	586	2.344
Google (sin optimizar)	12	293	1.172	5.688
Bing (con optimización)	4	12	24	48
Google (con optimización)	6	23	47	96

7.1.3. Fluctuaciones entre las versiones sin optimizar y optimizada

Al analizar los sectores verdes (buena cobertura), amarillos (cobertura pobre) y rojos (sin cobertura) en distintas áreas (Tablas 7, 8, 9 y 10), se observa una fluctuación en la asignación de sectores entre versiones, reflejando inconsistencias en la representación de cobertura tras la optimización. Estas fluctuaciones se deben al método en que se reabusechan las elevaciones ya obtenidas y la menor precisión en el cálculo del porcentaje de ocupación de los sectores más cercanos al gateway.

Para áreas de 1 km² (Tabla 7), los cambios en la asignación de sectores son relativamente pequeños, pero aun así, la versión optimizada introduce 8 resultados distintos. Esta diferencia supone un 8% del total de sectores, que es notable en un área pequeña, ya que indica que los resultados de cobertura no son totalmente coherentes con la versión anterior.

Tabla 7 - Variaciones de la versión optimizada en áreas de 1x1 km

1 KM ²	Sectores verdes	Sectores amarillos	Sectores rojos
Sin optimizar	57	2	41
Optimizada	61	1	38
Sectores distintos	4	1	3

En el análisis de áreas de 25 km² (Tabla 8), las inconsistencias en la asignación de sectores continúan siendo evidentes, y la versión optimizada presenta 24 resultados distintos en total. A nivel porcentual, esta diferencia representa un 0,96% del total de sectores, lo que sigue siendo una desviación leve en una cuadrícula de 2.500 sectores. Sin embargo, el hecho de que se presenten alteraciones tanto en la cantidad de sectores con buena cobertura como en los sectores con cobertura moderada evidencia una tendencia al aumento de estas variaciones conforme aumenta el área.

Tabla 8 – Variaciones de la versión optimizada en áreas de 5x5 km

25 KM ²	Sectores verdes	Sectores amarillos	Sectores rojos
Sin optimizar	1.037	89	1.374
Optimizada	1.047	77	1.376
Sectores distintos	10	12	2

Al aumentar el área a 100 km² (Tabla 9), las diferencias se intensifican. En esta cuadrícula, la versión optimizada introduce 276 variaciones. En términos porcentuales, esto representa un 2,76% del total de sectores, indicando que, conforme aumenta el área de análisis, la versión optimizada presenta cada vez más inconsistencias con respecto a la versión anterior. Estos cambios aun no son preocupantes porque en áreas de análisis de este tamaño, una desviación del 2,76% compromete poco la fiabilidad en las interpretaciones de cobertura.

Tabla 9 - Variaciones de la versión optimizada en áreas de 10x10 km

100 KM ²	Sectores verdes	Sectores amarillos	Sectores rojos
Sin optimizar	3.473	396	6.131
Optimizada	3.611	303	6.086
Sectores distintos	138	93	45

Finalmente, en áreas de 400 km² (Tabla 10), la versión optimizada presenta la mayor desviación, introduciendo 4.928 sectores distintos. Esta variación representa un 12,32% del total de sectores en un área de 40.000 sectores, lo cual constituye una variación considerable que puede afectar el análisis de cobertura en cuadrículas de esta magnitud.

Tabla 10 - Variaciones de la versión optimizada en áreas de 20x20 km

400 KM ²	Sectores verdes	Sectores amarillos	Sectores rojos
Sin optimizar	8.624	1.257	30.119
Optimizada	11.088	1.097	27.815
Sectores distintos	2.464	160	2.304

En conclusión, aunque tras la optimización se muestran mejoras sustanciales en tiempo y eficiencia de peticiones, se introducen alteraciones que disminuyen la precisión en la cobertura representada en las cuadrículas, especialmente en áreas grandes.

Las siguientes ilustraciones muestran las diferencias presentadas entre la versión sin optimizar (Ilustración 9) y la optimizada (Ilustración 10) de la herramienta:



Ilustración 9 - Cobertura de Teruel con nodos de comprobación (original)

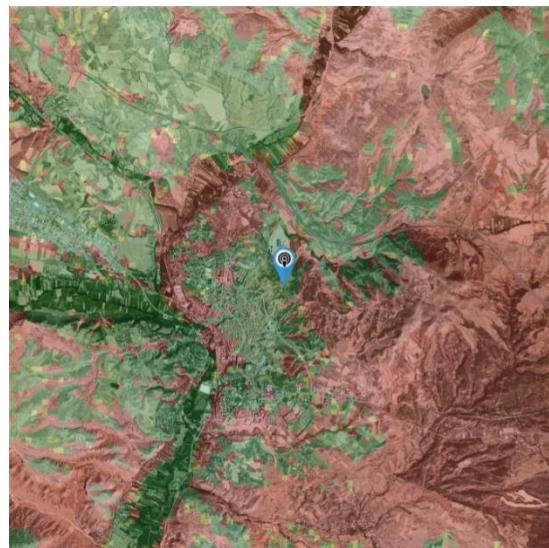


Ilustración 10 - Cobertura de Teruel (optimizada)

7.2. APIs de elevaciones

En este apartado se aborda la implementación de una serie de mejoras para el uso de APIs de elevaciones dentro de FresSim. Se describen las estrategias adoptadas, como el uso de una API self-hosted para evitar restricciones de consultas, la incorporación de un selector de API y otro del modo generación para adaptar el análisis a diferentes necesidades y un sistema de estimación de peticiones que optimiza la planificación de recursos. Estas innovaciones fortalecen la capacidad del simulador para realizar análisis precisos, escalables y ajustados a las demandas del proyecto.

7.2.1. Selector de modo

Dado que la versión optimizada de la herramienta muestra un margen considerable de fluctuaciones al trabajar con áreas extensas, mientras que la primera versión, aunque más lenta, mantiene una alta precisión en sus resultados, se ha implementado un selector de modos que permite ajustar la herramienta según las necesidades de cada análisis. Este selector brinda la opción de trabajar en "modo precisión", correspondiente a la versión sin optimizar, o en "modo velocidad", que hace uso de las optimizaciones.

Tal y como se aprecia en la Ilustración 11, para mejorar la usabilidad y garantizar que el usuario comprenda las diferencias entre estos modos, se ha añadido un botón sobre el mapa que despliega una ventana modal. Esta ventana explica el funcionamiento de cada opción, detallando el compromiso entre velocidad y precisión que caracteriza cada modo, de modo que el usuario pueda seleccionar el ajuste que mejor se adecúe a los requisitos específicos de su análisis y al tamaño del área de cobertura.

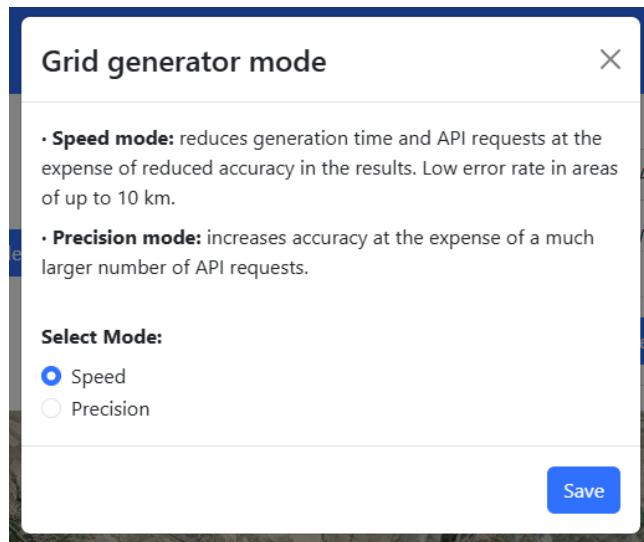


Ilustración 11 - Selector de modo

7.2.2. APIs de elevaciones self-hosted

Con el objetivo de superar las limitaciones de consulta de las APIs públicas para áreas de gran extensión, se ha evaluado la implementación de alternativas de código abierto que permitan eliminar esta restricción. Tras una primera fase de investigación, se seleccionaron tres APIs de código abierto: Open-Elevation, Open-Meteo y Open Topo Data. Todas ellas permiten un despliegue sencillo mediante Docker, aunque para esta implementación se optó por Open-Meteo y Open Topo Data debido a su documentación más detallada.

Para que estas APIs operen en un entorno propio, requieren datasets específicos de elevaciones. Existen varias opciones gratuitas para estos datasets, entre las que destacan Copernicus[35] perteneciente a la ESA (European Space Agency) y SRTM[36] de la NASA (National Aeronautics and Space Administration). Tras la descarga y las pruebas iniciales, se descartó Open-Meteo debido al gran tamaño de los datos de Copernicus, que

ocupa más de 600 GB en su versión de resolución de 30 metros, además de que la versión predeterminada de 90 metros es insuficiente para analizar áreas pequeñas con precisión. Además, Open-Meteo no ofrece en su documentación un método claro para integrar otros datasets, por lo que se ha optado por mantenerla únicamente como API pública.

Por su parte, Open Topo Data no incluye datasets preinstalados, pero su documentación facilita la incorporación de estos, admitiendo además una variedad de formatos y fuentes. Esta flexibilidad ha permitido mantener Open Topo como la opción seleccionada para el despliegue local. Con el fin de integrarla en FresSim, se descargaron dos conjuntos de datos: SRTMGL1[37] (datos globales con una resolución de 30 metros) y EU-DEM[38] (cobertura europea a 25 metros de resolución). Estos archivos ocupan aproximadamente 100 GB para SRTM y 23 GB para EU-DEM. En el caso de SRTM, si se desea optimizar la velocidad de lectura, se puede descomprimir, incrementando su tamaño a cerca de 350 GB.

El uso de Open Topo en un entorno local elimina las restricciones de peticiones diarias y permite ajustar el número de puntos por consulta. Esto hace posible procesar áreas de análisis sin limitaciones de tamaño, lo cual es fundamental para FresSim. Tras varias pruebas con distintas configuraciones de número de puntos por petición, se determinó un límite óptimo de 2.000 puntos por consulta. Además, Open Topo utiliza el mismo algoritmo de codificación en la URL que Google para gestionar grandes volúmenes de puntos en una sola petición.

Estas mejoras permiten que el simulador realice análisis precisos sin las limitaciones impuestas por APIs públicas, facilitando tanto la escalabilidad en el análisis de grandes áreas como la optimización de recursos y tiempos de consulta.

Tabla 11 - Métricas de tiempo de Open Topo en función del modo

	1 KM ²	25 KM ²	100 KM ²	400 KM ²
Modo velocidad	0,15 mins	0,74 mins	1,43 mins	3,59 mins
Modo precisión	0,35 mins	8,60 mins	31,98 mins	118,16 mins

La Tabla 11 de tiempos en el despliegue de Open Topo ilustra el impacto de una API sin restricciones de consultas diarias, que permite analizar áreas significativamente mayores que en las versiones anteriores que empleaban Bing o Google Maps. En modo velocidad, los tiempos de análisis son muy superiores a los de Bing o Google; en modo precisión, aunque el tiempo es considerablemente mayor, elimina la restricción de tamaño, lo que lo hace ideal para áreas grandes que requieren datos más precisos. Esta configuración permite al simulador ofrecer un análisis de cobertura preciso sin las limitaciones de las APIs públicas, maximizando así su alcance y flexibilidad en función de las necesidades del proyecto.

7.2.3. Selector de API

Considerando las diferencias entre las APIs de elevaciones empleadas a lo largo del proyecto, se ha incluido un selector que permita al usuario elegir la API más adecuada a sus necesidades y preferencias tal como se muestra en la Ilustración 12. Algunas de estas opciones, ya sean gratuitas o de pago, requieren una API key, por lo que se han agregado campos específicos para que el usuario pueda ingresar sus propias credenciales. Esta medida permite distribuir la carga entre múltiples credenciales y otorga al usuario la flexibilidad de gestionar su propio uso y cuota, evitando que una sola credencial soporte toda la demanda de la aplicación. Además, este enfoque abre la posibilidad de utilizar opciones como Google Maps Platform sin que ello represente un coste adicional para el proyecto, ya que cada usuario puede gestionar el pago o uso de sus peticiones.

De forma predeterminada, la aplicación selecciona la API de Open-Topo como fuente de elevaciones, ya que es pública, no requiere credenciales y no tiene coste alguno para el usuario, lo que la convierte en una opción accesible y económica para análisis estándar.

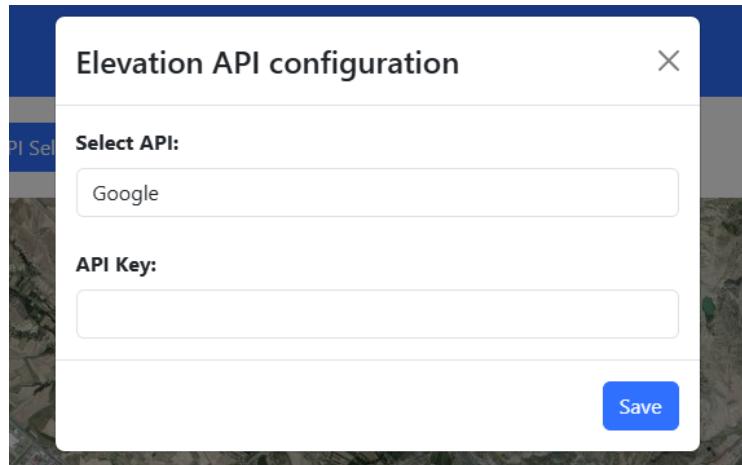


Ilustración 12 - Selector de API de elevaciones

7.2.4. Estimación de peticiones

Dado que cada API presenta limitaciones específicas, como el número máximo de peticiones permitidas por día o costes asociados al uso intensivo, se ha implementado un sistema de estimación de peticiones. Este sistema añade un campo en la interfaz que muestra una estimación en tiempo real del número de peticiones que se realizarán a la API seleccionada, considerando los parámetros actuales configurados por el usuario para la generación de resultados.

La estimación se ajusta dinámicamente según los ajustes seleccionados en el generador, como el número de puntos intermedios o el tamaño del área de análisis. De este modo, se puede prever el impacto de los parámetros establecidos sobre la API en uso. Al recibir esta información antes de proceder, se pueden ajustar los parámetros para minimizar costes o evitar superar los límites de consulta impuestos por las APIs.

Esta función aporta una transparencia adicional sobre el impacto del uso de cada API, permitiendo a los usuarios gestionar mejor tanto los recursos económicos como técnicos en su aplicación de análisis.

7.3. Despliegue en el servidor

Una vez completado el proceso de desarrollo, se desplegó la nueva versión de FresSim⁴ en el servidor (entorno definitivo), reemplazando la versión previa que había estado en uso. Aunque durante las pruebas en el entorno de desarrollo no se detectaron errores, al implementarlo en el servidor surgieron problemas al calcular áreas de gran extensión. Esto generó fallos inesperados que fueron solucionados mediante ajustes en la configuración del servidor, particularmente en los parámetros de timeout del proxy inverso.⁵.

Tras resolver este inconveniente y poder realizar las pruebas de rendimiento, los tiempos de ejecución mejoraron significativamente, gracias en gran parte a la mayor velocidad del servidor al gestionar las peticiones y procesar datos mediante el despliegue de Open Topo Data en el mismo. Esto permitió optimizar la generación de resultados y garantizó una experiencia más ágil para el usuario final.

Los datos proporcionados en las tablas reflejan una notable mejora en los tiempos de ejecución al comparar el entorno de desarrollo local con el servidor en ambos modos, velocidad y precisión.

En el modo velocidad, mostrado en la Tabla 12, la disminución de los tiempos es especialmente evidente para áreas más grandes. Por ejemplo, para 400 km², el tiempo de procesamiento en el servidor se redujo de 3,59 minutos a 2,71 minutos, lo que representa una mejora del 24%. Esta optimización se debe a la mayor capacidad del servidor para procesar peticiones y gestionar cálculos con eficiencia.

Tabla 12 - Comparación de tiempos en modo velocidad entre servidor y local

VELOCIDAD	1 km ²	25 km ²	100 km ²	400km ²
Local	0,15 mins	0,74 mins	1,43 mins	3,59 mins
Servidor	0,07 mins	0,35 mins	0,82 mins	2,71 mins

⁴ Disponible en <http://initiot.unizar.es/fressim>

⁵ Este problema se detalla en el punto 6 de la sección *Problemas Encontrados*

En el modo precisión, mostrado en la Tabla 13, las diferencias son aún más significativas, especialmente para áreas extensas. Para un área de 400 km², el tiempo pasó de 118,16 minutos en local a 67,99 minutos en el servidor, lo que supone una mejora cercana al 42%.

Tabla 13 - Comparación de tiempos en modo precisión entre servidor y local

PRECISIÓN	1 km ²	25 km ²	100 km ²	400km ²
Local	0,35 mins	8,60 mins	31,98 mins	118,16 mins
Servidor	0,16 mins	4,15 mins	16,58 mins	67,99 mins

En ambos casos, la implementación en un entorno más potente ha demostrado ser un paso crucial para mejorar el rendimiento general de FresSim, haciendo que la herramienta sea más eficiente y viable para análisis de áreas grandes y complejas.

8. PROBLEMAS ENCONTRADOS

En esta sección se detallan las principales dificultades que han aparecido durante el desarrollo del proyecto destacando las limitaciones técnicas y se analiza cómo estos problemas han llevado a implementar opciones adicionales en la herramienta para mejorar su eficiencia y adaptabilidad en diferentes entornos.

8.1. Fallos de petición

En ciertas ocasiones, las solicitudes enviadas a la API de Bing fallaban por razones externas y ajenas a la aplicación. En la primera versión, estos fallos ocurrían con poca frecuencia y su impacto era limitado, ya que solo afectaban a un sector aislado del mapa, sin comprometer la integridad general de la visualización.

Sin embargo, en la segunda versión del sistema, la estrategia cambió para reducir el consumo de recursos: se optó por obtener datos de elevación de varios sectores a la vez. Esta modificación, aunque beneficiosa, aumentó el riesgo, ya que un fallo en una única solicitud a la API resultaba en un fallo en cadena de todos los sectores restantes, generando un mapa incompleto y distorsionado que no reflejaba adecuadamente la cobertura real.

Para solucionar este problema, se implementó un mecanismo de detección y repetición de solicitudes fallidas a la API. Este sistema permite identificar automáticamente los fallos en la comunicación con la API y repetir la solicitud fallida hasta un máximo de X intentos, donde X es un valor ajustable que puede definirse en el archivo de configuración de la aplicación. Esta funcionalidad de repetición asegura que, aunque se produzca un fallo, el sistema pueda intentar nuevamente obtener los datos de elevación sin interrumpir el flujo completo.

Es cierto que la repetición de solicitudes implica un mayor número de peticiones a la API, lo que puede considerarse un uso intensivo de los recursos. Sin embargo, se concluyó que esta es la opción más eficiente en términos de consumo de recursos. Sin el mecanismo de repetición, un único fallo en la petición original provocaría la necesidad de reiniciar el proceso completo, lo que provocaría un consumo aún mayor de peticiones a la API y un

desperdicio de tiempo y peticiones considerable, afectando la experiencia de uso y la fiabilidad del sistema.

8.2. Rendimiento en cuadrículas grandes

Una vez que se genera el área de cobertura y se visualiza en el mapa, el rendimiento del navegador experimenta una reducción significativa, resultando en episodios frecuentes de "lag" o retraso en la respuesta. Esto se debe a que, para ciertas resoluciones de visualización, como la de 50 metros por sector en un área de 400 km², el número de sectores a representar asciende a unos 160.000. Cada uno de estos sectores debe ubicarse correctamente en el mapa, y cada vez que el usuario realiza una acción de desplazamiento o cambio de zoom, es necesario recalcular y reposicionar todos estos elementos gráficos.

Se han realizado pruebas en múltiples navegadores para descartar que el problema sea específico de un entorno en particular. Sin embargo, el mismo fallo se ha manifestado de manera consistente en todos ellos, sin diferencias apreciables en cuanto al rendimiento. En el lado del cliente, el navegador no lleva a cabo cálculos adicionales más allá de los necesarios para que Leaflet, la biblioteca de mapas, genere la cuadrícula en función de las coordenadas predefinidas que el servidor ya ha calculado y proporcionado.

Por lo tanto, tras diversas pruebas y análisis, se ha llegado a la conclusión de que este problema de rendimiento no tiene una solución factible dentro de las limitaciones actuales de los navegadores y de la capacidad de procesamiento requerida para manejar una cuadrícula de esta densidad.

8.3. La API de elevaciones de Bing Maps dejó de funcionar

Durante la implementación de la tercera versión de la aplicación, surgió un problema inesperado cuando la API de elevaciones de Bing comenzó a devolver errores de código 403 en todas las peticiones. Tras revisar la documentación y los registros, se identificó que este error se debía a una prohibición de las credenciales utilizadas para acceder a la API, lo que hacía que la clave API resultara inválida. Al investigar más a fondo, se descubrió que esta denegación había sido originada en la cuenta de Microsoft utilizada para obtener dichas credenciales, la cual había sido bloqueada por Unizar por razones desconocidas. Esto representaba un obstáculo significativo, ya que, debido a la próxima

retirada de Bing Maps REST Services programada para junio de 2025, no es posible crear nuevas cuentas ni obtener nuevas claves API para estos servicios. Como resultado, la cuenta bloqueada imposibilitaba la generación de una nueva clave para continuar con la integración en Bing.

La solución que ofrece Microsoft (propietario de Bing) a este problema es utilizar Azure Maps, que es la plataforma que sustituirá a Bing REST Services. El problema de esta plataforma es que en mayo de 2023 se eliminó la API de elevaciones, dando como solución utilizar la de Bing, que a su vez será eliminada en junio de 2025.

Este bloqueo forzó la migración de la aplicación hacia Google Maps, lo que implicó un cambio integral en el sistema de peticiones y en los parámetros utilizados. En el caso de Google Maps, el número máximo de puntos que se pueden solicitar por petición es de 512, la mitad de los 1024 que permitía Bing Maps, lo cual dobló el número de solicitudes necesarias para cubrir áreas de tamaño equivalente. Sin embargo, a pesar del aumento en la cantidad de peticiones, el rendimiento general del sistema mejoró. Mientras que Bing permitía un máximo de 5 peticiones por segundo con un límite diario de 50.000, Google Maps permite hasta 100 peticiones por segundo y no impone un límite diario, lo que proporciona una mayor flexibilidad y rapidez en la generación de mapas de cobertura.

No obstante, el cambio a Google Maps también introdujo nuevas limitaciones. A diferencia de Bing Maps, que no generaba costes por su uso, los servicios de Google tienen un coste asociado: cada 1000 peticiones generan un cargo de 5 USD. La cuenta dispone de un crédito mensual de 200 USD que se renueva cada mes, pero una vez agotado, cualquier solicitud adicional será facturada. Con la integración de Google Maps, se ha eliminado la limitación en cuanto a la extensión máxima de la cuadrícula, pero la operación de la aplicación ahora está condicionada por un factor económico, un aspecto que antes no estaba presente y que requerirá una gestión cuidadosa para evitar costes adicionales inesperados.

8.4. Perdida de precisión con el modo velocidad

En el modo velocidad, los sectores externos utilizan el número completo de puntos de elevación, que por defecto es de 60 (ajustable en la configuración). Sin embargo, como se muestra en las Figuras 9 y 10, cuando se reutilizan estas elevaciones para los sectores

más cercanos al gateway, el número de puntos de elevación se reduce progresivamente, ya que solo se aprovecha una parte de los datos del sector externo.

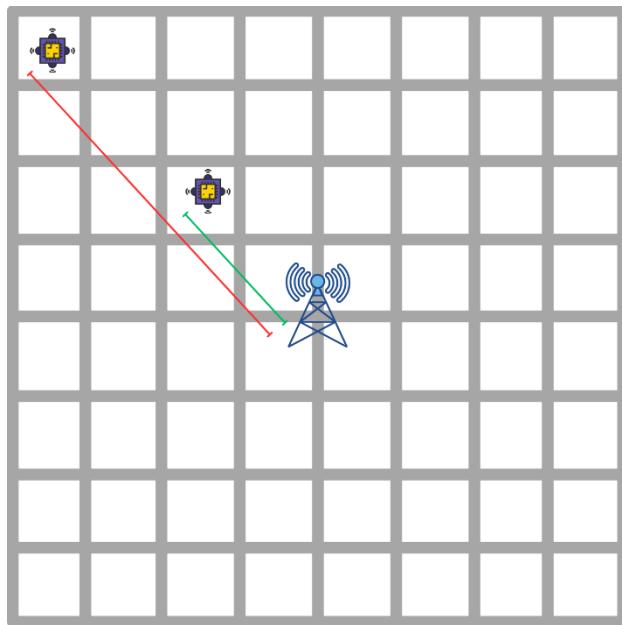


Figura 9 - Ejemplo de área de análisis por sectores

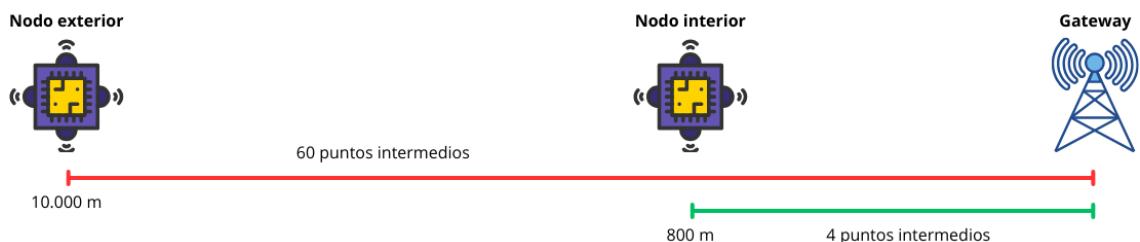


Figura 10 - Número de puntos intermedios en función de la distancia

Una solución para mejorar la precisión es incrementar el número de puntos intermedios entre el nodo exterior y el gateway, aunque esto también aumenta el número de peticiones a la API de elevaciones. Por ejemplo, al utilizar la API de Google para calcular un área de 400 km² con 60 puntos intermedios, se requieren 96 peticiones; si se aumentara a 100 puntos intermedios, el número de peticiones subiría a 158, lo que tendría implicaciones de coste y tiempo.

Dado que el incremento en el número de peticiones es moderado, se recomienda elevar el número de puntos de elevación a 100 o 120 en áreas grandes cuando se utiliza el modo

velocidad. Sin embargo, debido a las limitaciones de las APIs públicas, esta ampliación podría restringir el tamaño de las áreas analizadas. Para mayor flexibilidad, se ha añadido un selector de puntos intermedios en el modal del generador, permitiendo ajustar este parámetro según las necesidades del análisis.

8.5. Espacio en disco necesario

Aunque en un principio se consideró la API de Open-Meteo como opción para proporcionar datos de elevación en la herramienta FresSim, se descartó su uso para la implementación local debido al gran tamaño de los datasets necesarios. Open-Meteo requiere datos de alta resolución para ofrecer resultados precisos, y los archivos de elevación de Copernicus que emplea, en su versión de 30 metros, ocupan más de 600 GB. Este volumen de datos plantea desafíos logísticos y de almacenamiento, especialmente en servidores locales que deben gestionar grandes áreas de análisis sin comprometer la rapidez de procesamiento y el rendimiento general de la aplicación.

La descarga y gestión de datasets de este tamaño no solo implican un esfuerzo adicional en términos de infraestructura, sino que también afectan la escalabilidad y la sostenibilidad del proyecto en entornos donde los recursos de almacenamiento son limitados. Por esta razón, se ha decidido mantener Open-Meteo solo como una opción de API pública para usuarios que no dispongan de credenciales para otras APIs, mientras que para la implementación local de FresSim se seleccionó Open Topo.

Aunque Open Topo requiere menos espacio de almacenamiento, esto sigue representando un problema para desplegar la aplicación en entornos con poca disposición de almacenamiento y genera sobre el tamaño del área de análisis una dependencia del almacenamiento del servidor o una limitación del tamaño del área por tener que utilizar APIs con limitaciones.

A pesar de que Open Topo reduce significativamente el espacio necesario en comparación con otras APIs de elevación, aún plantea un desafío en entornos con almacenamiento limitado. La carga de datos de elevación necesarios para operar con precisión en FresSim demanda una infraestructura de servidor con capacidad suficiente para almacenar y gestionar estos archivos. Esta dependencia del almacenamiento impone restricciones sobre el tamaño del área de análisis que se puede procesar: en entornos

donde el espacio de almacenamiento es insuficiente, la aplicación debe restringir el tamaño de las áreas de análisis o recurrir a APIs con limitaciones de consultas para áreas más grandes. Esto genera un equilibrio delicado entre la precisión de la cobertura y las capacidades de infraestructura del servidor, afectando la flexibilidad del despliegue en diferentes entornos.

8.6. Peticiones pesadas

Tras desplegar la aplicación en el servidor, se detectó un problema recurrente al intentar analizar áreas de gran tamaño: las peticiones fallaban con un “Error 400 – Bad Request”. Inicialmente, se asumió que el problema podría estar relacionado con el tamaño de la respuesta del servidor, por lo que se implementaron diversas soluciones para mitigarlo. Entre estas soluciones se incluyó la paginación de datos, el uso de streams para manejar grandes volúmenes de información y la compresión de las respuestas antes de enviarlas al cliente. Sin embargo, ninguna de estas estrategias logró resolver el problema.

Al profundizar en la investigación, se observó un patrón consistente en los fallos: todas las peticiones fallaban exactamente 60 segundos después de ser enviadas al servidor, arrojando el mismo error. Esto llevó a la hipótesis de que el problema era un timeout en lugar de un error en el tamaño de la respuesta. En consecuencia, se realizaron ajustes para aumentar los límites de timeout en distintos componentes de la aplicación, incluyendo el cliente y la configuración de Next.js. No obstante, estos cambios tampoco resolvieron el problema.

Un análisis más detallado reveló que, aunque el cliente recibía el error de timeout, el servidor continuaba procesando la petición en segundo plano, lo que indicaba que el fallo no residía en la aplicación en sí. Finalmente, se identificó que el origen del problema estaba en la configuración del proxy NGINX, que gestionaba las solicitudes hacia el servidor. Ajustar los parámetros de timeout de NGINX permitió resolver el problema de forma definitiva, habilitando el análisis de áreas de gran tamaño sin interrupciones.

9. CONCLUSIÓN

El presente Trabajo de Fin de Grado ha desarrollado un simulador para la optimización de despliegues de redes LoRaWAN, centrado en la mejora de la planificación de redes IoT mediante la integración de análisis geoespacial y modelado de la Zona de Fresnel. A lo largo del proyecto, se ha implementado una herramienta capaz de generar mapas de cobertura precisos en función de las características topográficas del terreno, utilizando APIs de elevación y análisis por sectores. Esta herramienta proporciona a los usuarios una visión de la cobertura de red, facilitando decisiones informadas en la ubicación de nodos y gateways en entornos complejos.

La herramienta se ha desarrollado en diferentes iteraciones, en cada una de las cuales se resolvieron problemas específicos de rendimiento, precisión y eficiencia en el número de peticiones a las APIs de elevación. En cada iteración, se optimizó el uso de recursos y se redujeron significativamente los tiempos de procesamiento, alcanzando finalmente un modelo escalable y flexible que permite analizar áreas de cientos de kilómetros en pocos minutos. La última versión implementa un sistema de reutilización de datos y un selector de modos que permite al usuario ajustar la herramienta según sus necesidades de velocidad o precisión, ofreciendo opciones adaptables para distintas aplicaciones de IoT.

La migración hacia una API de pago (Google Maps) debido a la discontinuación de Bing Maps ha sido un reto importante, ya que puede generar costes adicionales y requerimientos de gestión de API keys. Sin embargo, la flexibilidad y el rendimiento de la herramienta se han beneficiado considerablemente con esta integración. En paralelo, se ha implementado la opción de utilizar Open Topo en local, eliminando las restricciones de las APIs públicas y permitiendo un despliegue escalable para análisis a gran escala sin incurrir en costos recurrentes.

Como resultado, este proyecto ofrece una herramienta avanzada que supera las limitaciones de los sistemas existentes en el análisis de línea de visión y de la Zona de Fresnel. Estas características convierten a la herramienta en una solución para la planificación eficiente de redes LoRaWAN facilitando su adopción en sectores clave como la agricultura inteligente, la gestión de infraestructuras y la monitorización ambiental.

En futuras extensiones del proyecto, se podrían incorporar:

- Modelos de propagación de señal más complejos.
- Compatibilidad con otras tecnologías como 5G o Wi-Fi.
- Modo colaborativo para que varios usuarios trabajen simultáneamente.
- Análisis automático de mejores ubicaciones y alturas de gateways basados en áreas generadas.
- Adaptación dinámica del número de puntos intermedios al tamaño y resolución.
- Mecanismo de aviso al usuario del conteo de peticiones restante para las APIs con limitaciones.

10. REFERENCIAS

- [1] Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J., & Watteyne, T. (2017). *Understanding the limits of LoRaWAN*. IEEE Communications magazine, 55(9), 34-40.
- [2] Iova, O., Murphy, A.L., Picco, G.P., Ghiro, L., Molteni, D., Ossi, F., Cagnacci, F. (2017). *LoRa from the city to the mountains: exploration of hardware and environmental factors*. In: International Conference on Embedded Wireless Systems and Networks (EWSN) 2017, Uppsala, Sweden, 20-22 February 2017. Uppsala University: 317-322. ISBN: 9780994988614, from: <http://hdl.handle.net/10449/42132>
- [3] Alejandro Padrón G., José A. Rosendo R., Alberto Herrera B., Rafael Prieto M. V. Gerardo Calva O. (2007). *Cálculo de elevación para un enlace vía microondas mediante el análisis de zonas de Fresnel*. In: Congreso de Instrumentación (SOMI XXII) 2007, Monterrey, Nuevo León, México.
- [4] Rueda, E. A., Medina, F. F., & Barrera, J. F. (2009). *Criterio generalizado para la distinción entre difracción de Fraunhofer y Fresnel*. Grupo de Óptica y Fotónica, Instituto de Física, Universidad de Antioquia.
- [5] Doñate, P., Sánchez Bruno, M., Torres-Sanz, V., Sangüesa, J. A., Garrido, P., & Martínez, F. J. (2024). *Herramienta para el análisis de la orografía en redes LoRaWAN*. Computer Science and System Engineering Department, University of Zaragoza, España.
- [6] Vicente Torres-Sanz, Pablo Doñate, Julio A. Sangüesa, Francisco J. Martínez, Piedad Garrido (2024). *Enhancement LoRaWAN network deployment in rural environments using FresSim*. Computer Science and System Engineering Department, University of Zaragoza, España.
- [7] NASA JPL (2013). *NASA Shuttle Radar Topography Mission Global 1 arc second* [Data set]. NASA EOSDIS Land Processes Distributed Active

- Archive Center. Accessed 2024-11-01, from
<https://doi.org/10.5067/MEaSURES/SRTM/SRTMGL1.003>
- [8] European Space Agency. (2019). *Copernicus DEM – Global and European Digital Elevation Model*. From <https://dataspace.copernicus.eu/explore-data/data-collections/copernicus-contributing-missions/collections-description/COP-DEM>
- [9] Haxhibeqiri, J., De Poorter, E., Moerman, I., & Hoobeke, J. (2018). A survey of LoRaWAN for IoT: From technology to application. *Sensors*, 18(11), 3995. <https://doi.org/10.3390/s18113995>
- [10] Semtech Corporation. (n.d.). LoRa technology. Semtech. Retrieved November 19, 2024, from <https://www.semtech.com/lora>
- [11] LoRa Alliance. (n.d.). About LoRaWAN. LoRa Alliance. Retrieved November 19, 2024, from <https://lora-alliance.org/about-lorawan/>
- [12] Baek, J., & Choi, Y. (2018). A new GIS-based algorithm to support initial transmitter layout design in open-pit mines. *Journal of Energy Resources Technology*, 140(11), 111201. <https://doi.org/10.1115/1.4040564>
- [13] National Training Center. (n.d.). *The Fresnel Zone*. National Training Center. Retrieved November 19, 2024, from <https://nationaltrainingcenter.com/the-fresnel-zone/#:~:text=The%20Fresnel%20Zone%20is%20a,clear%20of%20obstructions%20and%20interference.>
- [14] ns-3 Consortium. (n.d.). About ns-3. ns-3. Retrieved November 19, 2024, from <https://www.nsnam.org/about/>
- [15] OMNeT++ Team. (n.d.). Introduction to OMNeT++. OMNeT++. Retrieved November 19, 2024, from <https://omnetpp.org/intro/>
- [16] Esri. (n.d.). ArcGIS Pro overview. Esri. Retrieved November 19, 2024, from <https://www.esri.com/en-us/arcgis/products/arcgis-pro/overview>

- [17] TTN Mapper Team. (n.d.). TTN Mapper. TTN Mapper. Retrieved November 19, 2024, from <https://ttnmapper.org/>
- [18] HeyWhatsThat. (n.d.). Hey, what's that?. HeyWhatsThat. Retrieved November 19, 2024, from <https://www.heywhatsthat.com/>
- [19] Solwise. (n.d.). Wireless elevation tool. Solwise. Retrieved November 19, 2024, from <https://www.solwise.co.uk/wireless-elevationtool.html>
- [20] SCADACore. (n.d.). RF line of sight tool. SCADACore. Retrieved November 19, 2024, from <https://www.scadacore.com/tools/rf-path/rf-line-of-sight/>
- [21] MathWorks. (n.d.). MATLAB. MathWorks. Retrieved November 19, 2024, from <https://es.mathworks.com/products/matlab.html>
- [22] VE2DBE. (n.d.). Radio Mobile Online. VE2DBE. Retrieved November 19, 2024, from <https://www.ve2dbe.com/english1.html>
- [23] Aalto University. (n.d.). FLORA: Flow-level simulation for RAN. Aalto University. Retrieved November 19, 2024, from <https://flora.aalto.fi/>
- [24] Lancaster University. (n.d.). LoRaSim: LoRa simulation tool. Lancaster University. Retrieved November 19, 2024, from <https://www.lancaster.ac.uk/scc/sites/lora/lorasim.html>
- [25] Docker. (n.d.). Docker. Docker. Retrieved November 19, 2024, from <https://www.docker.com/>
- [26] Meta. (n.d.). React documentation. Meta. Retrieved November 19, 2024, from <https://es.react.dev/>
- [27] Leaflet. (n.d.). Leaflet: An open-source JavaScript library. Leaflet. Retrieved November 19, 2024, from <https://leafletjs.com/>
- [28] React Leaflet Team. (n.d.). React Leaflet documentation. React Leaflet. Retrieved November 19, 2024, from <https://react-leaflet.js.org/>

- [29] Node.js Foundation. (n.d.). Node.js. Node.js Foundation. Retrieved November 19, 2024, from <https://nodejs.org/>
- [30] Vercel. (n.d.). Next.js: The React framework. Vercel. Retrieved November 19, 2024, from <https://nextjs.org/>
- [31] Microsoft. (n.d.). Bing Maps REST services: Elevations. Microsoft. Retrieved November 19, 2024, from <https://learn.microsoft.com/en-us/bingmaps/rest-services/elevations/>
- [32] Google Developers. (n.d.). Elevation API overview. Google. Retrieved November 19, 2024, from <https://developers.google.com/maps/documentation/elevation/overview>
- [33] Open-Meteo. (n.d.). Elevation API. Open-Meteo. Retrieved November 19, 2024, from <https://open-meteo.com/en/docs/elevation-api>
- [34] OpenTopoData. (n.d.). OpenTopoData. OpenTopoData. Retrieved November 19, 2024, from <https://www.opentopodata.org/>
- [35] European Space Agency (ESA). (n.d.). Copernicus: Observing the Earth. ESA. Retrieved November 19, 2024, from https://www.esa.int/Applications/Observing_the_Earth/Copernicus
- [36] NASA Science. (n.d.). Shuttle Radar Topography Mission (SRTM). NASA. Retrieved November 19, 2024, from <https://science.nasa.gov/mission/srtm/>
- [37] U.S. Geological Survey (USGS). (n.d.). SRTM 1 Arc-Second Global dataset. USGS. Retrieved November 19, 2024, from <https://lpdaac.usgs.gov/products/srtmgl1v003/>
- [38] European Commission. (n.d.). EU-DEM: Digital elevation model. Eurostat. Retrieved November 19, 2024, from <https://ec.europa.eu/eurostat/web/gisco/geodata/digital-elevation-model/eu-dem>

ANEXO I. CARACTERÍSTICAS DE LORAWAN

Las principales características de LoRaWAN son:

- Topología de red en estrella: utiliza una arquitectura de red en estrella, en la que los dispositivos finales (nodos) se comunican directamente con los gateways. Estos gateways retransmiten los datos al servidor de red central, donde se procesan y dirigen hacia las aplicaciones correspondientes. Los nodos no interactúan entre sí, lo que simplifica la arquitectura y reduce la complejidad de las comunicaciones.
- Bajo consumo de energía: está diseñado para dispositivos que funcionan con baterías, LoRaWAN permite que los nodos operen durante años con una sola batería. Esto es posible gracias a su capacidad para transmitir datos en ráfagas cortas y entrar en modo de bajo consumo cuando no están en uso.
- Clases de dispositivos: se definen tres clases de dispositivos para adaptarse a diferentes necesidades de comunicación:
 - Clase A: todos los dispositivos implementan esta clase básica. Permite comunicaciones bidireccionales, donde los nodos abren breves ventanas de recepción tras enviar un mensaje. Es la opción más eficiente energéticamente.
 - Clase B: los dispositivos programan ventanas de escucha adicionales sincronizadas mediante balizas transmitidas por los gateways. Esto mejora las oportunidades de recepción, aunque aumenta el consumo energético.
 - Clase C: los dispositivos están disponibles continuamente para recibir mensajes, salvo cuando están transmitiendo. Es ideal para aplicaciones donde la latencia debe ser mínima, aunque requiere mayor consumo de energía.
- Seguridad: utiliza cifrado AES-128 para garantizar la confidencialidad, integridad y autenticidad de los datos. Implementa claves únicas para cada

dispositivo, asegurando que solo los nodos autorizados puedan comunicarse con la red.

- Bandas de frecuencia libre: opera en bandas de frecuencia sin licencia, como 868 MHz en Europa y 915 MHz en América, lo que elimina los costes asociados al uso del espectro y permite implementaciones económicas.
- Privacidad: permite crear redes privadas adaptadas a necesidades específicas o utilizar redes públicas existentes, como The Things Network. Esto ofrece flexibilidad para implementaciones en diferentes escenarios.

ANEXO II. TECNOLOGIAS UTILIZADAS

En el desarrollo de este Trabajo Fin de Grado, se ha recurrido a una serie de tecnologías actuales tanto para la implementación de la mejora de la aplicación de despliegue de redes LoRaWAN, como para la visualización de mapas de cobertura y el procesamiento de datos. A continuación, se describen las principales herramientas y tecnologías empleadas:

- **Docker**[25]: es una plataforma que permite a los desarrolladores crear, desplegar y ejecutar aplicaciones en entornos aislados llamados contenedores. Esta tecnología ha sido utilizada para garantizar que la aplicación funcione de manera consistente en cualquier entorno, facilitando el despliegue y la integración de distintos componentes del sistema. Gracias a Docker, se han podido empaquetar las diferentes dependencias y servicios de la aplicación, asegurando un entorno reproducible y escalable, lo que resulta esencial para el despliegue de aplicaciones IoT que interactúan con diversas capas de software y hardware.
- **React**[26]: es una biblioteca de JavaScript de código abierto diseñada para la construcción de interfaces de usuario, especialmente en aplicaciones de una sola página (SPA). Esta tecnología se ha empleado para el desarrollo de la interfaz gráfica de la aplicación, proporcionando una experiencia de usuario fluida y reactiva. La capacidad de React para crear componentes reutilizables ha facilitado la construcción de la interfaz de usuario de la aplicación, permitiendo la integración eficiente de mapas interactivos y la visualización de los resultados del análisis de cobertura.
- **Leaflet**[27]: es una biblioteca de JavaScript de código abierto para la creación de mapas interactivos, que destaca por su simplicidad, rendimiento y capacidad para manejar grandes volúmenes de datos geoespaciales. En este proyecto, Leaflet ha sido empleado para generar los mapas interactivos que muestran las zonas de cobertura de la red LoRaWAN.
- **React-leaflet**[28]: es un conjunto de componentes que integra Leaflet con React, permitiendo la creación de mapas interactivos en aplicaciones construidas con esta última.
- **Node.js**[29]: es un entorno de ejecución de JavaScript del lado del servidor que permite construir aplicaciones rápidas y escalables.

- **Next.js[30]:** es un marco de desarrollo basado en React que permite crear aplicaciones web rápidas y optimizadas para SEO. Facilita tanto el renderizado del lado del servidor (SSR) como el renderizado estático de páginas, lo que permite a los desarrolladores crear aplicaciones que se cargan rápidamente y con una excelente experiencia de usuario. Entre sus principales características están el enrutamiento automático basado en archivos, la carga de código optimizada, el soporte para API Routes, y herramientas avanzadas de optimización.
- **Bing Maps Elevation API[31]:** proporciona datos de elevación para ubicaciones o rutas específicas a nivel global. Con esta API, se pueden obtener valores de elevación en metros para uno o varios puntos, o generar un perfil de elevación para una ruta específica. También permite consultar tipos de datos de elevación, como el promedio o la máxima elevación entre dos puntos en un camino.
- **Google Maps Elevation API[32]:** permite obtener datos de elevación en metros para ubicaciones específicas, ya sea en puntos únicos o a lo largo de una ruta definida. Los datos de elevación de Google incluyen información detallada sobre la altitud de puntos específicos, incluso a nivel del suelo, y se actualizan regularmente con información satelital y de otras fuentes de Google.
- **Open-Meteo Elevation API[33]:** es un servicio diseñado para obtener la altitud o elevación de un punto específico en el mapa a partir de coordenadas de latitud y longitud. Utiliza datos de modelos de terreno y es ideal para aplicaciones que requieren información de elevación en tiempo real o al acceder a datos meteorológicos precisos relacionados con la topografía del área consultada. Tiene una versión pública, de pago y self-hosted.
- **Open Topo Data[34]:** proporciona datos de altitud mediante el uso de modelos de elevación digital (DEM) de fuentes como SRTM, EU-DEM, entre otros. Esta API está orientada a quienes necesitan obtener información detallada de elevación en diversos entornos geográficos, y permite elegir entre varias fuentes de datos para ajustarse a las necesidades de precisión o cobertura del usuario. Tiene una versión pública y self-hosted.

ANEXO III. DESPLIEGUE DE OPEN TOPO

Open Topo Data ofrece diversas opciones para desplegar su API en un servidor, siendo la utilizada en este TFG el despliegue mediante Docker. A continuación, se describen los pasos necesarios para implementarlo:

1. Clonar repositorio oficial

Descargar el código de la API desde su repositorio en GitHub ejecutando el siguiente comando en una terminal:

```
$ git clone https://github.com/ajnisbet/opentopodata.git
```

2. Descargar dataset

Los datasets permitidos se encuentran listados en la documentación oficial de Open Topo Data. Para este manual, se ha utilizado el dataset SRTMGL1 v003, proporcionado por el Land Processes Distributed Archive Center de la NASA, con cobertura global y resolución de 30 metros. Los pasos son los siguientes:

1. Registrarse en NASA Earthdata⁶.
2. Acepta las siguientes licencias en la sección *EULAs* de "Mi perfil":
 - "Sentinel EULA"
 - OB.DAAC: ESA EULA for Sentinel-3 Data Use"
 - "MERIS EULA"

⁶ <https://urs.earthdata.nasa.gov>

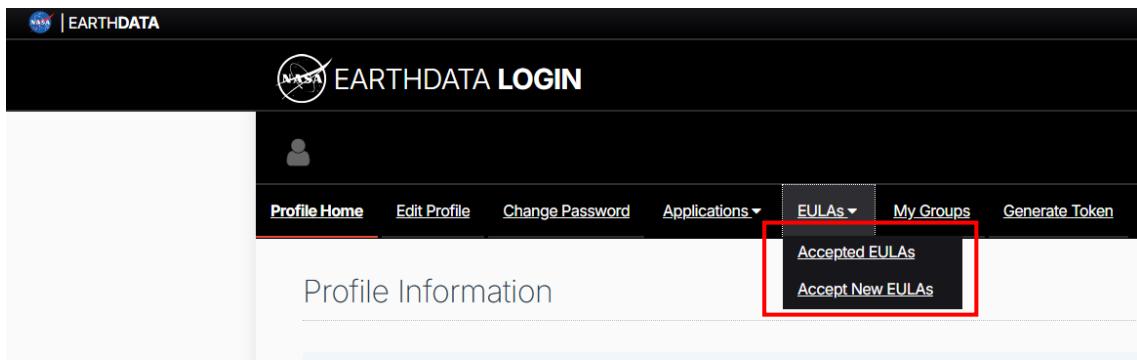


Ilustración 13 - Earthdata EULAs

3. Acceder a la herramienta EARTHDATA SEARCH⁷ e iniciar sesión con las credenciales recién creadas.
4. Configurar el filtro: *Instruments* → *View All* → *SRTM*.

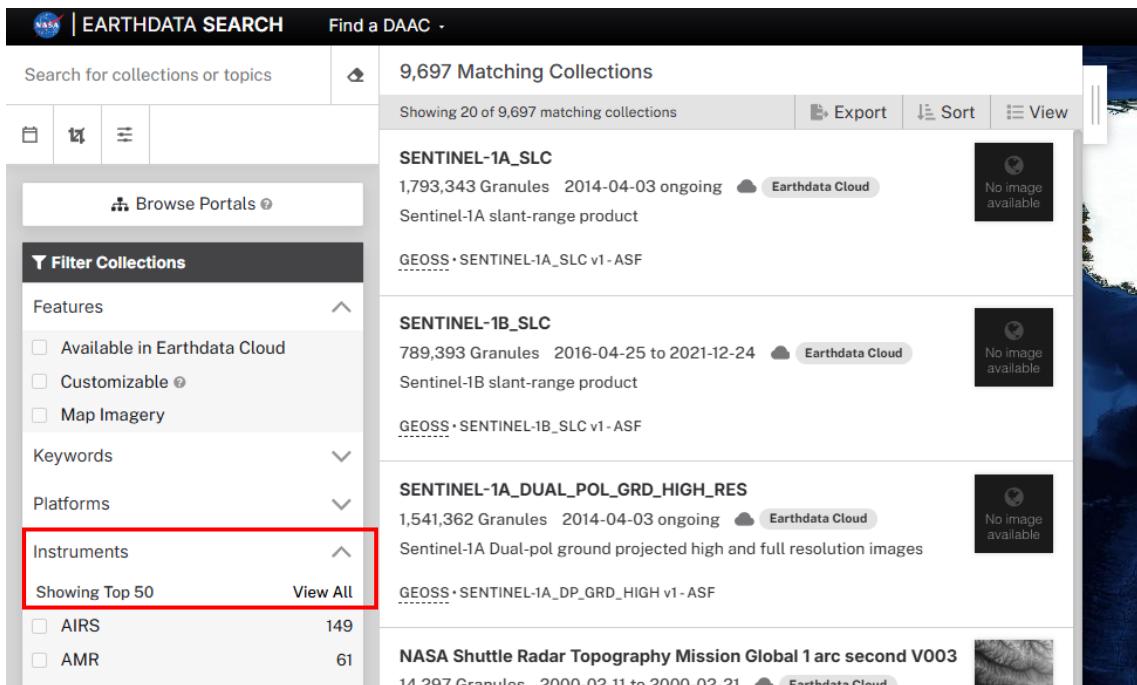


Ilustración 14 - Filtro de Earthdata

⁷ <https://search.earthdata.nasa.gov/>

5. Una vez filtrado, seleccionar *NASA Shuttle Radar Topography Mission Global 1 arc second V003* y clicar en el botón *Download All*, seguido de *Done* y *Download Data*. Se abrirá una página en la que aparecerán las opciones de descarga disponibles y comenzarán a prepararse los paquetes para la descarga. La descarga ocupa alrededor de los 120 gigabytes y consta de 14.297 ficheros.

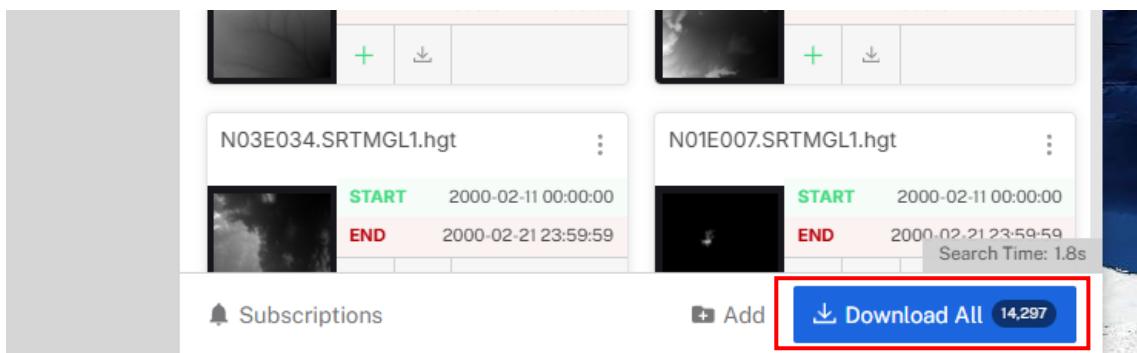


Ilustración 15 - Descarga de Earthdata (parte 1)

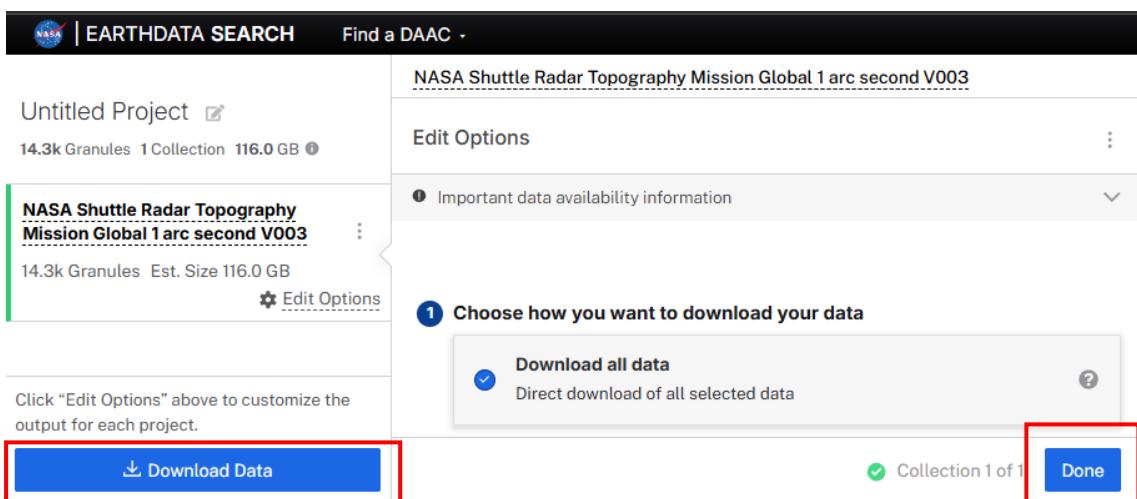


Ilustración 16 - Descarga de Earthdata (parte 2)

Cuando haya terminado la descarga, mover todos los archivos a la carpeta *data* ubicada en el directorio raíz de la API. Aunque es posible utilizar los archivos en formato ZIP, es conveniente descomprimir todos los archivos para aumentar el rendimiento en gran medida (ocupa alrededor de los 375 gigabytes).

3. Configuración de Open Topo Data

Una vez añadidos los archivos, es necesario que crear el archivo de configuración de Open Topo. Hay archivo de muestra llamado *example-config.yaml* en el directorio raíz de Open Topo, el cual se puede utilizar como base. Se debe eliminar el dataset de prueba y crear uno nuevo que apunte al directorio dentro de la carpeta *data* en el que se haya copiado el dataset. También es posible modificar el número máximo de puntos por petición permitido y el control de acceso. Debe quedar de la siguiente manera:

```

1  # 400 error will be thrown above this limit.
2  max_locations_per_request: 2000
3
4
5  # CORS header. Should be null for no CORS, '*' for all domains, or a url with
6  # protocol, domain, and port ('https://api.example.com/'). Default is null.
7  access_control_allow_origin: "*"
8
9
10 datasets:
11  # Config for 30 metre SRTM.
12  - name: srtm30m
13    path: data/srtm-30m-nasa-v3.0-hgt/
14  # filename_epsg: 4326 # This is the default value.
15  # filename_tile_size: 1 # This is the default value.
16

```

Ilustración 17 - Fichero de configuración de Open Topo

4. Compilar y ejecutar la API con Docker

Una vez preparado todo, abrir una terminal en el directorio raíz de Open Topo y compilar y ejecutar la API mediante los siguientes comandos:

- En entornos Windows:

```
$ docker build --tag opentopodata --file docker/Dockerfile .
```

```
$ docker run --rm -it -volume
"RUTA_ABSOLUTA_A_OPENTOPO\data:/app/data:ro" -p 5000:5000 -e
N_UWSGI_THREADS=8 opentopodata sh -c "/usr/bin/supervisord -c
/app/docker/supervisord.conf"
```

- En entornos Linux:

```
$ make build
```

```
$ make run
```

5. Acceso a la API

Una vez iniciada, la API estará disponible en el puerto 5000 del servidor. Si se despliega localmente, se podrá acceder a través de:

```
http://localhost:5000/v1/srtm30m?
```

Para enviar peticiones se debe utilizar el parámetro `locations=lat1,lon1/lat2,lon2/latN,lonN`. Por ejemplo, para las coordenadas latitud 40.35190 y longitud -1.09624 :

```
http://localhost:5000/v1/srtm30m?Locations=40.35190,-1.09624
```

Esto devuelve una respuesta con la siguiente estructura:

Dar formato al texto

```
{
  "results": [
    {
      "dataset": "srtm30m",
      "elevation": 1026,
      "location": {
        "lat": 40.3519,
        "lng": -1.09624
      }
    }
  ],
  "status": "OK"
}
```

Ilustración 18 - Respuesta de Open Topo

ANEXO IV. MANUAL DE USUARIO

1. Selección de la API de elevaciones

Por defecto, las elevaciones se obtienen a través de la API gratuita de Open-Meteo, la cual no requiere una clave de acceso (API key). Para cambiar la API utilizada, hay que hacer clic en el botón *API Selector* situado sobre el mapa. Esto abrirá una ventana donde se muestran las APIs disponibles y un campo para ingresar una clave, si es necesario. Para guardar los cambios se debe clicar en el botón *Save*. A la izquierda del botón se ubica un indicador de la API que se está utilizando.

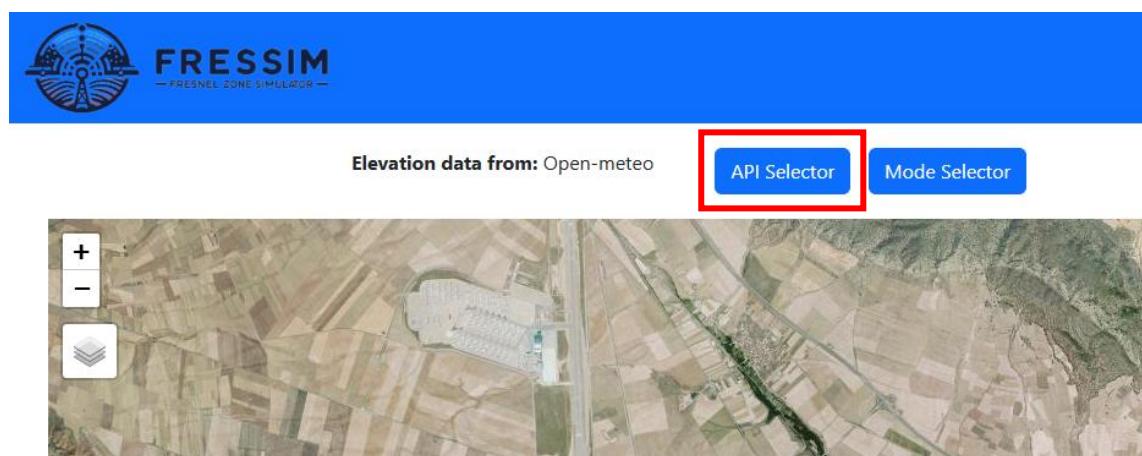


Ilustración 19 - Botón del selector de API

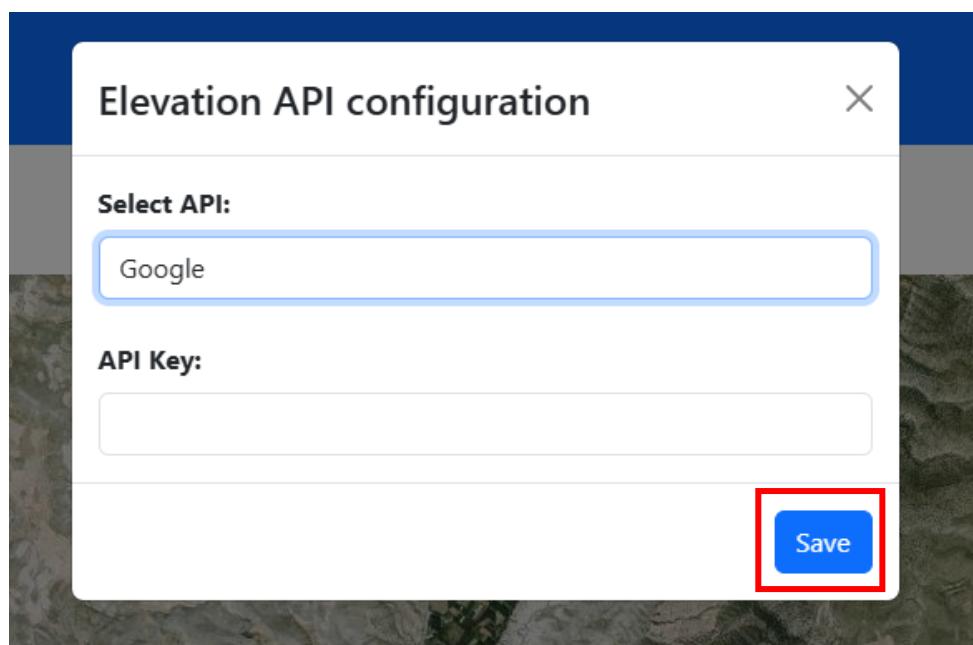


Ilustración 20 - Panel del selector de API

2. Selección de modo de análisis

La aplicación ofrece dos modos de uso: uno optimizado para mayor velocidad y otro para una precisión superior (por defecto modo velocidad). Para elegir entre ambos, hacer clic en el botón *Mode Selector* ubicado sobre el mapa, que abrirá una ventana con una descripción detallada de cada modo y un selector para realizar la elección. Para aplicar los cambios, haz clic en el botón *Guardar*.

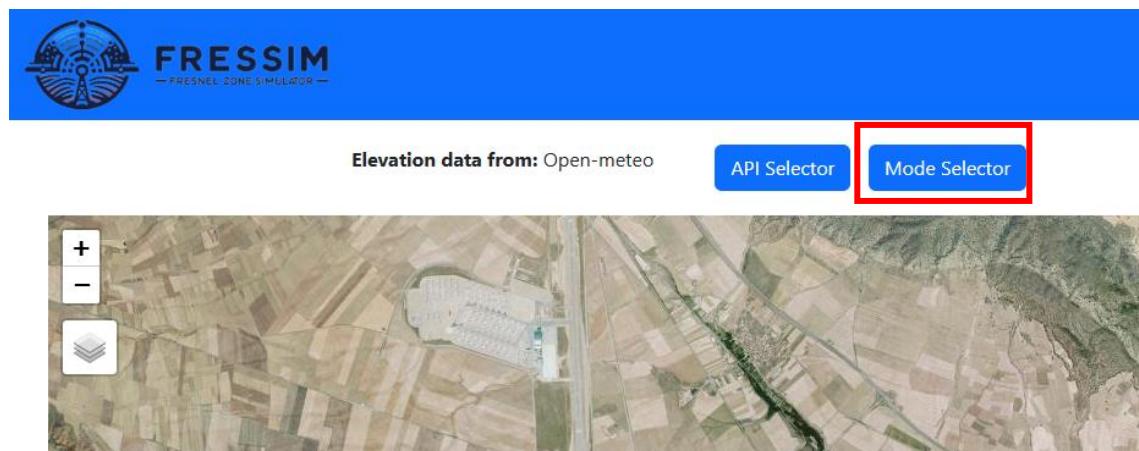


Ilustración 21 - Botón del selector de modo

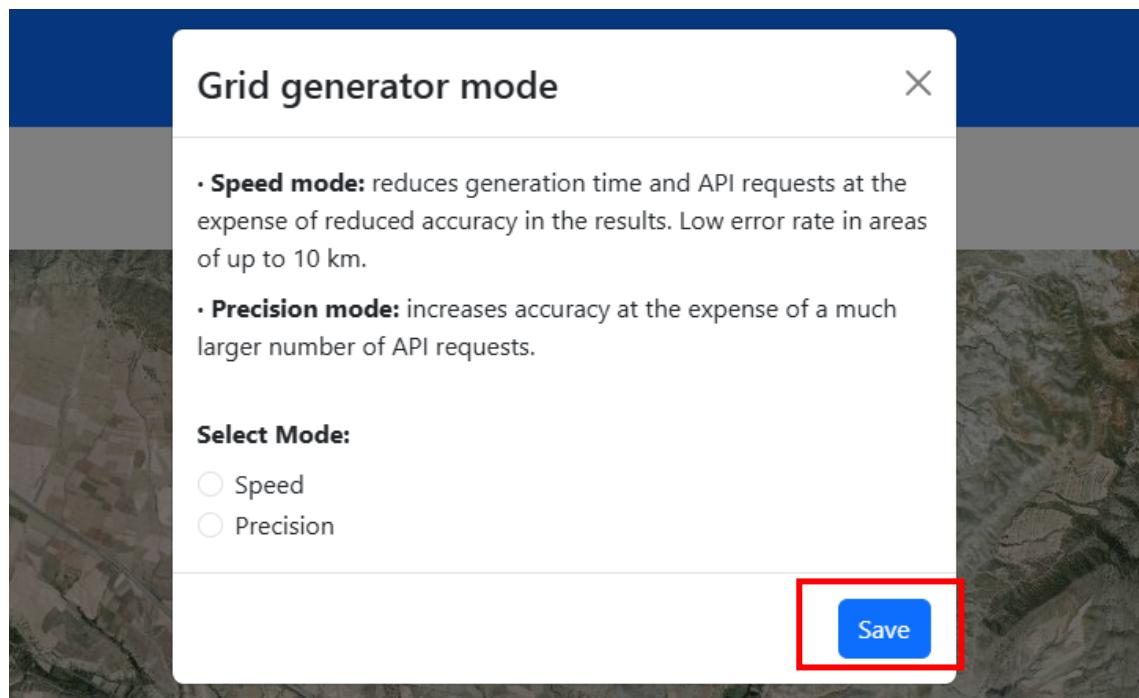


Ilustración 22 - Panel del selector de modo

3. Carga de datos de fuentes externas

La aplicación permite la carga de datos en formato JSON provenientes de las plataformas ChipStack y TTN. Para realizar la carga de datos, estos deben cargarse antes de iniciar la interacción con el mapa. El cuadro de carga se encuentra ubicado sobre el mapa. Para cargar un archivo, primero se selecciona la plataforma desde el menú desplegable y luego se hace clic en el botón *Select JSON Files*, lo cual abrirá una ventana para seleccionar el archivo deseado. Una vez cargado, los elementos se procesarán y se representarán automáticamente en el mapa.

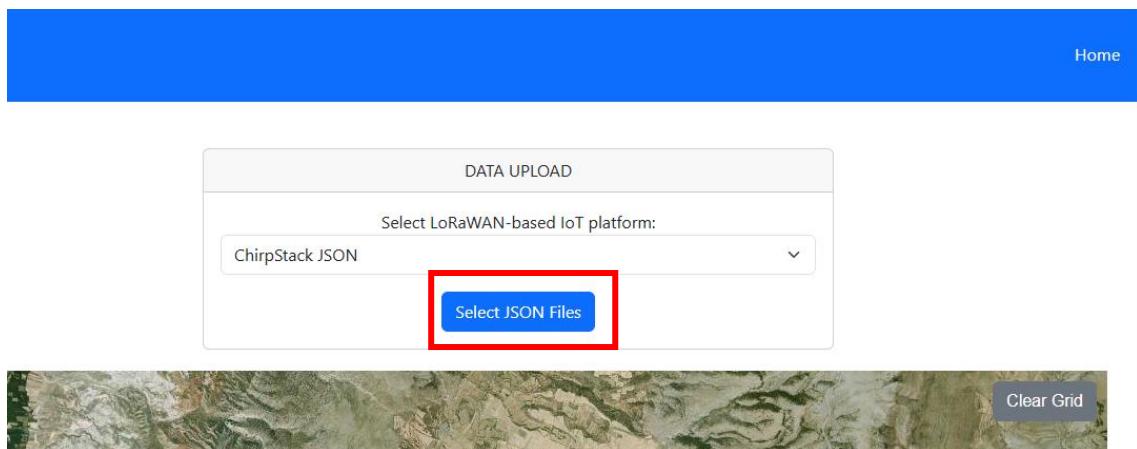


Ilustración 23 - Botón de carga de datos

4. Manejo de los nodos y gateways

4.1 Añadir elementos

Para añadir un gateway, se debe hacer clic derecho en el punto deseado del mapa. Del mismo modo, se pueden añadir nodos realizando un clic izquierdo. Al hacer clic con cualquiera de los dos botones, aparecerá una ventana solicitando la confirmación del elemento añadido. Los gateways se representarán con un marcador azul, mientras que los nodos aparecerán con marcadores de color rojo o verde, según su estado.

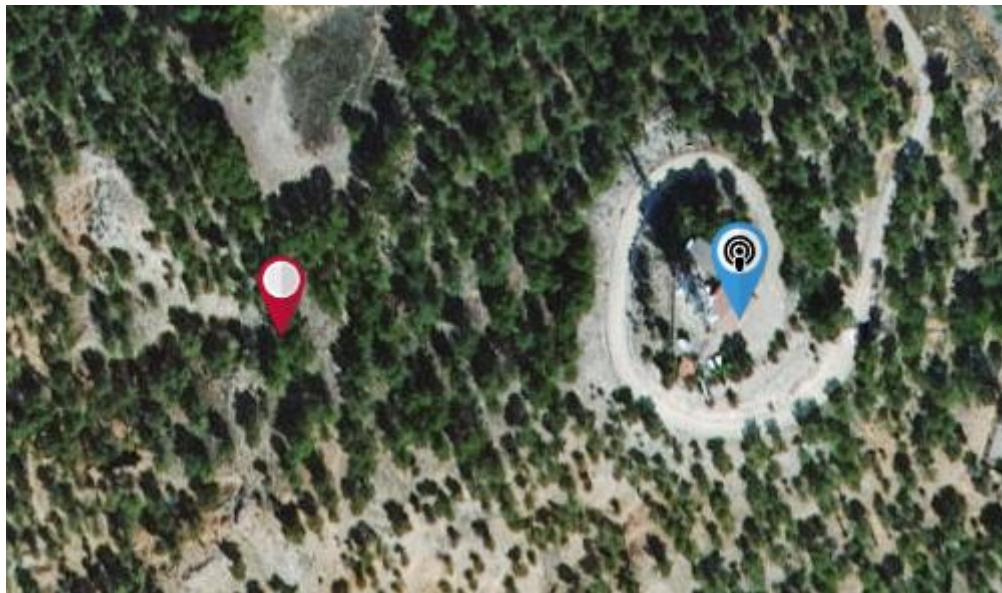


Ilustración 24 - Nodos y gateways en el mapa

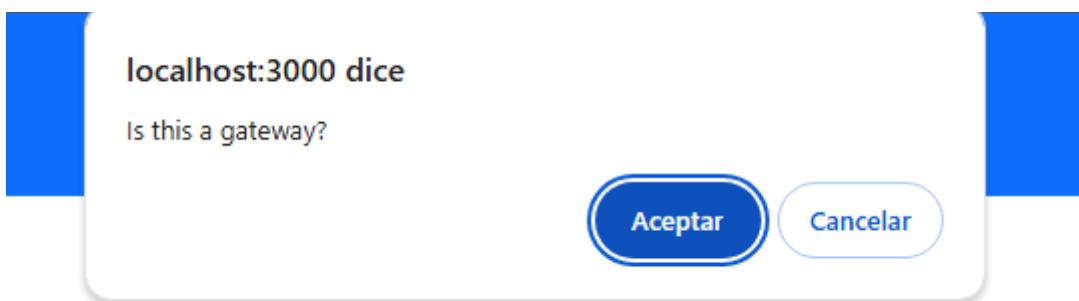


Ilustración 25 - Confirmación añadir gateway

4.2 Eliminar elementos

Para eliminar cualquiera de los dos elementos, se debe hacer clic sobre su marcador, lo que desplegará un menú emergente con varias opciones. Para proceder con la eliminación, se debe seleccionar el botón rojo con el icono de una papelera.



Ilustración 26 - Botón para eliminar nodos y gateway

4.3 Modificar la posición de un gateway

Para modificar la posición de un gateway, se debe hacer clic en su marcador y luego en el botón azul con el icono de una flecha. Esto abrirá una ventana donde se pueden ingresar las nuevas coordenadas. Para aplicar los cambios, se debe hacer clic en el botón *Save*.



Ilustración 27 - Botón para modificar posición de nodo y gateway

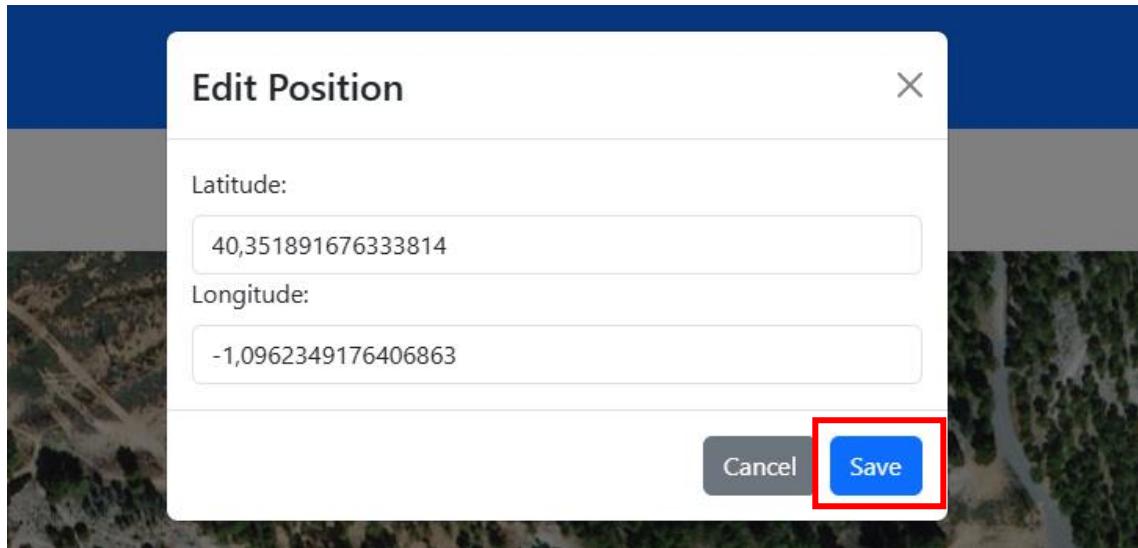


Ilustración 28 - Panel de modificación de posición

4.4 Modificar la altura de un elemento

Para modificar la altura de cualquiera de los dos elementos, se debe hacer clic en su marcador para desplegar un menú emergente con su información. A continuación, se debe seleccionar el ícono del lápiz junto a la opción *Ground level*. Esto abrirá una ventana donde se puede ingresar la nueva altura. Para guardar los cambios, es necesario hacer clic en el botón *Guardar*.



Ilustración 29 - Botón de modificación de la altura de un gateway

4.5 Cambiar nombre de un elemento

Para modificar el nombre de cualquiera de los dos elementos, se debe hacer clic en su marcador para desplegar un menú emergente con su información. A continuación, se debe seleccionar el ícono del lápiz junto a al nombre por defecto (*new-gw* o *new-point*). Esto abrirá una ventana donde se puede ingresar el nuevo nombre. Para guardar los cambios, es necesario hacer clic en el botón *Guardar*.



Ilustración 30 - Botón de edición del nombre

4.6 Modificar RSSI y SNR de un nodo

Para modificar el RSSI o SNR de un nodo, se debe hacer clic en su marcador para desplegar un menú emergente con su información. A continuación, se debe seleccionar el ícono del lápiz junto a *RSSI: / SNR:*. Esto abrirá una ventana donde se pueden ingresar los nuevos valores. Para guardar los cambios, es necesario hacer clic en el botón *Guardar*.

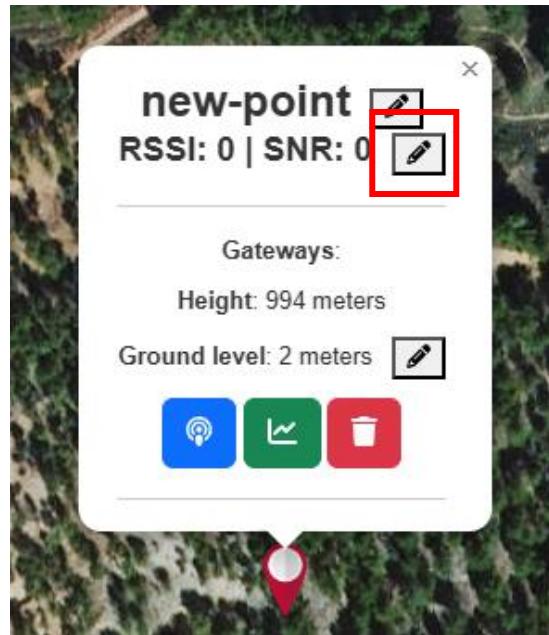


Ilustración 31 - Botón de modificación del RSSI y SNR

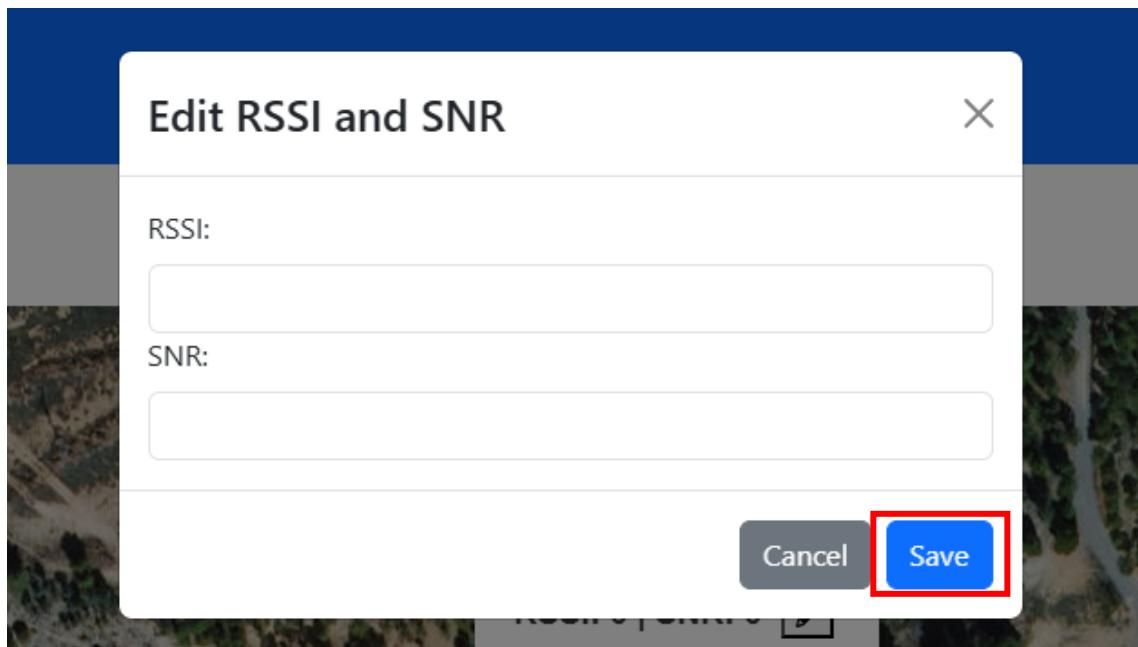


Ilustración 32 - Panel de modificación del RSSI y SNR

5. Generación de gráficas punto a punto

Las gráficas muestran las elevaciones entre el gateway y el nodo, así como la elipse de la primera zona de Fresnel junto con el porcentaje de ocupación. Para generar estas gráficas, es necesario que el nodo tenga asignado un gateway. La asignación de un gateway se realiza mediante el botón azul con el icono de una antena, ubicado en la ventana de información de los nodos. Al hacer clic en este botón, se despliega una ventana en la que se puede seleccionar uno o varios gateways para asignar al nodo.

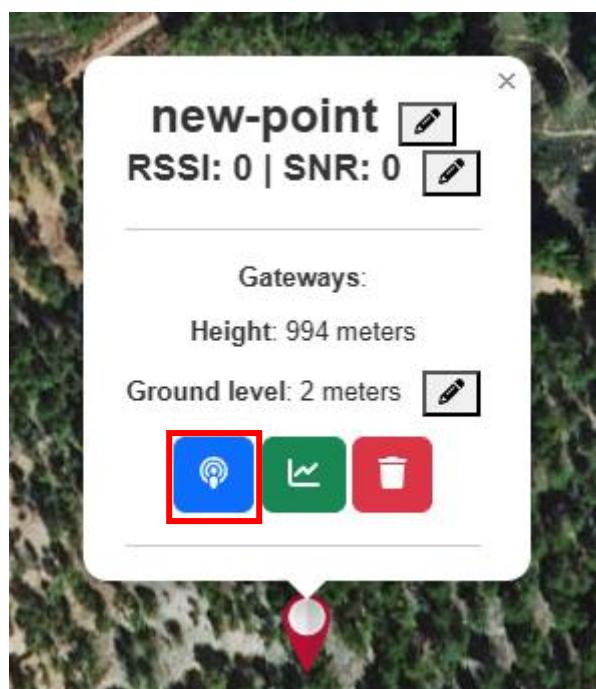


Ilustración 33 - Botón de generación de gráficas

Después de asignar los gateways, es necesario hacer clic en el botón verde con el icono de una gráfica. Esto abrirá una ventana donde se pueden ingresar valores adicionales y seleccionar los gateways asignados con los que se desea generar las gráficas. Una vez introducidos los datos, se debe hacer clic en el botón *Generate*, lo cual calculará automáticamente el nivel de ocupación y generará las gráficas correspondientes. Se abrirá una ventana independiente para cada gateway seleccionado.

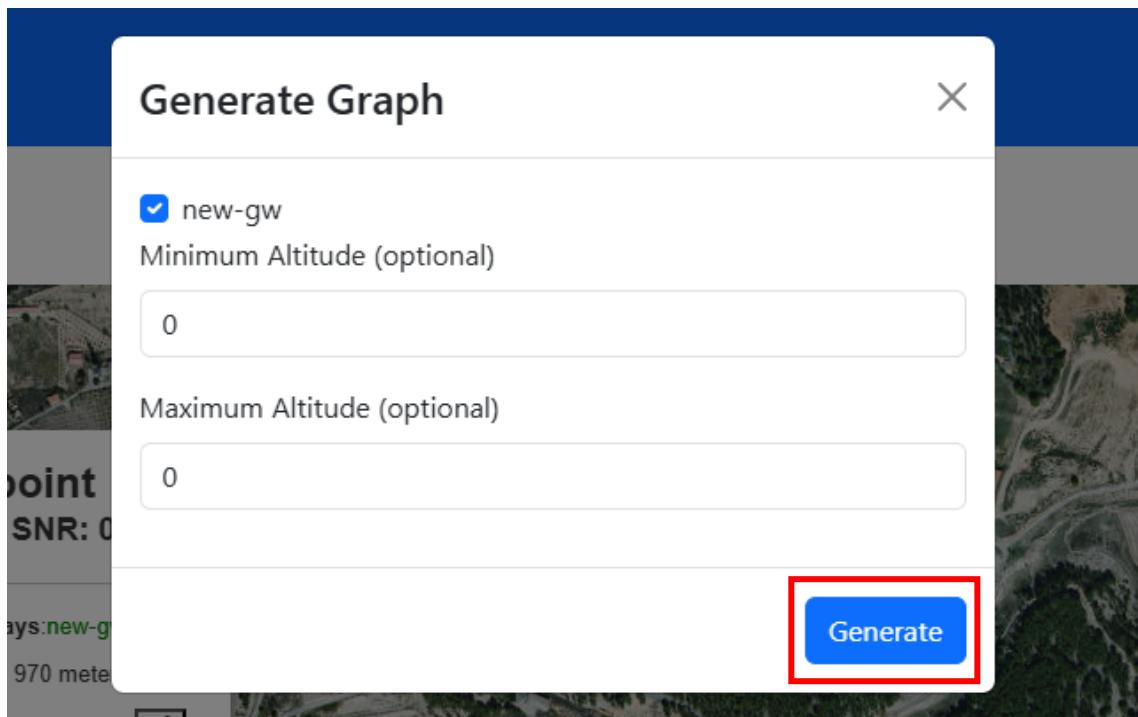


Ilustración 34 - Panel de generación de gráficas

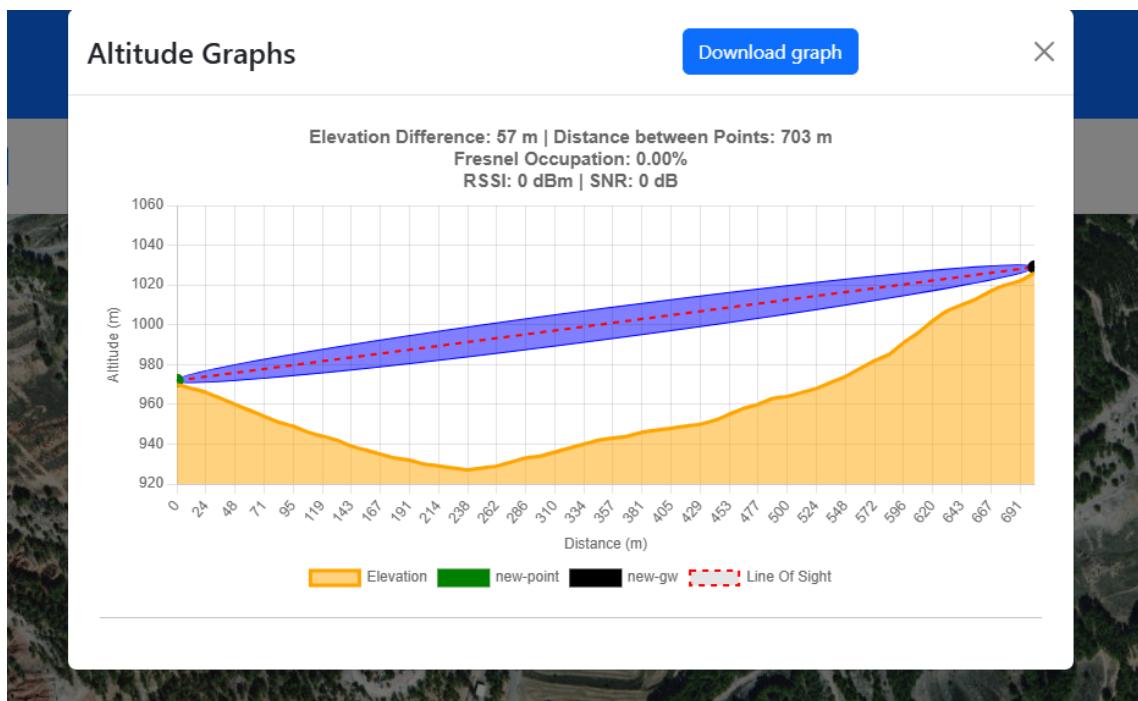


Ilustración 35 - Gráfica de la zona de Fresnel y elevaciones

6. Mapas y filtros

Para cambiar el mapa o seleccionar los filtros de sectores, coloca el cursor sobre el botón de capas, ubicado en la esquina superior izquierda del mapa. Al situar el cursor sobre este botón, se desplegará el menú correspondiente.

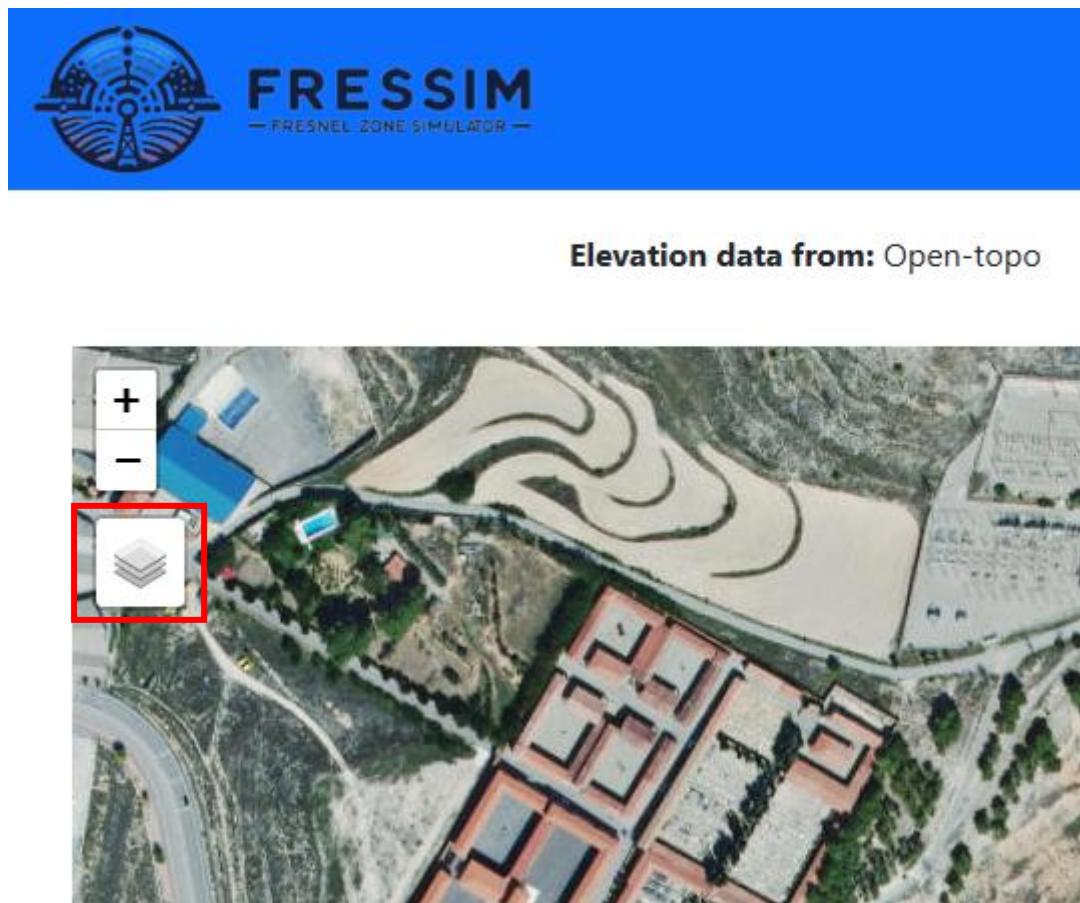


Ilustración 36 - Botón de capas del mapa



Elevation data from: Open-topo

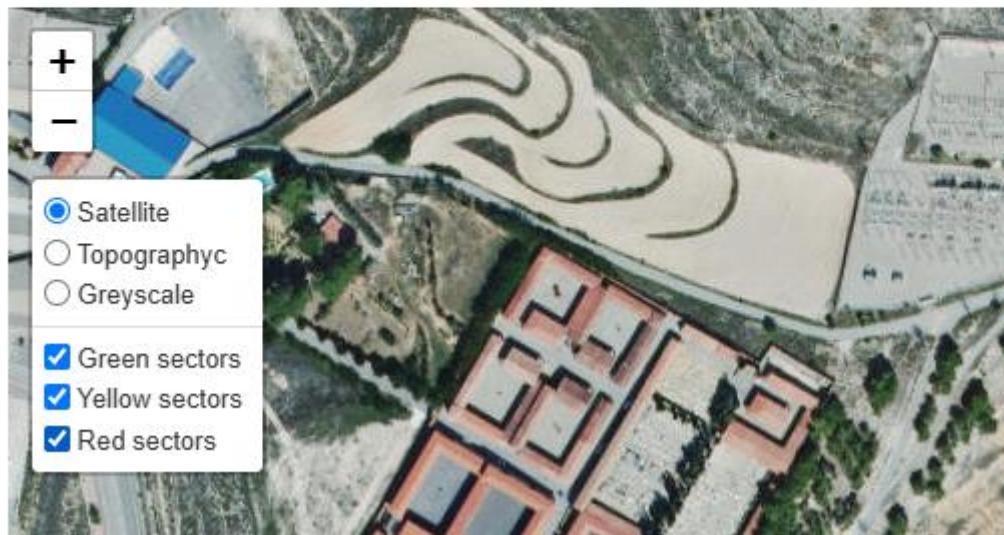


Ilustración 37 - Panel de capas del mapa

7. Exportar datos

Después de añadir elementos al mapa, es posible generar un archivo JSON con todos los datos. Para exportar estos datos, se debe hacer clic en el botón *Export JSON* situado debajo del mapa, lo que generará y descargará automáticamente un archivo en formato JSON.

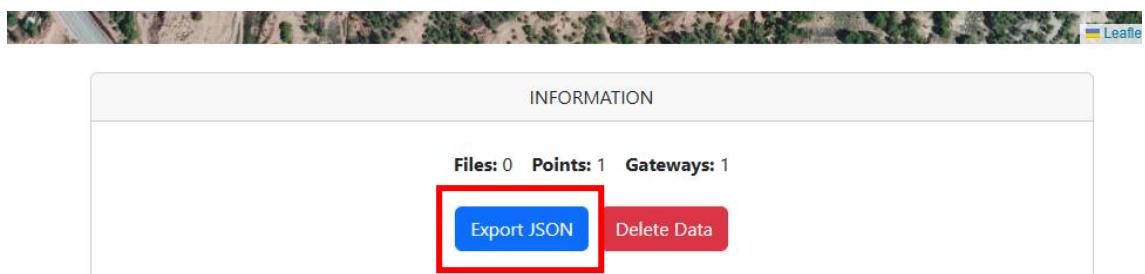


Ilustración 38 - Botón para exportar

8. Generar mapas de cobertura

Para generar el mapa de cobertura de un gateway, se debe hacer clic en su marcador para abrir un menú emergente con su información. Dentro del menú, se debe seleccionar el botón azul con el icono de barras de cobertura.



Ilustración 39 - Botón para generar mapas de cobertura

Se abrirá una ventana en la que se puede ingresar la altura sobre el suelo de los nodos, la escala del área de análisis y el número estimado de solicitudes a la API de elevaciones que se utilizarán. Después de seleccionar los parámetros deseados, se debe hacer clic en el botón *Generate* para iniciar la generación del mapa de cobertura.

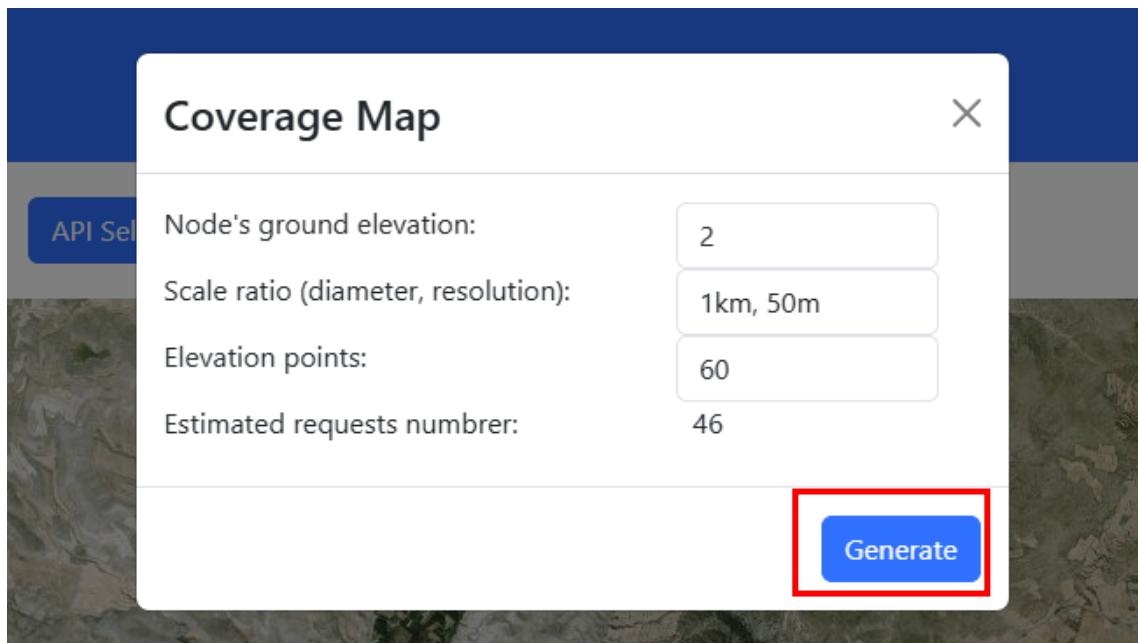


Ilustración 40 - Panel de generación de mapas de cobertura

Al pulsar el botón, se generará y mostrará el mapa de cobertura alrededor del gateway, utilizando tres colores para indicar distintos niveles de cobertura: verde para buena cobertura, amarillo para cobertura pobre y rojo para áreas sin cobertura.