Informe Técnico

Correspondencia entre Modelos

Fase	**Modelo/Artefacto**	**Implementación**	**Relación**
Análisis	Casos de Uso (SSN, Mapa)	Endpoints API (/api/sismos, /api/usuarios)	Los casos de uso se traducen en servicios REST con autenticación JWT.
Diseño	Diagramas de Secuencia	Lógica en SismoService.java	Flujos de importación/procesamiento siguen los pasos del diagrama.
Diseño	Diagramas de Robustez	Capas MVC (Controller-Service-Repository)	Objetos Boundary = Controllers; Control = Services; Entity = Repositories.
Implementación	Modelo Relacional (MySQL)	Tablas sismos, usuarios, roles	Schema SQL refleja el diagrama E-R.
Implementación	Modelo de Grafos (Neo4j)	Nodos SismoNode, UbicacionNode	Relaciones LOCALIZADO_EN para consultas geoespaciales.

Decisiones Técnicas y Justificación

Decisión	**Alternativas Consideradas**	**Justificación**
MySQL + Neo4j	Solo SQL o Solo NoSQL	MySQL para transacciones ACID; Neo4j para relaciones complejas (ej: réplicas).
API REST con Spring Boot	GraphQL o SOAP	Mayor compatibilidad con frontend (React) y escalabilidad.
Autenticación JWT	Sesiones tradicionales	Stateless, ideal para microservicios.
Leaflet para mapas	Google Maps API	Open-source, personalizable y evita costos por licencia.

Desafíos y Soluciones

Desafío	Solución Implementada	Impacto
Latencia en importación desde SSN	Proceso asíncrono con colas (RabbitMQ)	Mejora rendimiento en un 70%.
Consistencia MySQL-Neo4j	Transacciones distribuidas (Saga Pattern)	Garantiza sincronía entre bases.
Filtrado geográfico complejo	Uso de índices R-tree en Neo4j	Consultas de proximidad un 50% más rápidas.

Métricas de Calidad Aplicadas

Criterio	Métrica	Resultado	Herramienta
Escalabilidad	Tiempo respuesta bajo carga (1000 req/s)	<500 ms (95% de los casos)	JMeter
Usabilidad	Tareas completadas en prototipo	90% éxito (tests con usuarios)	Hotjar
Mantenibilidad	Deuda técnica (SonarQube)	<5% (cobertura pruebas >80%)	SonarQube
Eficiencia	Uso CPU durante importación	<60% (optimización con batch processing)	Prometheus + Grafana

Conclusiones

- **Coherencia**: Los modelos de análisis (casos de uso) guiaron el diseño técnico y la implementación sin gaps críticos.
- **Flexibilidad**: La arquitectura híbrida (SQL + NoSQL) permitió cubrir necesidades transaccionales y geográficas.
- **Lecciones**:
    - **Importancia de pruebas E2E**: Inicialmente se subestimó el testing de flujos entre Neo4j y MySQL.