



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE COMPUTO

PROYECTO

Informe de implementación

presenta:

Balderas Hernández David Vadhir

Díaz González Lizeth

Hernández García Jaime Gabriel

Meza Bravo Iván Marcelino

Ingeniería de Software

Fecha

04/04/2025



Informe de Implementación

1. Decisiones técnicas tomadas

Para el desarrollo del sistema de alerta y visualización de sismos, se optó por una arquitectura modular y escalable, basada en las siguientes decisiones técnicas:

- **Adquisición de datos:** Se eligió consumir datos del Servicio Sismológico Nacional (SSN) utilizando su API oficial cuando esté disponible, y como alternativa, la carga de archivos CSV estructurados.
- **Modelado del conocimiento:** Se empleó una base de datos orientada a grafos (como Neo4j) para representar entidades clave como magnitud, profundidad y localización, y sus relaciones.
- **Visualización geográfica:** Se integró una API de mapas interactivos (como Leaflet.js o Google Maps API) para representar visualmente tanto la ubicación de los eventos como sus conexiones.
- **Frontend responsivo:** Se utilizó una interfaz web moderna con HTML5, CSS3 y frameworks como React o Vue.js para asegurar compatibilidad móvil y una experiencia fluida.
- **Backend y procesamiento:** Para los cálculos de trilateración y análisis histórico, se utilizó Python con bibliotecas como NumPy, Pandas y Scikit-learn.

2. Problemas encontrados y soluciones implementadas

Problema: La precisión de los datos de ubicación al trabajar con archivos CSV del SSN varía según la fuente.

- **Solución:** Se implementó un módulo de validación y limpieza de datos que estandariza y corrige posibles errores de formato.

Problema: Las operaciones de trilateración y graficación en tiempo real causaban ralentizaciones.

- Solución: Se optimizó el rendimiento usando procesamiento por lotes y almacenamiento en caché para eventos históricos.

Problema: El despliegue en dispositivos móviles mostraba fallos de escalabilidad en el mapa interactivo.

- Solución: Se utilizó diseño responsivo y se limitaron las entidades visibles simultáneamente para mantener la fluidez.

Problema: Dificultad en la interpretación de los grafos por parte de los usuarios no técnicos.

- Solución: Se agregó una sección de ayuda interactiva y un panel lateral con descripciones detalladas por nodo.

3. Tecnologías y patrones utilizados

- Frontend: HTML5, CSS3, JavaScript (React.js o Vue.js), Leaflet.js / Google Maps API.
- Backend: Python, Flask / Django.
- Base de datos: Neo4j (para grafos), PostgreSQL (para registros históricos).
- Procesamiento de datos: Pandas, NumPy, Scikit-learn.

Patrones de diseño:

- MVC (Modelo-Vista-Controlador)
- Repository Pattern para la abstracción de bases de datos
- Observer para notificaciones automáticas ante eventos relevantes

4. Posibles mejoras y trabajos futuros

Predicción avanzada: Implementar modelos de aprendizaje automático más robustos (como redes neuronales recurrentes) para mejorar la predicción de sismos.

Mejoras en accesibilidad: Añadir soporte de voz y navegación por teclado para mejorar la inclusión.

Multilinguaje: Finalizar el soporte completo para español e inglés, incluyendo manuales y mensajes del sistema.

Cumplimiento legal: Integrar validaciones más estrictas sobre el tratamiento de datos personales para cumplir con GDPR y normativas locales.