



INSTITUTO POLITÉCNICO NACIONAL

**ESCUELA SUPERIOR DE
COMPUTACIÓN**

Sistema de búsqueda y recomendación de libros

Unidad de aprendizaje:

Ingeniería de Software

6CV3

Elaborado por:

Antony Horteales Morales

Pablo Daniel Granados Martínez

Isaac Pardo Gómez

Luis Gerardo Herrera Avila

Profesora:

Gabriel Hurtado Avilés

INSTITUTO POLITÉCNICO NACIONAL



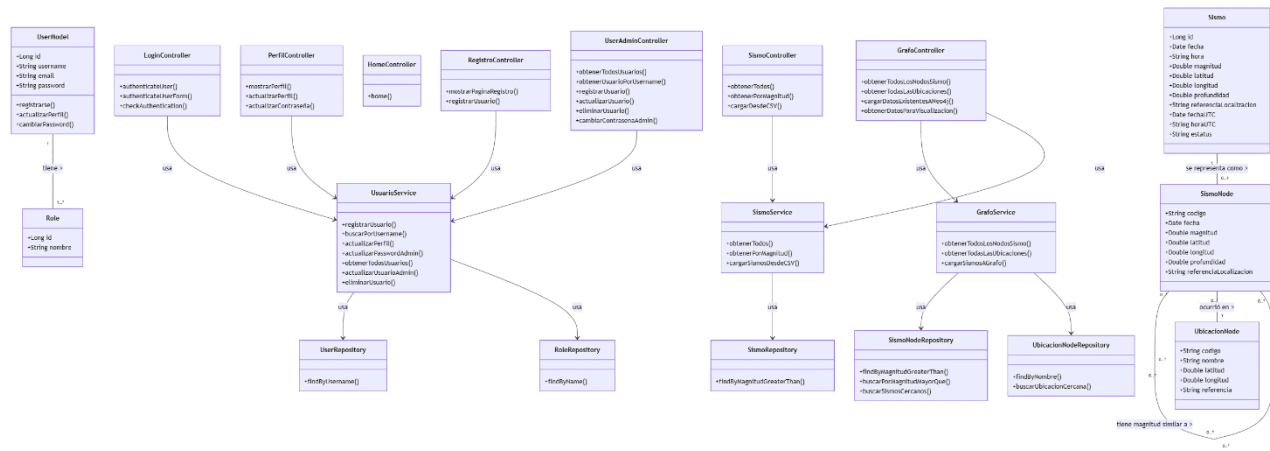
ESCOM

28 de Abril de 2025

Índice

Diagrama de Clases Conceptuales.....	1
Modelo de Dominio.....	3
Diccionario de Datos	7
Diagrama Entidad-Relación	11
Diagramas de Secuencia.....	12
Diagramas de Robustez	14
Modelo de Interfaz y Navegación	16
Diagrama de Clases de Diseño	21
Patrón de Diseño	22
Modelo de Implementación	23
Diagrama de Despliegue.....	24

Diagrama de Clases Conceptuales



Clases y Atributos Principales

- **UserModel**
 - *Atributos:* id, username, email, password
 - *Métodos:* registrarse(), actualizarPerfil(), cambiarPassword()
- **Role**
 - *Atributos:* id, nombre
- **Sismo**
 - *Atributos:* id, fecha, hora, magnitud, latitud, longitud, profundidad, referenciaLocalizacion, fechaUTC, horaUTC, estatus
- **SismoNode**
 - *Atributos:* codigo, fecha, magnitud, latitud, longitud, profundidad, referenciaLocalizacion
- **UbicacionNode**
 - *Atributos:* codigo, nombre, latitud, longitud, referencia
- **Controladores:**

- *GrafoController*: obtenerTodosLosNodosSismo(), obtenerTodasLasUbicaciones(), cargarDatosExistentesANeo4j(), obtenerDatosParaVisualizacion()
- *HomeController*: home()
- *LoginController*: authenticateUser(), authenticateUserForm(), checkAuthentication()
- *PerfilController*: mostrarPerfil(), actualizarPerfil(), actualizarContraseña()
- *RegistroController*: mostrarPaginaRegistro(), registrarUsuario()
- *SismoController*: obtenerTodos(), obtenerPorMagnitud(), cargarDesdeCSV()
- *UserAdminController*: obtenerTodosUsuarios(), obtenerUsuarioPorUsername(), registrarUsuario(), actualizarUsuario(), eliminarUsuario(), cambiarContraseñaAdmin()

- **Repositorios:**

- *RoleRepository*: findByName()
- *SismoRepository*: findByMagnitudGreaterThan()
- *UserRepository*: findByUsername()
- *SismoNodeRepository*: findByMagnitudGreaterThan(), buscarPorMagnitudMayorQue(), buscarSismosCercanos()
- *UbicacionNodeRepository*: findByName(), buscarUbicacionCercana()

- **Servicios:**

- *UsuarioService*: registrarUsuario(), buscarPorUsername(), actualizarPerfil(), actualizarPasswordAdmin(), obtenerTodosUsuarios(), actualizarUsuarioAdmin(), eliminarUsuario()
- *SismoService*: obtenerTodos(), obtenerPorMagnitud(), cargarSismosDesdeCSV()
- *GrafoService*: obtenerTodosLosNodosSismo(), obtenerTodasLasUbicaciones(), cargarSismosAGrafo()

Relaciones entre Clases

- **UserModel** tiene una relación de asociación de 1 a muchos ("1" -- "1..*") con **Role**.

- **Sismo** tiene una asociación de "1" a "0..1" con **SismoNode** (se representa como).
- **SismoNode** se relaciona consigo mismo en dos relaciones múltiples ("0.." -- "0.."):
 - Una por cercanía geográfica (*es cercano a*).
 - Otra por similitud de magnitud (*tiene magnitud similar a*).
- **SismoNode** está asociado a **UbicacionNode** en una relación de "0..*" a "1" (*ocurrió en*).

Cardinalidades

Las cardinalidades especificadas son:

- Un **UserModel** puede tener uno o varios (**1..***) roles asociados.
- Un **Sismo** puede representarse o no (**0..1**) por un **SismoNode**.
- Cada **SismoNode** puede estar relacionado con múltiples otros **SismoNode** (por cercanía y magnitud similar).
- Cada **SismoNode** está relacionado obligatoriamente con una única (**1**) **UbicacionNode**.

Modelo de Dominio

Entidades Principales y sus Reglas de Negocio

1. Usuario

Descripción: Representa a los usuarios del sistema con sus credenciales y roles.

Reglas de Negocio:

- Cada usuario debe tener un nombre de usuario único
- El email debe tener un formato válido y ser único en el sistema
- La contraseña debe almacenarse encriptada (BCrypt)
- Los usuarios pueden tener roles de "USER" o "ADMIN"
- Solo los administradores pueden gestionar otros usuarios
- Un usuario puede actualizar su propio perfil pero no su rol

Invariantes:

- username no puede ser nulo o vacío
- email debe contener un formato válido (contener '@' y '.')
- password debe tener al menos 8 caracteres
- Cada usuario debe tener al menos un rol asignado

2. Rol

Descripción: Define los roles de seguridad del sistema.

Reglas de Negocio:

- Existen dos roles principales: ROLE_ADMIN y ROLE_USER
- Los roles son asignados durante el registro o por un administrador
- Los roles no pueden ser eliminados si hay usuarios asignados

Invariantes:

- El nombre del rol debe comenzar con "ROLE_"
- No pueden existir roles duplicados

3. Sismo

Descripción: Representa un evento sísmico con sus características principales.

Reglas de Negocio:

- Cada sismo debe tener un código único identificador
- La magnitud debe ser un valor positivo
- La fecha debe ser válida y no futura
- Los sismos pueden cargarse individualmente o mediante archivos CSV
- Los sismos con magnitud >7 deben ser notificados a los usuarios relevantes

Invariantes:

- codigo debe ser único

- magnitud debe estar entre 0 y 10
- fecha no puede ser futura
- latitud debe estar entre -90 y 90
- longitud debe estar entre -180 y 180

4. SismoNode

Descripción: Representación de un sismo en la base de datos de grafos.

Reglas de Negocio:

- Cada SismoNode debe corresponder a un Sismo en la base relacional
- Los nodos pueden tener relaciones de cercanía o similitud
- Las relaciones se establecen automáticamente basadas en proximidad geográfica y magnitud similar

Invariantes:

- Debe existir correspondencia con un Sismo en la base relacional
- Las relaciones "CERCANO_A" deben tener distancia < 50km
- Las relaciones "SIMILAR_MAGNITUD" deben tener diferencia < 0.5 en magnitud

5. Nodo de Ubicación en Grafo

Descripción: Representa ubicaciones geográficas donde ocurren sismos.

Reglas de Negocio:

- Las ubicaciones se crean automáticamente al cargar sismos
- Varios sismos pueden estar relacionados a una misma ubicación
- Las ubicaciones se agrupan por región geográfica

Invariantes:

- Cada ubicación debe tener un código único
- El nombre no puede ser nulo

- Las coordenadas deben ser válidas

Invariantes del Sistema

1. Seguridad:

- Todas las solicitudes a endpoints protegidos deben estar autenticadas
- Las contraseñas nunca se almacenan en texto plano
- Las sesiones tienen un máximo de 1 sesión activa por usuario

2. Consistencia de Datos:

- Cada SismoNode en Neo4j debe corresponder a un Sismo en MySQL
- Las actualizaciones en usuarios deben reflejarse en ambos sistemas de autenticación
- No pueden existir usuarios sin rol asignado

3. Integridad Geográfica:

- Las coordenadas de sismos deben ser válidas
- Las relaciones de cercanía deben cumplir con umbrales geográficos definidos

Restricciones y Condiciones de los Datos

1. Restricciones de Usuario:

- Solo los administradores pueden acceder a /api/admin/**
- Los usuarios regulares solo pueden modificar su propio perfil
- Las contraseñas deben cumplir con política de complejidad (mínimo 8 caracteres)

2. Restricciones de Sismos:

- Los sismos no pueden eliminarse, solo marcarse como inactivos
- La carga masiva de sismos se realiza asincrónicamente
- Los sismos con magnitud >7 generan relaciones automáticas en el grafo

3. Restricciones de Grafos:

- Las relaciones entre nodos se generan automáticamente según reglas predefinidas

- La visualización del grafo puede filtrarse por magnitud
- Los nodos inactivos no se muestran en las visualizaciones

Diccionario de Datos

Entidad: Sismos

Nombre de la entidad: sismos

Descripción y propósito:

Almacena información sobre eventos sísmicos, incluyendo magnitud, ubicación, fecha y hora. Esta es la entidad central del sistema que contiene los datos principales para el seguimiento y análisis de sismos.

Atributos:

- id (bigint): Identificador único del sismo (clave primaria)
- fecha (date): Fecha del evento sísmico (obligatorio)
- hora (varchar(255)): Hora local del evento sísmico
- magnitud (double): Intensidad del sismo en escala correspondiente
- latitud (double): Coordenada de latitud donde ocurrió el sismo (obligatorio)
- longitud (double): Coordenada de longitud donde ocurrió el sismo (obligatorio)
- profundidad (double): Profundidad del epicentro del sismo (obligatorio)
- referencia_localizacion (varchar(255)): Descripción textual de la ubicación
- fecha_utc (date): Fecha del evento en tiempo UTC (obligatorio)
- hora_utc (varchar(255)): Hora del evento en tiempo UTC
- estatus (varchar(255)): Estado o clasificación del evento sísmico
- fechautc (date): Campo duplicado de fecha UTC (posible redundancia)
- horautc (varchar(255)): Campo duplicado de hora UTC (posible redundancia)

Restricciones:

- Clave primaria: id con auto-incremento
- Índices:
 - idx_sismos_magnitud: Índice para búsquedas por magnitud
 - idx_sismos_fecha: Índice para búsquedas por fecha
 - idx_sismos_ubicacion: Índice compuesto para búsquedas por ubicación (latitud, longitud)

Relaciones:

- En la base de datos relacional: No tiene relaciones explícitas definidas en el esquema SQL
- En Neo4j (base de datos de grafos): Representado como SismoNode que puede tener relaciones con UbicacionNode

Entidad: Usuarios

Nombre de la entidad: usuarios

Descripción y propósito:

Almacena información sobre los usuarios del sistema, permitiendo la autenticación y autorización para el acceso a las funcionalidades del sistema.

Atributos:

- id (bigint): Identificador único del usuario (clave primaria)
- nombre (varchar(255)): Nombre del usuario
- email (varchar(64)): Correo electrónico del usuario (obligatorio)
- password (varchar(255)): Contraseña del usuario (almacenada de forma segura)

Restricciones:

- Clave primaria: id con auto-incremento
- Restricción de unicidad: email debe ser único en el sistema

Relaciones:

- Relación muchos a muchos con la entidad roles a través de la tabla usuario_rols

Entidad: Roles

Nombre de la entidad: roles

Descripción y propósito:

Define los diferentes tipos de roles o permisos que pueden tener los usuarios en el sistema, permitiendo implementar un control de acceso basado en roles.

Atributos:

- id (bigint): Identificador único del rol (clave primaria)
- nombre (varchar(64)): Nombre del rol (obligatorio)

Restricciones:

- Clave primaria: id con auto-incremento
- Restricción de unicidad: nombre debe ser único

Relaciones:

- Relación muchos a muchos con la entidad usuarios a través de la tabla usuario_roles

Entidad asociativa: Usuario_Roles

Nombre de la entidad: usuario_roles

Descripción y propósito:

Tabla de enlace que implementa la relación muchos a muchos entre usuarios y roles, permitiendo asignar múltiples roles a cada usuario.

Atributos:

- usuario_id (bigint): Referencia al ID del usuario (parte de la clave primaria)
- rol_id (bigint): Referencia al ID del rol (parte de la clave primaria)

Restricciones:

- Clave primaria compuesta: (usuario_id, rol_id)
- Claves foráneas:
 - usuario_id referencia usuarios(id)
 - rol_id referencia roles(id)

Relaciones:

- Relaciona las entidades usuarios y roles

Entidad: SismoNode (Neo4j)

Nombre de la entidad: SismoNode

Descripción y propósito:

Representación en grafo de los eventos sísmicos, permitiendo realizar análisis de relaciones entre sismos y ubicaciones que serían más complejos en un modelo relacional.

Atributos:

- Hereda atributos similares a la entidad sismos en MySQL
- Implementado en Neo4j, no en el esquema SQL proporcionado

Restricciones:

- Identificación por IDs únicos en Neo4j

Relaciones:

- Relaciones con nodos UbicacionNode para análisis espacial y geográfico

Entidad: UbicacionNode (Neo4j)

Nombre de la entidad: UbicacionNode

Descripción y propósito:

Representación en grafo de ubicaciones geográficas relevantes para el análisis sísmico, permitiendo modelar relaciones complejas entre ubicaciones y sismos.

Atributos:

- Posiblemente incluye coordenadas, nombres de región, población afectada, etc.
- Implementado en Neo4j, no en el esquema SQL proporcionado

Restricciones:

- Identificación por IDs únicos en Neo4j

Relaciones:

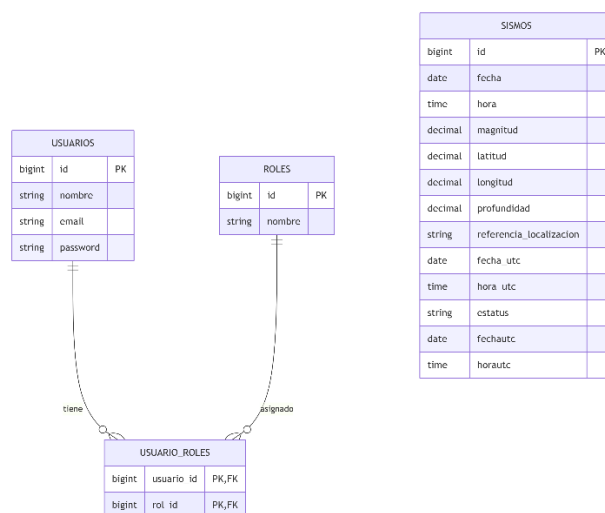
- Relaciones con nodos SismoNode para análisis espacial y geográfico

Diagrama Entidad-Relación

Estructura de la Base de Datos Relacional (MySQL)

El sistema utiliza un modelo relacional compuesto por las siguientes tablas:

Tabla	Clave Primaria (PK)	Claves Foráneas (FK)	Descripción
usuarios	id	—	Almacena información de usuarios registrados.
roles	id	—	Define los roles disponibles en el sistema (ej: admin, usuario).
usuario_roles	(usuario_id, rol_id)	usuario_id → usuarios(id) rol_id → roles(id)	Tabla intermedia para la relación muchos-a-muchos entre usuarios y roles.
sismos	id	—	Registra datos de eventos sísmicos (magnitud, ubicación, etc.).



Relaciones entre Tablas

1. Usuarios ↔ Roles (Muchos-a-Muchos)

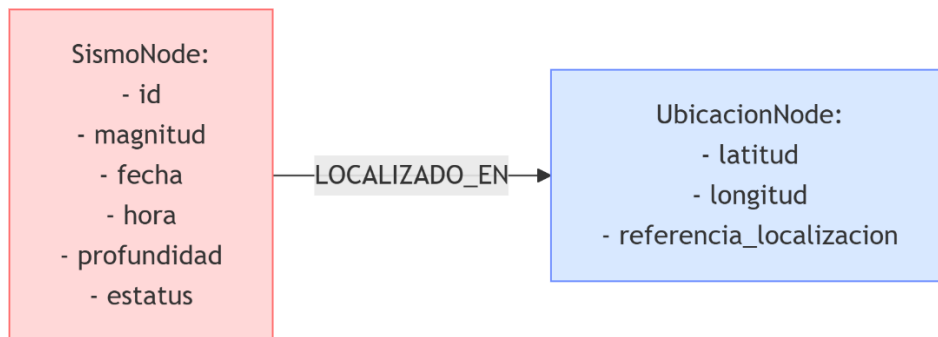
- Un usuario puede tener **múltiples roles** (ej: admin y editor).
- Un rol puede ser asignado a **varios usuarios**.
- Implementado mediante la tabla puente usuario_roles.

2. **Tabla** sismos

- **Independiente:** No tiene relaciones directas con otras tablas en este modelo.
- Contiene datos autónomos de eventos sísmicos (fecha, ubicación, magnitud).

Modelo de Grafos (Neo4j)

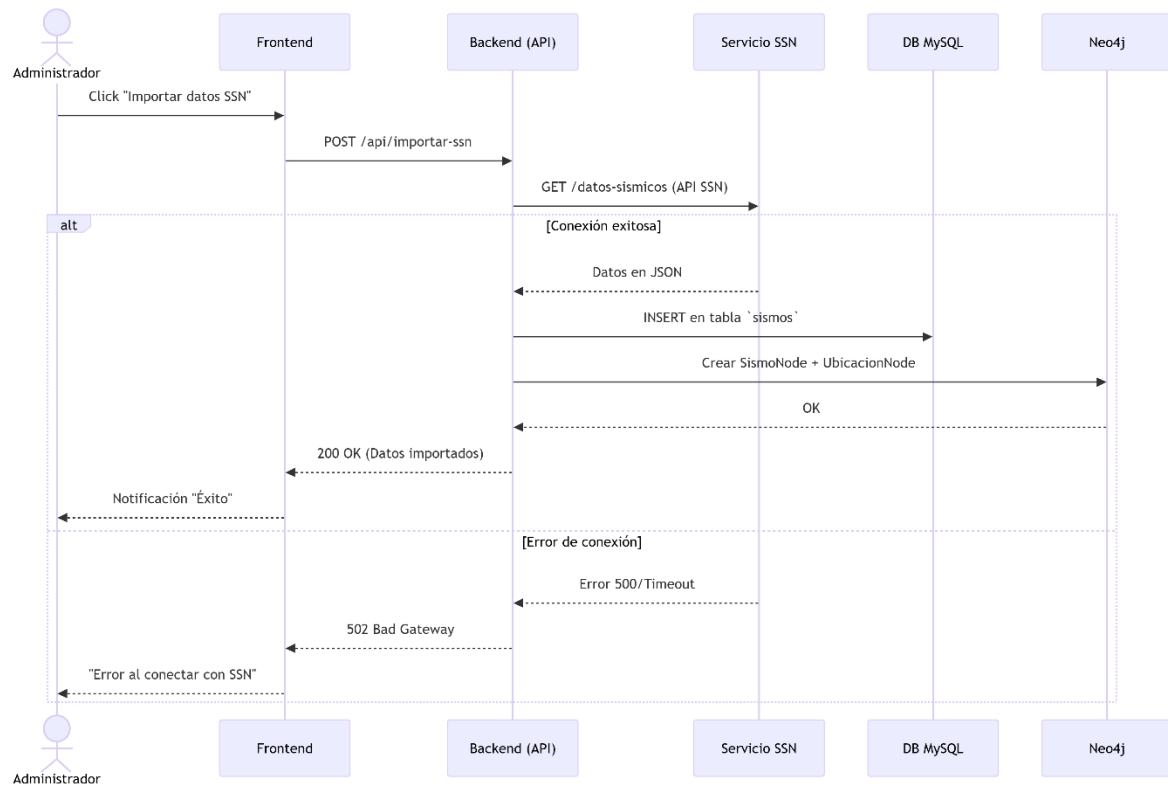
Para consultas complejas de ubicaciones y relaciones geográficas, se usa Neo4j con dos nodos principales:



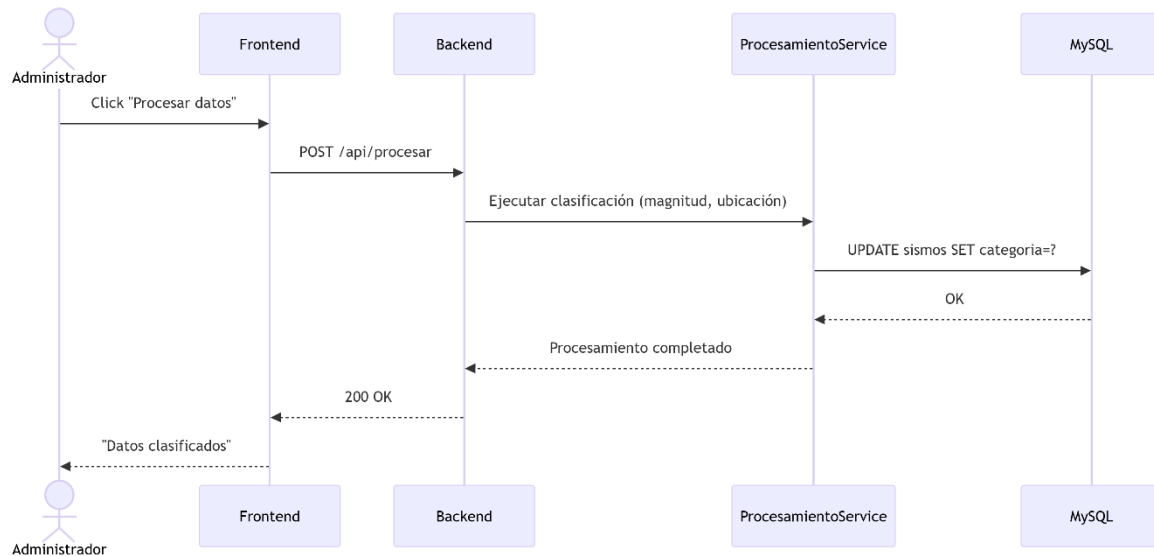
- **Relación:** Cada SismoNode está conectado a un UbicacionNode mediante la relación LOCALIZADO_EN.

Diagramas de Secuencia

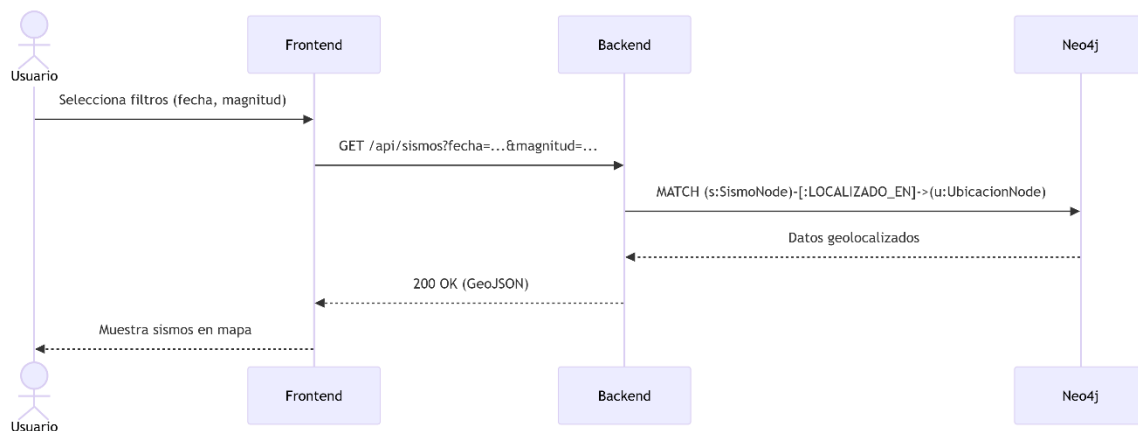
Importar datos desde el SSN



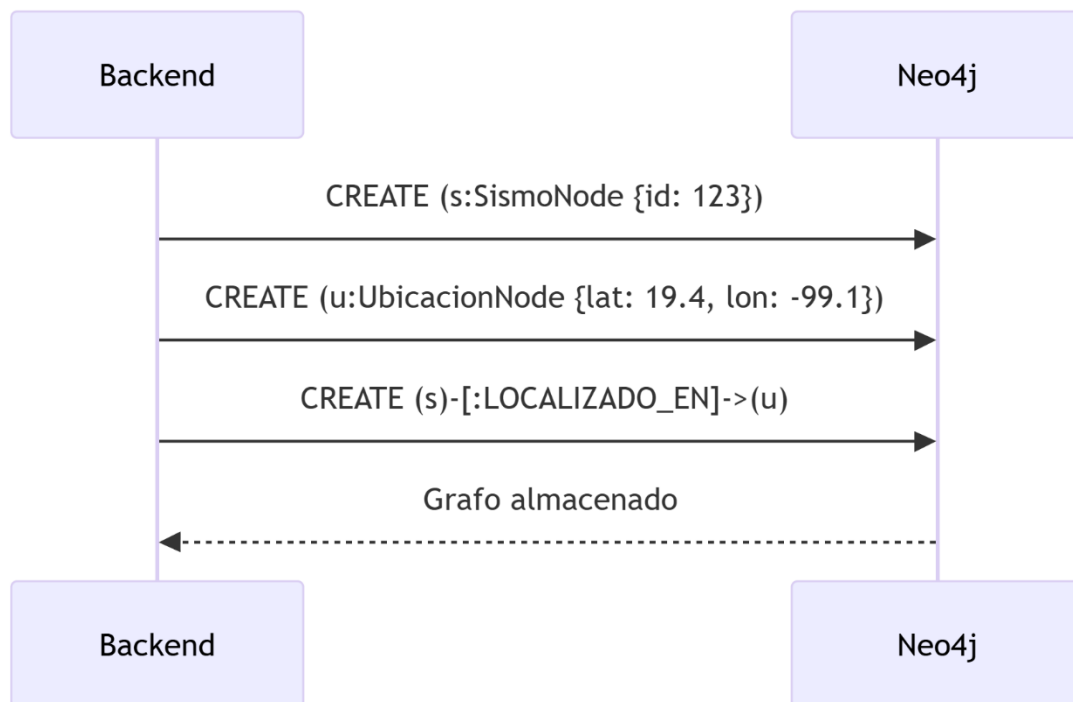
Procesar y clasificar datos sísmicos



Visualizar eventos sísmicos en el mapa

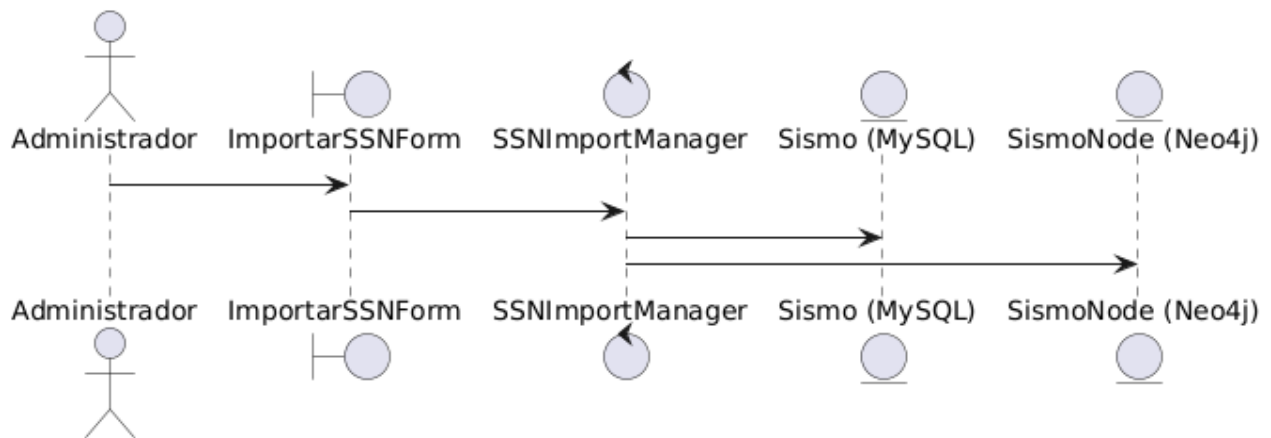


Generar grafos de conocimiento

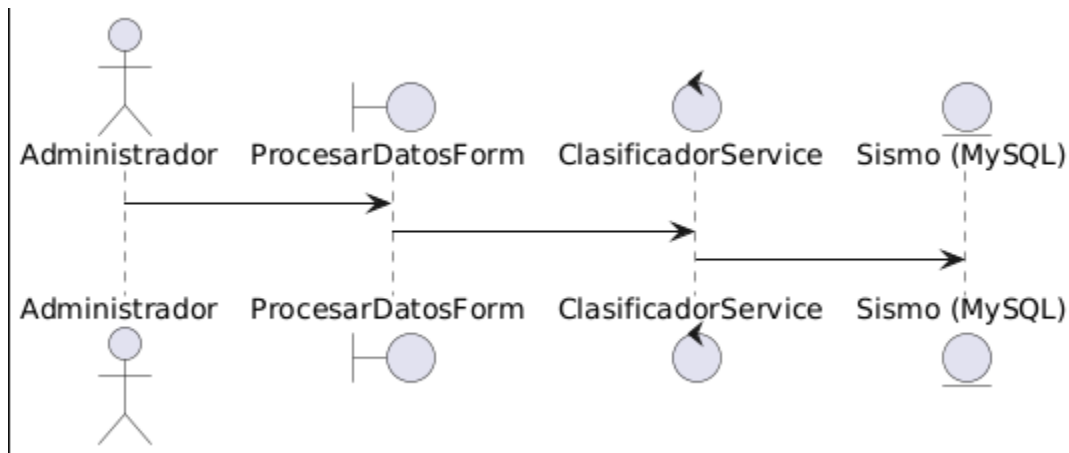


Diagramas de Robustez

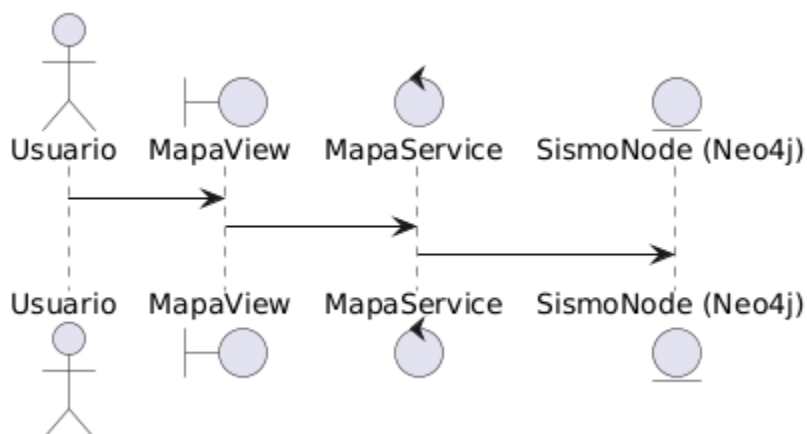
Importar datos desde el SSN



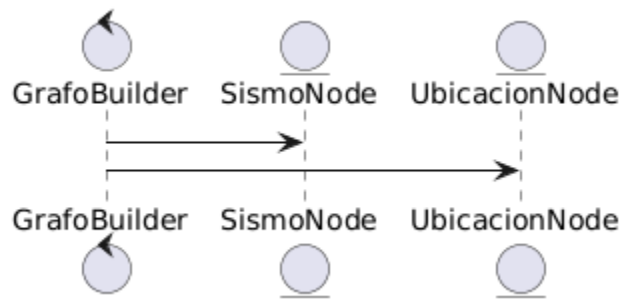
Procesar y clasificar datos sísmicos



Visualizar eventos sísmicos en el mapa

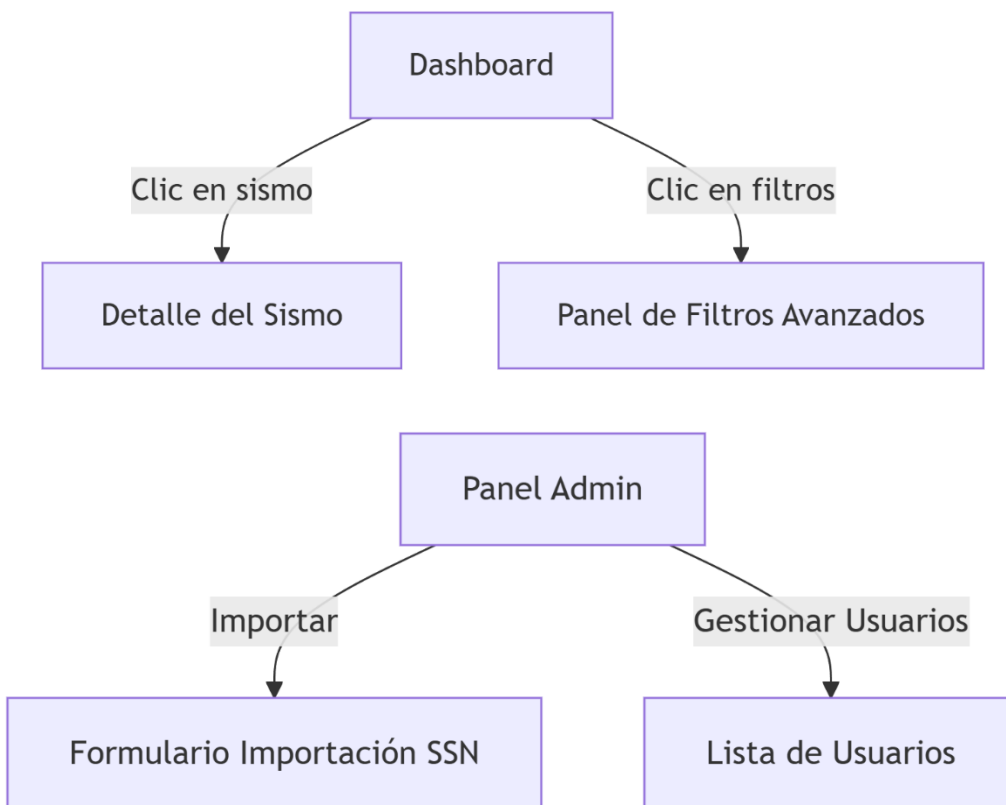


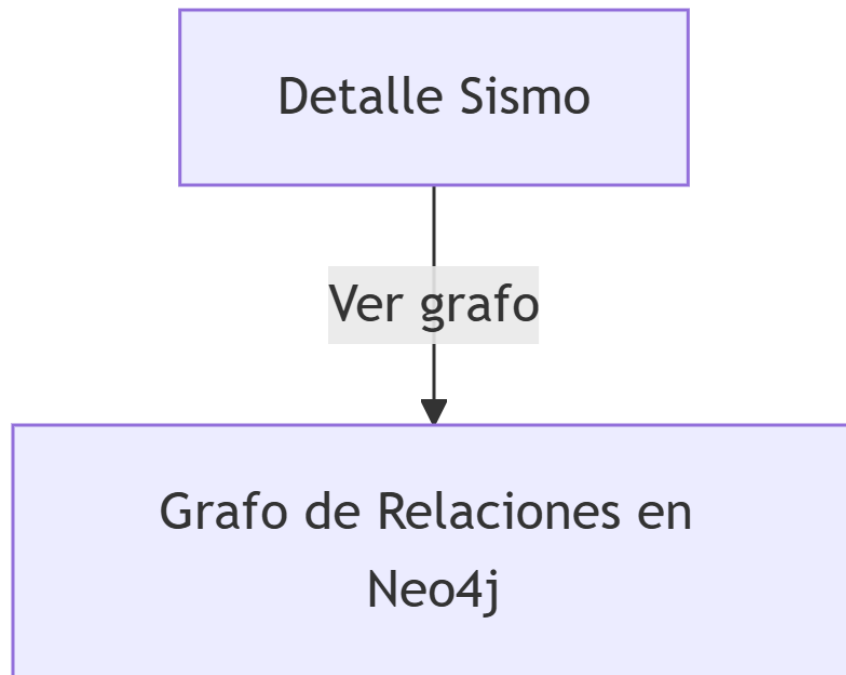
Generar grafos de conocimiento



Modelo de Interfaz y Navegación

Diseño de pantallas principales





Inicio de Sesión

Iniciar Sesión

Correo:

Contraseña:

¿Eres nuevo?

Registrarse

Registro

Registrarse

Nombre:

Correo:

Contraseña: ⓘ

Repite la contraseña:

¿Ya tienes una cuenta?

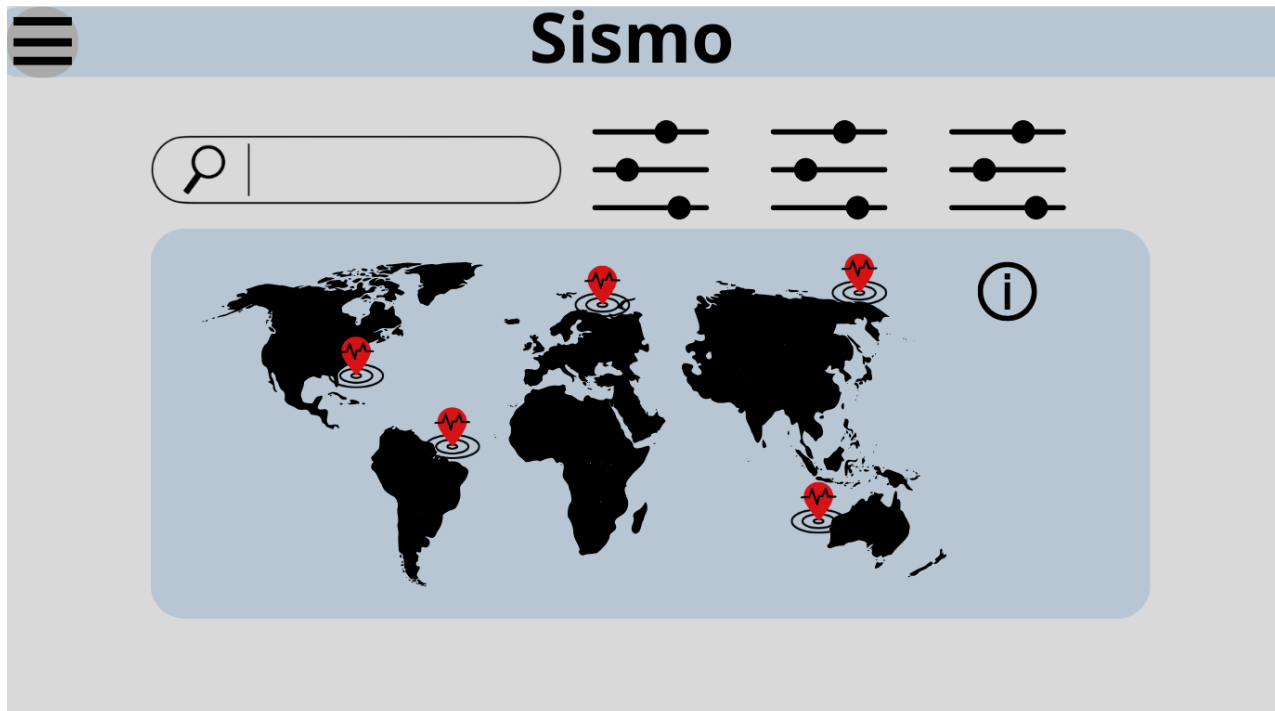
Iniciar Sesión

Gestión de Usuarios

ID	Nombre	Email	Rol	Acciones
	Nombre			
		Email:		
		Rol:		
				<div>Actualizar cuenta</div> <div>Eliminar cuenta</div>

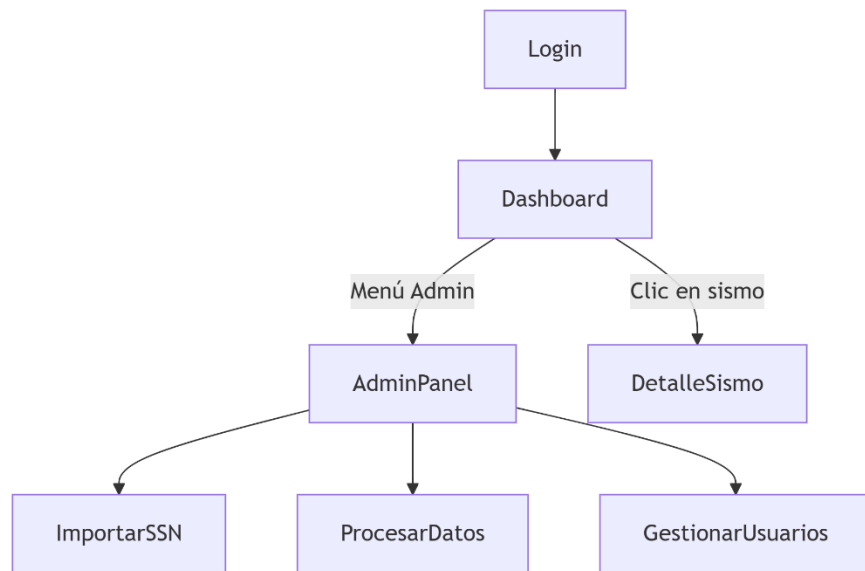
Importar desde el SSN

Agrega documento CSV:



Modelo de navegación por roles

Flujo para administrador



Flujo para usuario

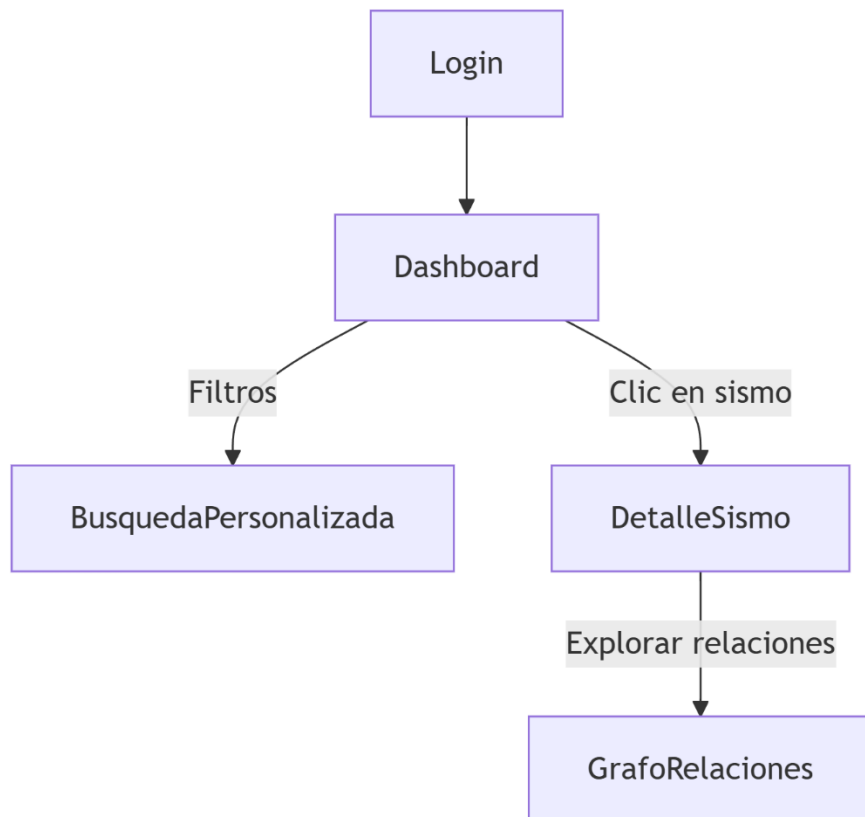
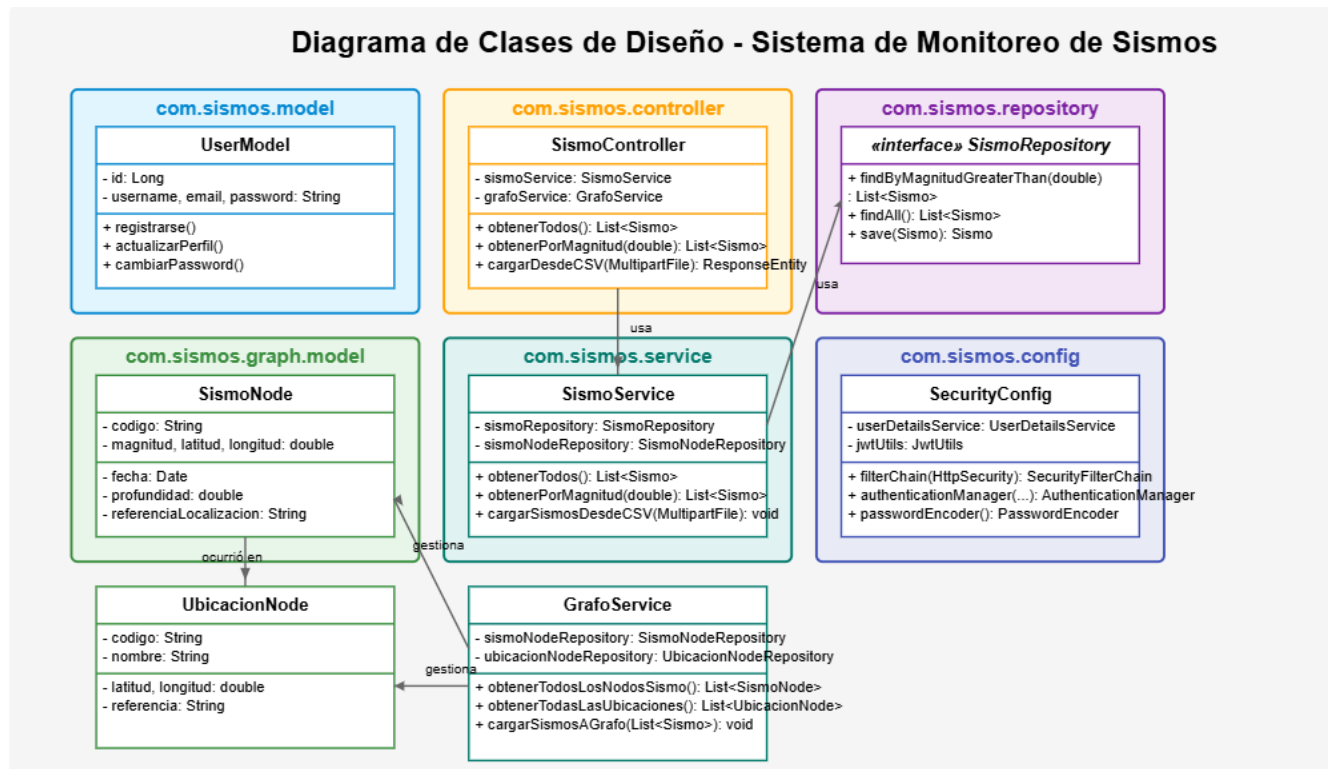
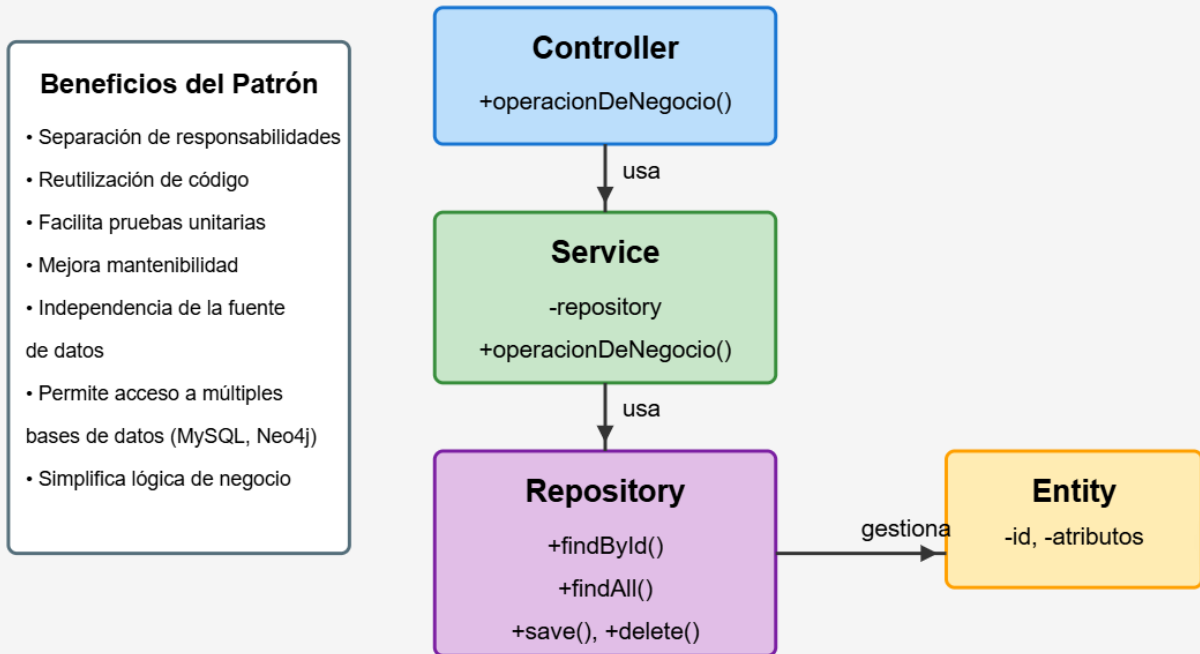


Diagrama de Clases de Diseño



Patrón de Diseño

Patrón Repository + Service Layer



Modelo de Implementación

Arquitectura Técnica - Sistema de Monitoreo de Sismos

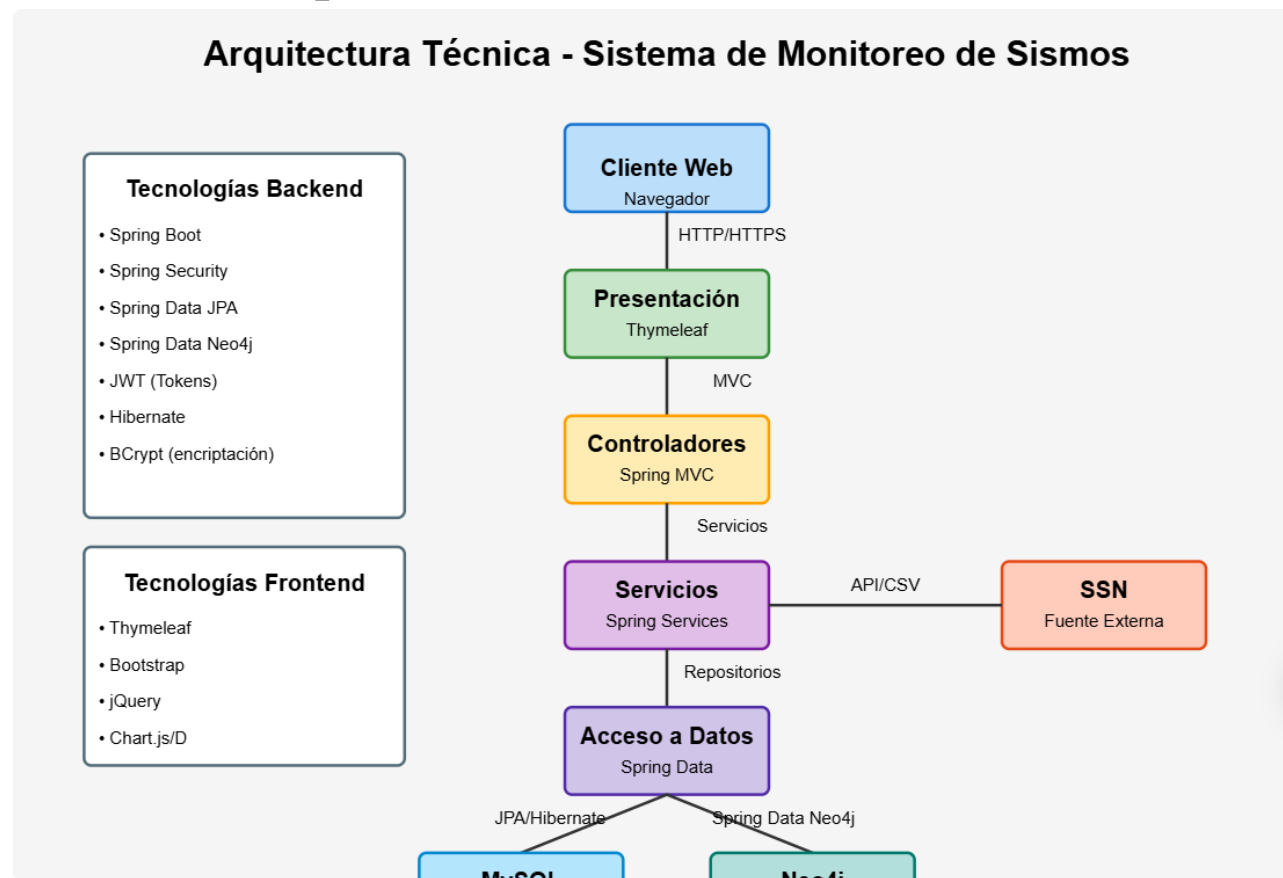


Diagrama de Despliegue

Diagrama de Despliegue - Sistema de Monitoreo de Sismos

