



Universidad Politécnica
de Madrid



Escuela Técnica Superior de
Ingenieros Informáticos

Grado en Ingeniería Informática

Trabajo Fin de Grado

TOOLSET para la GESTIÓN de INFORMACIÓN SÍSMICA

Autor: Samuel Salgueiro Sánchez

Tutor(a): José Antonio Mateo Cortés

Madrid, junio de 2024

Este Trabajo Fin de Grado se ha depositado en la Escuela Técnica Superior de Ingenieros Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado

Grado en Ingeniería Informática

Título: TOOLSET para la GESTIÓN de INFORMACIÓN SÍSMICA (8105)

Abril 2024

Autor: Samuel Salgueiro Sánchez

Tutor:

José Antonio Mateo Cortés

Departamento de Arquitectura y Tecnología de Sistemas Informáticos

Escuela Técnica Superior de Ingenieros Informáticos

Universidad Politécnica de Madrid

RESUMEN

El proyecto se centra en el desarrollo de una aplicación web diseñada para simplificar la gestión de datos sísmicos, dirigida especialmente a sismólogos. La aplicación ofrece herramientas innovadoras, como un frontend web intuitivo y una base de datos eficiente, con el objetivo de mejorar la eficiencia en el análisis y la interpretación de la información sísmica.

La motivación principal detrás de este proyecto está en la necesidad de abordar la amenaza que representan los terremotos para la vida humana y la infraestructura. Al mejorar la gestión de datos sísmicos, se busca proporcionar a los sismólogos una herramienta más eficaz para comprender y prevenir los riesgos asociados con los movimientos telúricos.

El desarrollo de la aplicación se basa en una arquitectura cliente-servidor, con un frontend desarrollado en JavaScript y HTML, y un backend implementado en Node.js y Express.js. Se han utilizado diversas tecnologías y librerías, como Leaflet.js y Bootstrap, para garantizar un rendimiento óptimo y una experiencia de usuario atractiva.

En resumen, el proyecto se presenta como una solución práctica y efectiva para la gestión de datos sísmicos, con el potencial de mejorar significativamente la capacidad de los sismólogos para comprender y prevenir los riesgos asociados con los terremotos.

ABSTRACT

The project focuses on the development of a web application designed to simplify seismic data management, specifically aimed at seismologists. The application offers innovative tools, such as an intuitive web frontend, an efficient database, aimed at improving efficiency in the analysis and interpretation of seismic information.

The main motivation behind this project lies in the need to address the threat that earthquakes pose to human life and infrastructure. By improving seismic data management, it seeks to provide seismologists with a more effective tool to understand and prevent the risks associated with tellural movements.

The development of the application is based on a client-server architecture, with a frontend developed in JavaScript and HTML, and a backend implemented in Node.js and Express.js. Various technologies and libraries, such as Leaflet.js and Bootstrap, have been used to ensure optimal performance and an attractive user experience.

In short, the project presents itself as a practical and effective solution for seismic data management, with the potential to significantly improve the ability of seismologists to understand and prevent the risks associated with earthquakes.

Tabla de contenidos

CAPÍTULO 1. INTRODUCCIÓN.....	1
1.1 Motivación.....	2
CAPÍTULO 2. ESTADO DEL ARTE	3
CAPÍTULO 3. ANÁLISIS Y DISEÑO.....	7
3.1 Análisis.....	7
3.2 Diseño.....	10
CAPÍTULO 4. DESARROLLO	12
4.1 Arquitectura utilizada.....	12
4.1.1 Componentes de la arquitectura	12
4.1.2 Funcionamiento del modelo cliente-servidor	12
4.1.3 Ejemplo en el contexto del proyecto.....	13
4.1.4 Ventajas de la arquitectura.....	13
4.2 Componentes del proyecto.....	13
4.2.1 Cliente.....	14
4.2.2 Servidor	14
CAPÍTULO 5. CASO DE ESTUDIO.....	16
5.1 Caso de estudio 1.....	16
5.2 Caso de estudio 2.....	16
5.3 Caso de estudio 3.....	17
5.4 Caso de estudio 4.....	17
5.5 Conclusiones.....	18
CAPÍTULO 6. CONCLUSIONES Y TRABAJOS FUTUROS	19
6.1 Logros y Resultados	19
6.2 Desafíos y Aprendizajes	19
6.3 Trabajo Futuro:.....	20
CAPÍTULO 7. ANÁLISIS DE IMPACTO.....	21
BIBLIOGRAFÍA.....	22

CAPÍTULO 1. INTRODUCCIÓN

La investigación y el análisis de la actividad sísmica son fundamentales para comprender y mitigar los riesgos asociados a los terremotos. Los sismólogos desempeñan un papel crucial en esta tarea, ya que dependen en gran medida de la disponibilidad de datos sísmicos precisos y actualizados para llevar a cabo sus investigaciones y tomar decisiones informadas. Sin embargo, el proceso de recopilación y gestión de estos datos puede ser complejo y consumir mucho tiempo, especialmente cuando se trabaja con múltiples fuentes de información que no están estandarizadas.

En este contexto, surge la necesidad de desarrollar herramientas efectivas que permitan a los sismólogos acceder, gestionar y analizar la información sísmica de manera más eficiente. Es en este punto donde este trabajo de fin de grado (TFG) toma relevancia, al proponer un conjunto de herramientas innovadoras diseñadas específicamente para abordar estos desafíos.

El objetivo principal de este trabajo es proporcionar a los sismólogos un conjunto completo de herramientas que simplifiquen y agilicen el proceso de gestión de la información sísmica. Para lograr esto, se explorará y evaluará una variedad de Application Programming Interfaces (APIs) disponibles que ofrecen datos sísmicos, con el fin de identificar las más adecuadas para las necesidades específicas de este colectivo.

Una vez seleccionadas las APIs, se procederá al desarrollo de un frontend web intuitivo y fácil de usar que permita a los usuarios visualizar y manipular los datos sísmicos de manera efectiva. Este frontend actuará como la interfaz principal a través de la cual los sismólogos podrán acceder a la información, realizar consultas personalizadas y generar visualizaciones relevantes para sus investigaciones.

Además del frontal web, también se abordará la implementación de una base de datos sólida y eficiente para almacenar los datos sísmicos recuperados de las diferentes fuentes. El diseño de esta base de datos se realizará teniendo en cuenta las necesidades específicas de los sismólogos en cuanto a la estructura y organización de los datos.

Finalmente, se llevaría a cabo el despliegue de la aplicación en un servidor web, garantizando así su accesibilidad y disponibilidad para todos los usuarios interesados. Lamentablemente, este paso final no se pudo realizar debido a que se superarían las horas estimadas para la elaboración del TFG por lo que actualmente el aplicativo web se encuentra en una fase local.

1.1 Motivación

La motivación para llevar a cabo este trabajo de investigación surge de la necesidad de mejorar la eficiencia y efectividad en la gestión y análisis de datos sísmicos, un aspecto crucial en la sismología. Los terremotos representan una amenaza significativa para la vida humana, las infraestructuras y la economía, especialmente en regiones propensas a estos eventos naturales. La capacidad de predecir, comprender y mitigar los efectos de los terremotos depende en gran medida de la calidad y disponibilidad de los datos sísmicos.

Actualmente, los sismólogos enfrentan numerosos desafíos en su labor diaria. La recopilación de datos sísmicos de diversas fuentes, muchas veces no estandarizadas, puede ser un proceso duro y propenso a errores. Además, la falta de herramientas integradas que permitan un acceso rápido y sencillo a estos datos dificulta la realización de análisis exhaustivos y la toma de decisiones informadas.

Este proyecto se ve impulsado por el deseo de cerrar esta brecha tecnológica y proporcionar herramientas que les permitan enfocar sus esfuerzos en el análisis y la interpretación de los datos, en lugar de en la gestión administrativa de los mismos. Al desarrollar un conjunto de herramientas que incluya una interfaz web intuitiva, una base de datos eficiente y el despliegue accesible de la aplicación, se busca transformar la manera en que los sismólogos interactúan con la información sísmica.

En resumen, la motivación detrás de este trabajo consiste en la convicción de que, a través del desarrollo de tecnologías adecuadas, es posible mejorar la capacidad de los sismólogos para gestionar y analizar datos sísmicos, lo que a su vez tendrá un impacto positivo en la seguridad y el bienestar de las comunidades afectadas por los terremotos.

CAPÍTULO 2. ESTADO DEL ARTE

En la actualidad, los sismólogos disponen de varias herramientas web para la visualización y el análisis de terremotos. Sin embargo, estas herramientas presentan limitaciones que pueden afectar su utilidad y la experiencia del usuario. A continuación, se describirán algunas herramientas ya existentes.

GEOMAP

GEOMAP [4] es una herramienta que permite la visualización de terremotos ocurridos en los últimos 30 días a nivel mundial, con una magnitud superior a 2.5. Esta herramienta extrae datos de tres APIs diferentes, siendo una de ellas la del Servicio Geológico de Estados Unidos (USGS) [1].

Ventajas:

- Proporciona datos de terremotos de todo el mundo, lo cual es fundamental para su estudio.
- Los datos se actualizan continuamente, mostrando eventos recientes.

Desventajas:

- Al consultar detalles adicionales sobre los terremotos, la información no siempre se muestra de manera correcta, lo que puede limitar su utilidad para análisis detallados como se puede observar en la **Ilustración 1**.
- Puede carecer de opciones personalizables que mejoren la experiencia del usuario y faciliten la navegación y análisis de los datos.

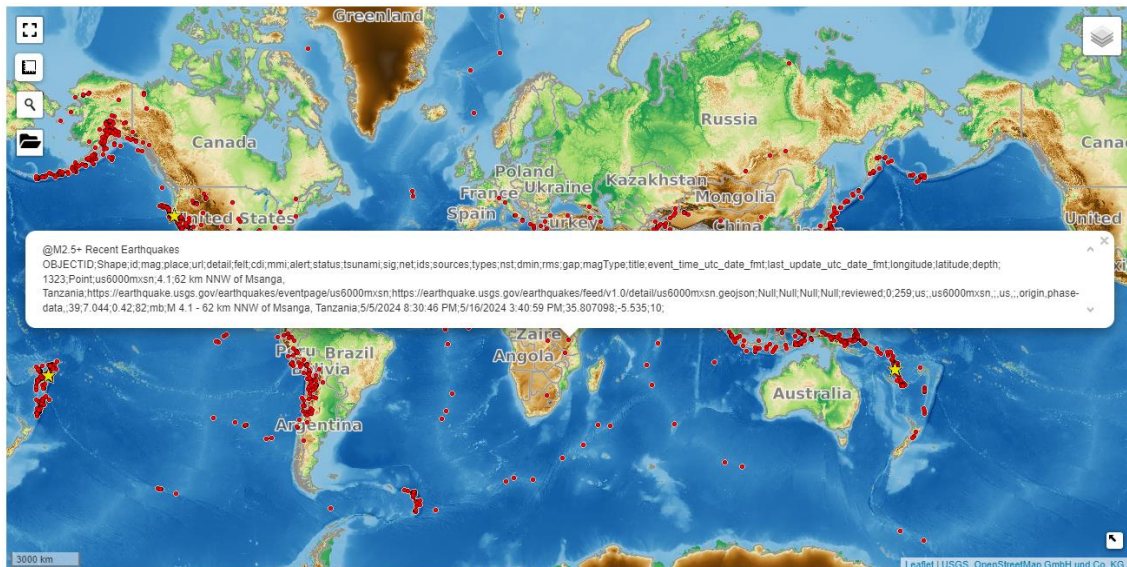


Ilustración 1. Mapa de GEOMAP con información de un terremoto

Visualizador del Instituto Geográfico Nacional (IGN)

El Instituto Geográfico Nacional de España ofrece varias herramientas para la visualización de terremotos en la región de España. Una de estas herramientas muestra los terremotos ocurridos en los últimos 30 días [3].

Ventajas:

- Esta herramienta es ideal para usuarios interesados específicamente en la actividad sísmica en España.
- Proporciona datos actualizados de los eventos sísmicos recientes en la región de España.

Desventajas:

- Solo muestra terremotos de los últimos 30 días, lo que no es suficiente para estudios a largo plazo.
- No permite la visualización de terremotos fuera de España, lo que limita su uso para estudios comparativos a nivel mundial.

Además, como se muestra más adelante en la **Ilustración 2** al querer ver la información de un terremoto en concreto, únicamente se muestra la fecha, la ubicación y la magnitud del mismo, en caso de que se quiera obtener más información hay que hacer CLICK en el enlace de “+info” el cual abrirá una nueva página donde detalla el resto de los datos del terremoto como se puede ver en la **Ilustración 3**.

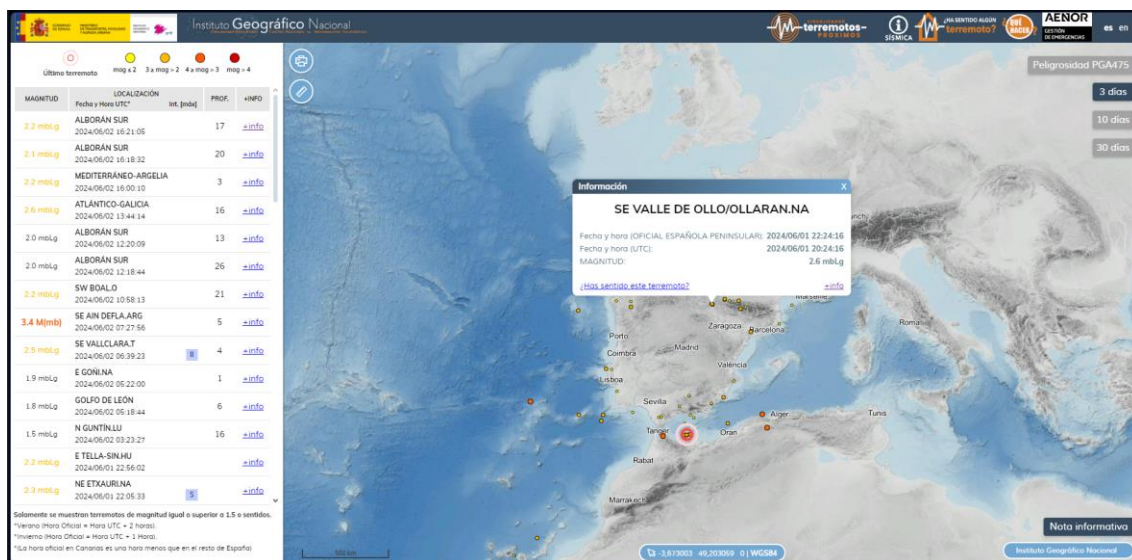


Ilustración 2. Web del Instituto Geográfico Nacional.

Información del terremoto

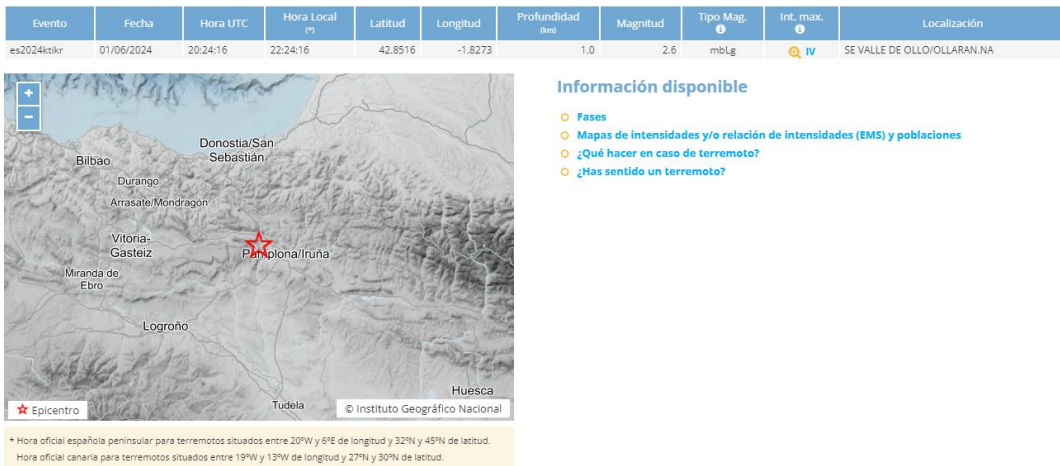


Ilustración 3. Información adicional del terremoto

Visualizador Histórico del Instituto Geográfico Nacional (IGN)

Otra herramienta del IGN permite seleccionar un rango de fechas para mostrar los terremotos en España, proporcionando una mayor flexibilidad temporal [2].

Ventajas:

- Los usuarios pueden seleccionar cualquier rango de fechas para obtener datos como se puede observar en la **Ilustración 4**.
- Permite obtener una visión más completa de la actividad sísmica en un período específico.

Desventajas:

- Los datos se presentan en un archivo separado por comas (CSV), lo que puede ser menos intuitivo y visualmente atractivo comparado con un mapa interactivo.
- La falta de visualización en un mapa puede dificultar la interpretación geoespacial de los datos para algunos usuarios, como se muestra en la **Ilustración 5** se visualiza un campo que corresponde a la localización y para acceder a la información adicional y por tanto la ubicación geográfica del terremoto, se tendrá que hacer **CLICK** en el campo “Mas Info” que abrirá una página donde detalla el resto de los datos del terremoto como se puede ver en la **Ilustración 3**.

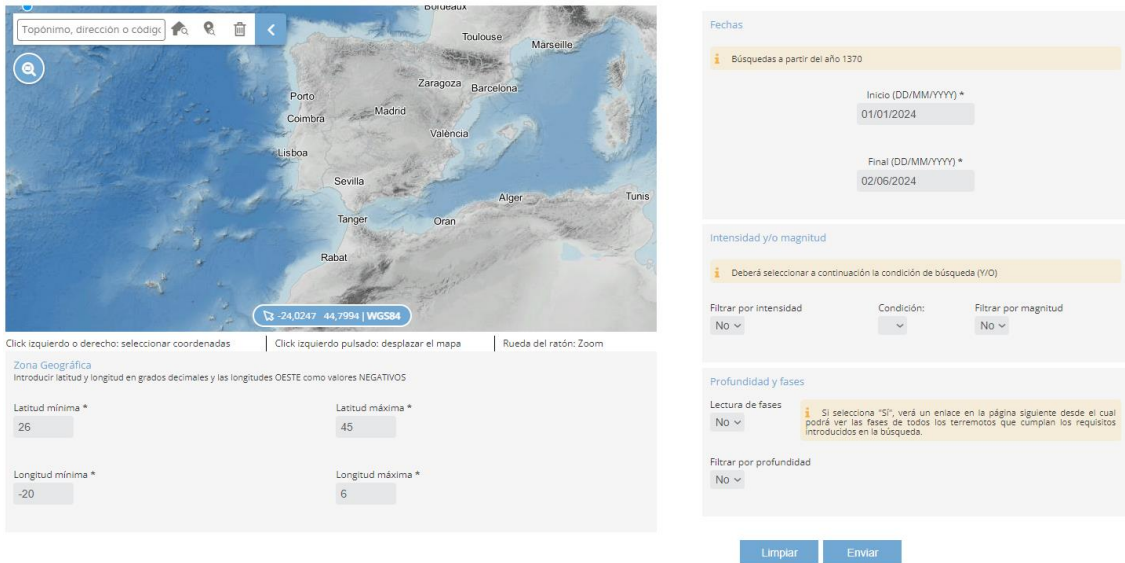


Ilustración 4. Mapa de la herramienta del IGN

Evento	Fecha	Hora UTC	Hora Local (h)	Latitud	Longitud	Profundidad (cm)	Magnitud	Tipo Mag.	Int. max.	Localización	Más Info
es2024aaatt	01/01/2024	00:23:02	01:23:02	42.7001	-7.1568	20.0	3.5	mbLg	III	SE TRIACASTELA.LU	Q
es2024aaecg	01/01/2024	02:03:51	03:03:51	42.2701	-7.6973	0.0	1.8	mbLg		NW BAÑOS DE MÓLGAS.OU	Q
es2024aagko	01/01/2024	03:14:07	04:14:07	38.8136	-7.9975	6.0	1.9	mbLg		N ARRAIOLOS.POR	Q
es2024aagwq	01/01/2024	03:28:13	04:28:13	39.7466	-0.3001	12.0	1.7	mbLg		W QUART DE LES VALLS.V	Q
es2024aagyq	01/01/2024	03:30:33	04:30:33	35.5233	-3.5161	18.0	1.8	mbLg		ALBORÁN SUR	Q
es2024aaigl	01/01/2024	04:09:59	05:09:59	36.4855	-4.5775	75.0	2.5	mb		SE FUENGIROLA.MA	Q
es2024aaimb	01/01/2024	04:16:30	05:16:30	36.3455	-5.8095	43.0	2.2	mbLg		NE BENALUP-CASAS VIEJAS.CA	Q
es2024aaajb	01/01/2024	04:59:37	04:59:37	27.7155	-18.0390	28.0	1.8	mbLg		W EL PINAR DE EL HIERRO.IHI	Q

Ilustración 5. CSV resultante de la búsqueda por rango de fechas

En resumen, con estos trabajos previos mencionados anteriormente, se pretenden tomar como ejemplo he implementar un aplicativo web en donde poder visualizar todos los terremotos a nivel mundial en un rango de fechas seleccionadas y magnitud mínima. Además, es de vital interés proporcionar al usuario una serie de elementos personalizables que mejoren su experiencia de usuario y que faciliten la tarea de obtención de datos precisos sin mucha complicación.

CAPÍTULO 3. ANÁLISIS Y DISEÑO

En este capítulo se presenta el análisis y diseño del TFG, detallando cómo se ha organizado y estructurado el sistema para lograr una administración eficiente y una arquitectura flexible. Se explicará la distribución de carpetas y directorios del proyecto, seguida de una descripción detallada del diagrama entidad-relación de la base de datos y el diagrama de secuencia que ilustra la interacción del usuario con la aplicación. Además, se proporcionará una visión completa del diseño de la interfaz de usuario.

3.1 Análisis

En esta sección se describirá la manera en que las carpetas y directorios del proyecto están organizados con el propósito de facilitar su administración y mantenimiento. A continuación, se presentará un diagrama de clases de la base de datos. Por último, se presentará un diagrama de secuencia que muestra la interacción y el flujo de los eventos entre los diferentes componentes de la aplicación.

A continuación, se procede a explicar como se ha estructurado el proyecto, tal y como se muestra la **Ilustración 6**:

- Carpeta *Servidor*: se encuentra todo lo relativo a la configuración del servidor. Allí se encuentran dos archivos, el primero de ellos “config.js” se encuentra la información del usuario y contraseña para la conexión con la base de datos, el segundo de ellos “server.js” además de funcionar como intermediario entre el frontal web y la base de datos, este recopila la información de la base de datos según los filtros que le indique el frontal.
- Carpeta *Script_Insertar_BBDD*: como su propio nombre indica, allí se encuentra un único archivo “InsertarDatosEnBBDD.java” que recopila información de la API del Servicio Geológico de Estados Unidos [1] en formato JSON y lo transforma para poder ser insertado en las distintas tablas de la base de datos.
- Carpeta *src*: se encuentra todo lo relativo al frontal web. Allí se encuentran principalmente dos archivos “index.html” y “script.js”, el primero de ellos es el que le da estilo a el frontal del aplicativo web, el segundo de ellos además de encargarse de “pintar” el mapa y las placas tectónicas, es el encargado de recoger los datos de los distintos filtros para enviarle una solicitud al servidor, para luego con la información recibida mostrar los datos en el mapa.

A continuación, se procede a explicar el diagrama entidad-relación, que es una representación gráfica que ilustra las entidades dentro de la base de datos y las relaciones entre ellas, el cual se representa en la **Ilustración 7**. El diagrama consta de dos entidades principales. La primera entidad es “TERREMOTOS”, en donde se representa la información del terremoto, a través de los campos “mag” que representa la magnitud del terremoto, “place” que representa la ubicación del terremoto, “country” que representa en que país ocurrió el terremoto, “time” que representa el tiempo en milisegundos en el que ocurrió el terremoto (el tiempo se empezó a medir desde el 1 de enero de 1970), “updated” que es el tiempo en milisegundos cuando se actualizó la información, “url” que es la url de la página web de USGS que contiene información adicional, “detail” que es la url de la API de USGS que proporciona más detalles en formato GeoJSON, “felt” que es el número de personas que reportaron haber sentido el terremoto, “cdi” que es la intensidad del terremoto en la escala de intensidad de Mercalli, “mmi” que es la intensidad del terremoto en la escala de intensidad de Mercalli modificada, “alert” que es el nivel de alerta del terremoto, “status”

que es el estado del terremoto, “tsunami” que indica si el terremoto generó un tsunami, “sig” que es la significancia del terremoto, “net” que es la red que detectó el terremoto, “code” que es el código único del terremoto, “ids” que son los identificadores asociados al terremoto, “sources” que son las fuentes que reportaron el terremoto, “types” que son los tipos de datos asociados al terremoto, “nst” que es el número de estaciones que contribuyeron a la localización del terremoto, “dmin” que es la distancia mínima entre la estación más cercana y el epicentro, “rms” que es la raíz media cuadrática de los residuos, “gap” que es el ángulo máximo entre las líneas que conectan el epicentro con las estaciones más cercanas, “magType” que es el tipo de magnitud, “title” que es el título del terremoto. La segunda es “GEOMETRÍA” en donde se almacena las coordenadas geográficas, así como la profundidad del terremoto. Estas dos entidades se relacionan a través del id de la entidad geometría con el geometry_id de la tabla terremotos.

La interacción del usuario con la aplicación se encuentra representada en el diagrama de secuencia de la **Ilustración 8**. La interacción se divide en dos pasos principales. El primer paso, el usuario introducirá un rango de fechas para visualizar los eventos sísmicos que ocurrieron en dicho rango y una magnitud mínima si lo desea, en caso de no introducir ninguna magnitud mínima a la hora de consultar con el servidor se tomara que la magnitud mínima es 0, y presionara el botón de mostrar para lanzar la consulta al servidor. En el segundo paso se le mostrara al usuario una serie de puntos en el mapa, este únicamente tendrá que situarse sobre cualquier punto que guste y presionarlo, una vez presionado se le mostrara en una ventana los datos relativos a ese terremoto como se mostrará más adelante en el apartado de diseño.

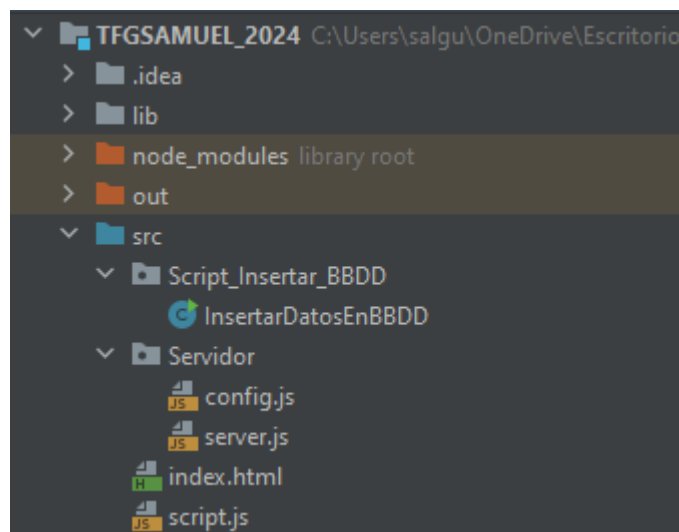


Ilustración 6. Estructura del proyecto.

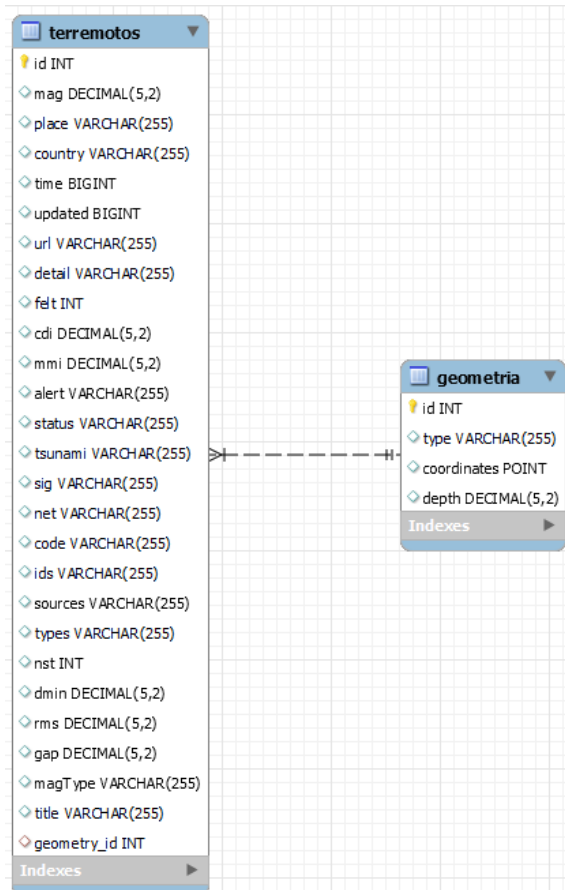


Ilustración 7. Modelo Entidad Relación de la base de datos.

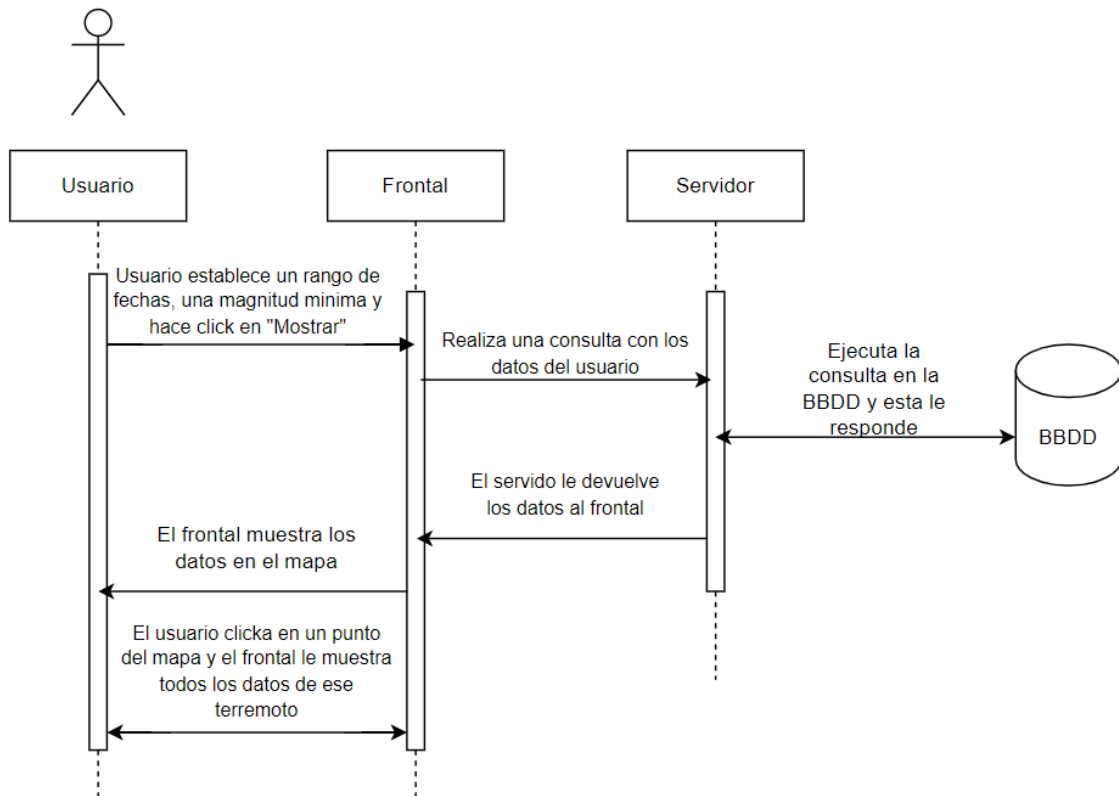


Ilustración 8. Diagrama de secuencia sobre la interacción de un usuario con la web para la visualización de un terremoto.

3.2 Diseño

Esta sección se centra en describir y detallar la interfaz de la aplicación desarrollada. Se proporcionará una visión completa de cómo se ha diseñado la interfaz para una experiencia de usuario intuitiva y eficiente.

Actualmente la web se encuentra en un entorno local y no ha sido desplegada en un servidor público. Por lo tanto, para acceder a la web, el usuario deberá contar con un dispositivo que tenga el proyecto alojado localmente. Al entrar en la web, el usuario se encontrará con tres cajas, un botón que indica la palabra “MOSTRAR”, un mapa con las placas tectónicas y una leyenda de colores a la izquierda de este para indicarle las distintas escalas de magnitud con su respectivo código de color como se muestra en la **Ilustración 9**.

Una vez el usuario introduzca el campo de fecha y una longitud mínima si lo desea, se mostrarán en el mapa distintos puntos correspondientes a los terremotos solicitados, si el usuario se sitúa encima de uno de ellos y hace “CLICK” en uno de ellos se le mostrara toda la información relevante del terremoto, todo eso se muestra en la **Ilustración 10** e **Ilustración 11**.



Ilustración 9. Estado inicial de la web.

HERRAMIENTA DE VISUALIZACIÓN DE SISMOS

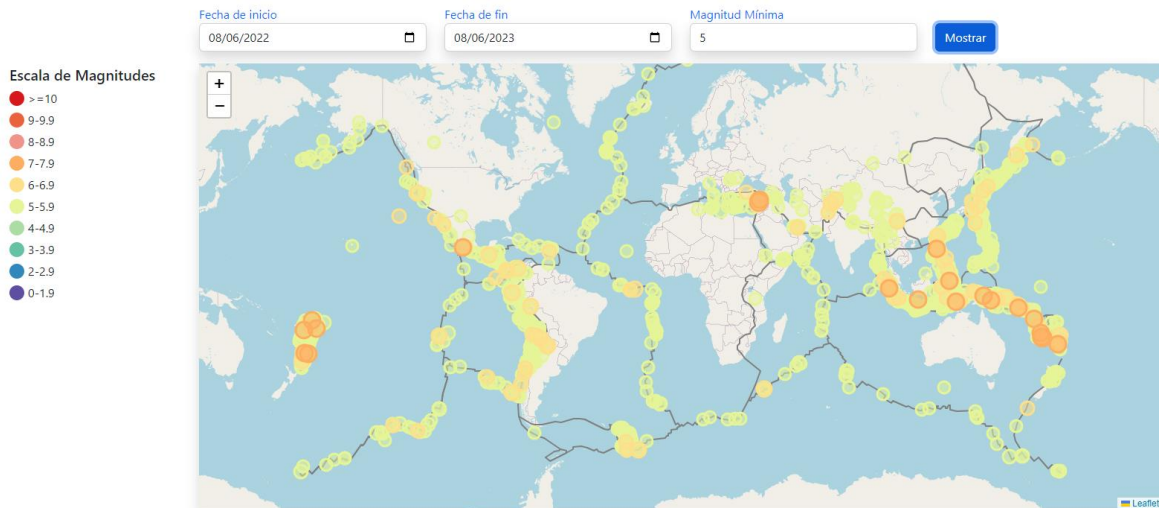


Ilustración 10. Mapa una vez presionado el botón mostrar.

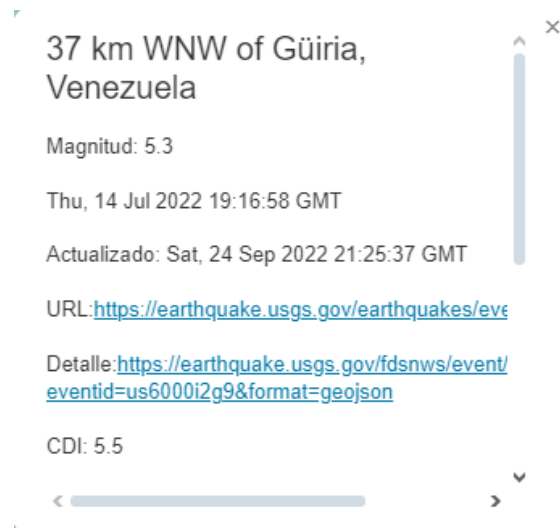


Ilustración 11. Parte de la información mostrada de un terremoto.

CAPÍTULO 4. DESARROLLO

En este capítulo se detalla el proceso de desarrollo del TFG, explicando cómo se ha implementado la arquitectura cliente-servidor para proporcionar una solución eficiente y escalable. Se describen los componentes principales de esta arquitectura, su funcionamiento y las tecnologías empleadas en cada uno de ellos. Además, se explican las ventajas de utilizar esta arquitectura en el contexto del proyecto. Posteriormente, se explica la implementación específica del cliente y del servidor, incluyendo las herramientas y lenguajes de programación utilizados, así como un script adicional desarrollado para poblar la base de datos.

4.1 Arquitectura utilizada

El proyecto está basado en una arquitectura cliente-servidor [5], que es un modelo de diseño de sistemas software capaz de gestionar múltiples solicitudes simultáneas. A continuación, se explica en detalle cómo funciona esta arquitectura, sus componentes y sus ventajas.

4.1.1 Componentes de la arquitectura

- **Cliente:** es una aplicación o dispositivo que inicia una solicitud de un recurso o servicio. En el contexto de una aplicación web, el cliente suele ser un navegador web que interactúa con el servidor a través de HTTP o HTTPS. Los clientes envían solicitudes al servidor y esperan respuestas, que generalmente son datos o archivos.
- **Servidor:** es una aplicación o dispositivo que responde a las solicitudes de los clientes. El servidor procesa las solicitudes, realiza las acciones necesarias (como consultas a una base de datos o procesamiento de datos) y devuelve la respuesta apropiada al cliente. En el contexto de una aplicación web, el servidor puede estar compuesto por varios subsistemas, como servidores web, servidores de aplicaciones y bases de datos.

4.1.2 Funcionamiento del modelo cliente-servidor

1. Solicitud del Cliente:

El cliente (por ejemplo, un navegador web) envía una solicitud HTTP/HTTPS al servidor. Esta solicitud puede ser para obtener una página web, enviar datos a través de un formulario, o solicitar datos específicos de una base de datos.

2. Procesamiento en el Servidor:

El servidor recibe la solicitud y la procesa. Este procesamiento puede involucrar lógica de negocio, validaciones y acceso a otros recursos.

3. Respuesta del Servidor:

Una vez que el servidor ha procesado la solicitud, genera una respuesta que puede incluir una página web HTML, un archivo, o datos en formato JSON o XML. Esta respuesta se envía de vuelta al cliente a través del mismo canal HTTP.

4. Visualización del Cliente:

El cliente recibe la respuesta del servidor y la procesa para mostrarla al usuario. En el caso de un navegador web, esto puede significar renderizar una página HTML, actualizar el contenido dinámico de la página mediante JavaScript, o mostrar mensajes de error.

4.1.3 Ejemplo en el contexto del proyecto

Para el desarrollo del aplicativo web, la arquitectura cliente-servidor se implementó de la siguiente manera:

- **Cliente:** El frontend de la aplicación, desarrollado con JavaScript y HTML, permite a los usuarios seleccionar un rango de fechas y una magnitud mínima para los terremotos. Estas selecciones se envían como solicitudes al servidor mediante la función “fetch()” que es una función propia de JavaScript para realizar peticiones HTTP.
- **Servidor:** El backend, desarrollado con Node.js y Express.js, recibe estas solicitudes. El servidor procesa las solicitudes realizando consultas a la base de datos MySQL y proporciona una API REST para obtener los terremotos que cumplen con los criterios especificados. Luego, el servidor envía los datos de vuelta al cliente en formato JSON.
- **Comunicación y Visualización:** Los datos recibidos por el cliente se utilizan para actualizar el estado de la aplicación y renderizar los terremotos en un mapa interactivo utilizando la librería Leaflet.js. Los usuarios pueden interactuar con el mapa para ver información detallada de cada terremoto.

4.1.4 Ventajas de la arquitectura

- **Escalabilidad:** Los servidores pueden ser escalados horizontalmente (añadiendo más servidores) para manejar un mayor número de solicitudes simultáneas.
- **Mantenibilidad:** La separación de responsabilidades entre cliente y servidor permite una mayor facilidad de mantenimiento y actualización de los componentes individuales sin afectar al otro.
- **Seguridad:** Los datos sensibles y la lógica de negocio se mantienen en el servidor, proporcionando una capa adicional de seguridad contra accesos no autorizados.
- **Flexibilidad:** Los clientes pueden ser de diferentes tipos (navegadores web, aplicaciones móviles, etc.) y seguir interactuando con el mismo servidor.

4.2 Componentes del proyecto

Como se ha descrito en el apartado anterior, el proyecto se desarrolló utilizando la arquitectura cliente-servidor. A continuación, se explicará en detalle las tecnologías empleadas en cada componente y el porqué de su elección. Cabe destacar que además del cliente y del servidor, se desarrolló un script escrito en lenguaje JAVA el cual se utilizó

para poblar la base de datos MySQL, con los datos de la API del Servicio Geológico de Estados Unidos [1].

4.2.1 Cliente

El cliente consta de un aplicativo web que se desarrolló utilizando JavaScript y HTML. Se eligieron estos lenguajes de programación debido a la familiaridad del desarrollador con los mismos y el gran parecido de JavaScript con el lenguaje JAVA que es el lenguaje utilizado y estudiado en la mayoría de las asignaturas del grado.

Tecnologías y librerías utilizadas:

- **JavaScript** [6]: Es un lenguaje de programación interpretado que se utiliza principalmente para crear contenido web dinámico. Permite a los desarrolladores implementar funcionalidades complejas en las páginas web, como la actualización de contenido sin recargar la página, la manipulación del DOM y la gestión de eventos.
- **HTML (HyperText Markup Language)** [7]: Es el lenguaje estándar para crear páginas web y aplicaciones web. Define la estructura de la página web utilizando una serie de elementos y etiquetas que el navegador interpreta para mostrar el contenido de manera adecuada.

Además de estos lenguajes fundamentales, se utilizaron las siguientes librerías para mejorar la funcionalidad y el diseño del cliente:

- **Leaflet** [8]: Es una librería JavaScript de código abierto utilizada para crear mapas interactivos. Es ligera, fácil de usar y tiene muchas funcionalidades útiles como agregar capas de datos, manejar eventos de usuario y mostrar marcadores.
- **Bootstrap** [9]: Es un framework de código abierto para desarrollar con HTML, CSS y JS. Permite diseñar sitios web responsivos de manera rápida y sencilla. Proporciona componentes predefinidos como botones, cuadros de texto y formularios, que se utilizaron para dar estilo a la interfaz de usuario del aplicativo web. Esto facilita la creación de una UI consistente y atractiva sin tener que escribir código CSS desde cero.
- **Fetch API** [10]: Es una interfaz propia de JavaScript para realizar solicitudes HTTP al servidor. Proporciona una forma sencilla y poderosa de interactuar con recursos de red. En el proyecto, Fetch API se utilizó para enviar solicitudes HTTP al servidor y recuperar los datos de los terremotos en formato JSON. Esto permitió actualizar dinámicamente el contenido del mapa en función de los filtros seleccionados por el usuario.

4.2.2 Servidor

El servidor de la aplicación está implementado utilizando Node.js y el framework Express.js y la base de datos esta implementada en el lenguaje MySQL.

Tecnologías utilizadas:

- **Node.js** [11]: Es un entorno de ejecución de JavaScript en el lado del servidor. Permite ejecutar código JavaScript en el servidor, lo que permite a los desarrolladores utilizar un solo lenguaje para el desarrollo completo de aplicaciones (tanto en el cliente como en el servidor).

- **Express.js** [12]: Es un framework web minimalista y flexible para Node.js que proporciona un gran conjunto de características para desarrollar aplicaciones web y móviles. Es fácil de configurar y usar, y permite manejar rutas, solicitudes y respuestas HTTP de manera eficiente. En el proyecto, Express.js se utilizó para crear el servidor que responde a las solicitudes HTTP del cliente. Esto incluye procesar las solicitudes, realizar consultas a la base de datos MySQL y devolver los datos en formato JSON.
- **MySQL** [13]: Es un sistema de gestión de bases de datos relacional de código abierto. Es conocido por su fiabilidad, rendimiento y facilidad de uso. En el proyecto, MySQL se utilizó para crear la base de datos en MySQLWorkBench donde se almacenaron y gestionaron los datos de los terremotos.

CAPÍTULO 5. CASO DE ESTUDIO

Para verificar si la aplicación cumple con el objetivo de mostrar los terremotos en un determinado rango de fechas y magnitud mínima, se diseñó un caso de estudio para el rango de fechas 08/06/2022-08/06/2023. Las pruebas que se realizaron consistieron en ir probando ese rango de fechas con la combinación de distintas magnitudes mínimas.

5.1 Caso de estudio 1

Para este primer caso de estudio se utilizó el rango de fechas especificado y la magnitud mínima 0, como se muestra en la **Ilustración 12**. El resultado de esta prueba fue un tiempo de respuesta muy alto siendo 10.66s el tiempo de respuesta de la consulta con el servidor ya que se recopilaban 133.971 terremotos. Además, al tener tanta cantidad de datos para mostrar en el mapa, la aplicación tiene un tiempo de respuesta alto al desplazamiento por el mapa y si se presiona un terremoto en particular también se percibe que el tiempo de respuesta a la hora de desplazarnos hacia abajo en el pop-up de la información del terremoto es bastante alto.

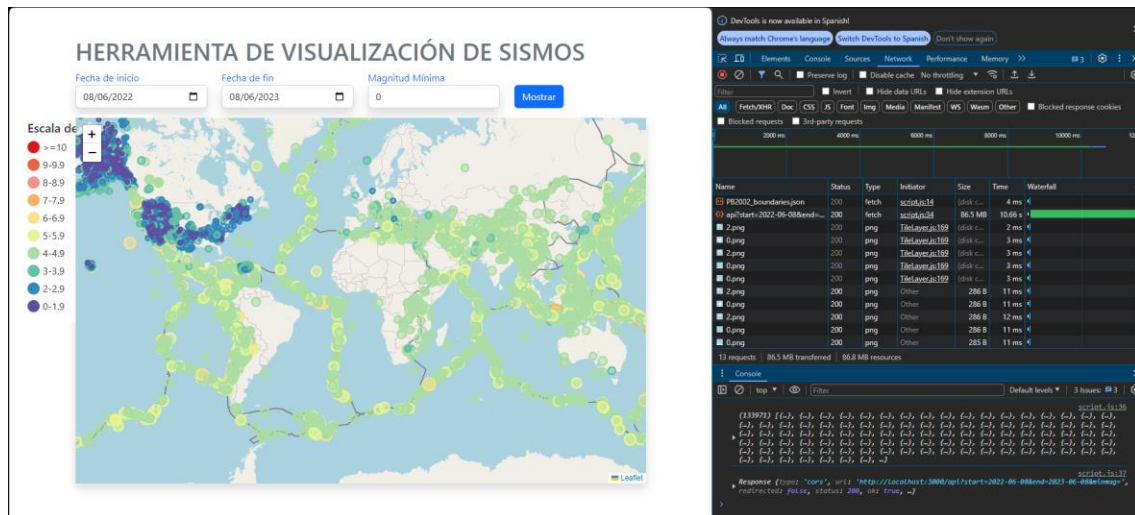


Ilustración 12. Caso de estudio 1

5.2 Caso de estudio 2

Para este segundo caso de estudio se utilizó el rango de fechas especificado y la magnitud mínima 2.1, como se muestra en la **Ilustración 13**. El resultado de la prueba también fue un tiempo de respuesta alto del servidor tardando este 9.01s y recopilando 35.384 terremotos. Sin embargo, el tiempo de respuesta al desplazamiento en el mapa no es tan alto como en el caso anterior, pero aun así sigue estando por encima de los valores óptimos, ya que existe un pequeño delay, si se presiona un terremoto en particular se percibe también un pequeño delay cuando se abre el pop-up, pero no se percibe ningún delay a la hora de desplazarse hacia abajo.

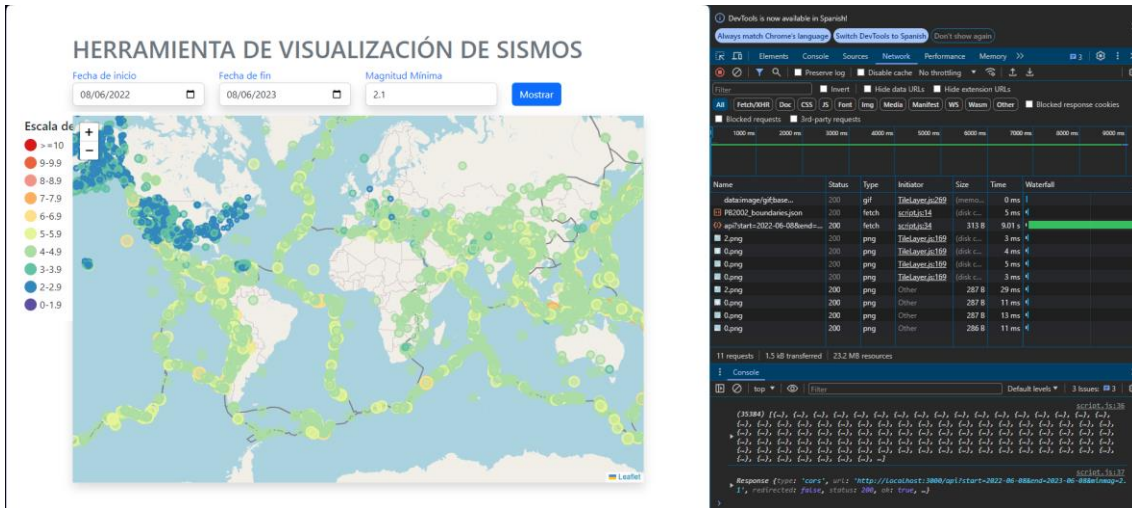


Ilustración 13. Caso de estudio 2

5.3 Caso de estudio 3

Para este tercer caso de estudio se utilizó el rango de fechas especificado y la magnitud mínima 3, como se muestra en la **Ilustración 14**. El resultado de la prueba también fue un tiempo de respuesta alto por parte del servidor (8.77s) recopilando éste 18.727 terremotos. Sin embargo, no existe ningún delay en el desplazamiento por el mapa y en el pop-up.

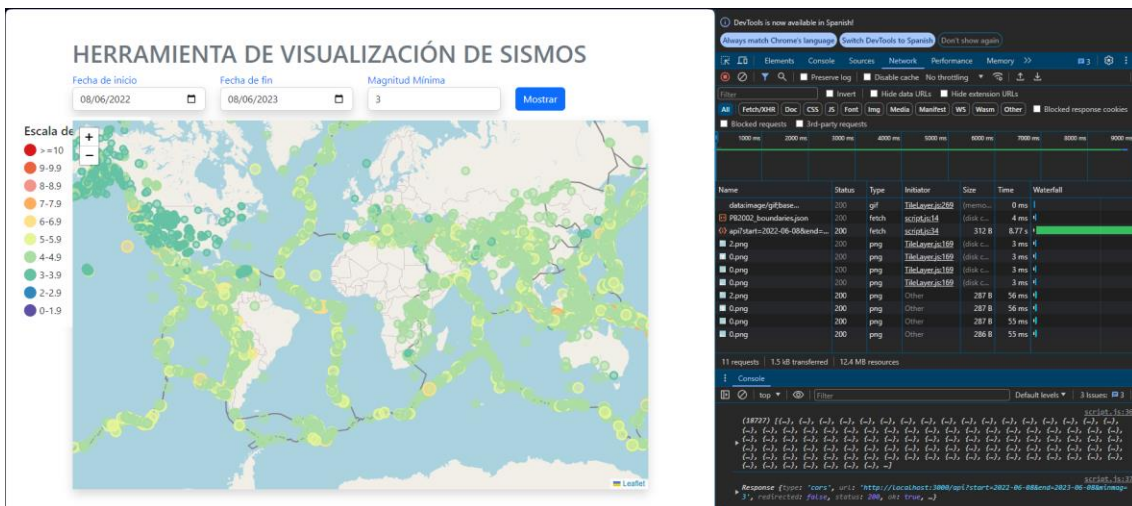


Ilustración 14. Caso de estudio 3

5.4 Caso de estudio 4

Para este cuarto y ultimo caso de estudio se utilizó el rango de fechas especificado y la magnitud mínima 5, como se muestra en la **Ilustración 15**. El resultado de la prueba fue un tiempo de respuesta bajo (85 ms) recopilando éste 1.524 terremotos. Además, no se detecta ningún tipo de delay en el desplazamiento en el mapa o en el pop-up.

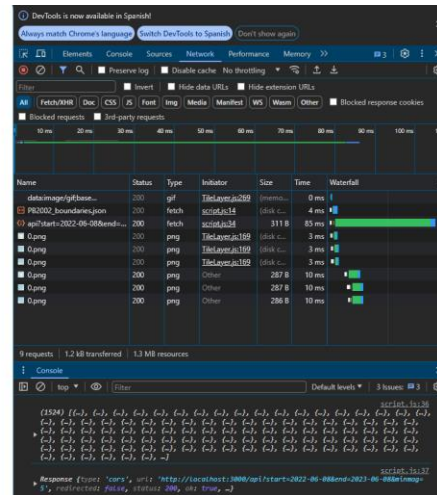


Ilustración 15. Caso de estudio 4.

5.5 Conclusiones

Una vez realizados distintos casos de prueba, se puede observar que el tiempo de respuesta del servidor es alto dependiendo de la cantidad de terremotos que recopile. Además, podemos denotar un límite de terremotos con los cuales la aplicación web funcione de manera ideal, siendo estos unos 25.000 terremotos, así la aplicación no tendrá estos tiempos de delay en el desplazamiento por el mapa y en el pop-up.

CAPÍTULO 6. CONCLUSIONES Y TRABAJOS FUTUROS

En general, los objetivos planteados desde un principio para el proyecto se lograron cumplir satisfactoriamente, con excepción del despliegue de la aplicación web. Este objetivo del proyecto resultó ser más complejo de lo previsto, ya que, además de desplegar el aplicativo web, era necesario también desplegar el servidor y la base de datos. Estas tareas implicaban una serie de conocimientos y habilidades que, en ese momento, superaban nuestras capacidades técnicas y experiencia.

6.1 Logros y Resultados

1. **Desarrollo del Frontend y Backend:** Se completó exitosamente el desarrollo del frontend utilizando JavaScript y HTML y del backend con Node.js y Express.
2. **Implementación de la Base de Datos:** Se diseñó y pobló una base de datos en MySQL.
3. **Interactividad y Visualización:** Se logró una visualización efectiva de los terremotos en un mapa interactivo utilizando Leaflet, brindando a los usuarios una experiencia de navegación intuitiva y rica en información.

6.2 Desafíos y Aprendizajes

1. **Despliegue de la Aplicación:** Uno de los mayores desafíos fue el despliegue de la aplicación. Esta tarea no se completó debido a la complejidad añadida de configurar y mantener un servidor y una base de datos en un entorno en línea, pero sí que se pudo desplegar en el entorno local.
2. **Competencias Adquiridas:** Durante el desarrollo del Trabajo de Fin de Grado, se pudieron poner en práctica una serie de competencias adquiridas a lo largo de la formación académica, tales como el diseño y desarrollo de software, la gestión de bases de datos, y la integración de diferentes tecnologías.
3. **Nuevos Conocimientos:** Además de aplicar conocimientos previos, se adquirieron nuevas habilidades y conocimientos en áreas como el desarrollo de aplicaciones web modernas, el uso de bibliotecas de JavaScript avanzadas para la visualización de los datos de los terremotos, específicamente sus coordenadas, y la implementación de APIs RESTful.

Este proyecto ha sido una experiencia invaluable para comprender los retos y las complejidades asociadas al desarrollo de aplicaciones web completas. Ha resaltado la importancia de una planificación detallada y de la adquisición continua de nuevas habilidades técnicas. Aunque no se logró el despliegue final, los conocimientos y experiencias obtenidos sentarán una base sólida para futuros proyectos.

6.3 Trabajo Futuro:

1. **Despliegue Completo:** Una dirección futura clara sería completar el despliegue de la aplicación.
2. **Optimización y Mejora:** Continuar optimizando el rendimiento de la aplicación y la base de datos para manejar mayores volúmenes de datos y mejorar la experiencia del usuario.
3. **Funcionalidades Adicionales:** Implementar funcionalidades adicionales basadas en el feedback de los usuarios, como alertas de terremotos en tiempo real, análisis predictivo, y visualización de datos históricos en diferentes formatos.

En resumen, el proyecto ha cumplido con éxito los objetivos fundamentales y ha servido como una plataforma de aprendizaje y desarrollo profesional, demostrando la importancia de la integración efectiva de diversas tecnologías y destacando áreas para crecimiento y mejora continua.

CAPÍTULO 7. ANÁLISIS DE IMPACTO

En este capítulo se analizará el impacto potencial del trabajo desarrollado en el TFG y se evaluará su relación con los Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030.

Teniendo en cuenta los diferentes contextos para analizar el trabajo desarrollado, el contexto donde podría tener un impacto significativo sería el económico ya que con la mejora de la gestión de la información sísmica se podrían reducir las pérdidas económicas causadas por los terremotos debido a que la población estaría mas informada de un posible terremoto gracias al trabajo que realizaran los sismólogos con esta nueva herramienta.

En cuanto a los Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030, el TFG podría estar alineado con los siguientes:

ODS 3: Salud y bienestar: La aplicación puede contribuir a mejorar la preparación y la respuesta ante los terremotos, lo que puede salvar vidas y reducir el número de heridos.


ODS 11: Ciudades y comunidades sostenibles: La aplicación puede contribuir a la reducción de los riesgos asociados a los terremotos en las ciudades, lo que puede hacerlas más sostenibles y resilientes.

ODS 13: Acción por el clima: La mejor gestión de la información sísmica puede contribuir a la reducción de los riesgos asociados a los terremotos, lo que puede ayudar a mitigar los efectos del cambio climático.

BIBLIOGRAFÍA

- [1] Web del Servicio Geológico de Estados Unidos (USGS) (2024) dirección: <https://www.usgs.gov/programs/earthquake-hazards> (visitado 02-06-2024)
- [2] Portal del Instituto Geográfico nacional (Catálogo de terremotos) dirección: <https://www.ign.es/web/ign/portal/sis-catalogo-terremotos> (visitado 02-06-2024)
- [3] Visualizador de terremotos del Instituto Geográfico nacional dirección: <https://visualizadores.ign.es/tproximos/> (visitado 02-06-2024)
- [4] GEOMAP visualizador de terremotos dirección: <https://www.geomap.com/es/terremotos#zoom=1&lat=80.2&lon=-114.7&layer=2&overlays=TFFFFFFF> (visitado 02-06-2024)
- [5] Arsys explicación arquitectura cliente servidor dirección: https://www.arsys.es/blog/todo-sobre-la-arquitectura-cliente-servidor#Ventajas_del_modelo_cliente-servidor (visitado 02-06-2024)
- [6] MDN Web Docs (JavaScript) dirección: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (visitado 02-06-2024)
- [7] MDN Web Docs (HTML) dirección: <https://developer.mozilla.org/en-US/docs/Web/HTML> (visitado 02-06-2024)
- [8] Leaflet dirección: <https://leafletjs.com/examples/quick-start/> (visitado 02-06-2024)
- [9] Bootstrap dirección: <https://getbootstrap.com/docs/5.2/getting-started/introduction/> (visitado 02-06-2024)
- [10] MDN Web Docs (Fetch API) dirección: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API (visitado 02-06-2024)
- [11] Node.js dirección: <https://nodejs.org/en> (visitado 02-06-2024)
- [12] Express.js dirección: <https://expressjs.com> (visitado 02-06-2024)
- [13] MySQL dirección: <https://www.mysql.com> (visitado 02-06-2024)

Este documento esta firmado por



Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
Fecha/Hora	Mon Jun 03 12:36:12 CEST 2024
Emisor del Certificado	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
Numero de Serie	561
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)