

INFORME DE IMPLEMENTACIÓN

Ingeniería de Software
6CV3

Fuentes Reyes Jayri Arath
Gathe Esquivel Arleth
Oviedo Lopez Joshua
Reyes Núñez Sebastián



Correspondencia entre modelos de análisis, diseño e implementación

Durante este sprint, se mantuvo la correspondencia entre los modelos de análisis y el diseño técnico planeado:

Modelo de dominio: Se siguió trabajando con la entidad Sismo, ligada directamente a la generación de reportes.

Vista: El mapa de sismos se actualizó para ofrecer opciones de interacción contextualizadas (descarga de reporte individual por sismo).

Controladores: Se integraron nuevos endpoints específicos para descargar la información de un sismo en PDF, respetando la separación de controladores para usuarios (VistaSismoController) y administradores (AdminSismoController).

Servicios: Se amplió SismoService para soportar consultas individuales de sismos, garantizando la reutilización de la lógica de acceso a datos.

Seguridad: Se actualizaron reglas de acceso en SecurityConfig para permitir que tanto usuarios como administradores pudieran descargar los PDFs, manteniendo la política de roles.

Decisiones técnicas tomadas y su justificación

Generar PDFs con iText:

Se optó por usar la librería iText para la creación dinámica de reportes en PDF debido a su flexibilidad, compatibilidad con Java 21, y la necesidad de un diseño más profesional en los documentos exportados.

Separar los endpoints de descarga:

Se crearon dos rutas de descarga (/mapa/descargar-pdf/{id} para usuarios normales y /admin/mapa/descargar-pdf/{id} para administradores) para respetar el control de roles y mantener organizada la estructura de seguridad.

Ampliación de servicios:

Se creó el método obtenerSismoPorId(Long id) en SismoService para centralizar la recuperación de registros de sismos, fomentando el principio de responsabilidad única (SRP) en el backend.

Corrección de mapeo de seguridad:

Se ajustó la configuración de SecurityConfig para que las rutas de descarga de PDF fueran accesibles tanto a ROLE_USER como a ROLE_ADMIN, solucionando errores 403 en descargas.

Desafíos encontrados y soluciones implementadas

Desafío	Solución Aplicada
Error 403 (acceso denegado) al intentar descargar un PDF	Se actualizaron explícitamente las reglas de autorización en SecurityConfig, asegurando que las rutas de descarga permitieran acceso a ambos roles (ROLE_USER y ROLE_ADMIN).
Confusión entre controladores de usuario y admin para descarga	Se estandarizó el manejo de endpoints separados y se organizó el flujo de controladores, garantizando consistencia.
Error en tipo de ID (Integer vs Long) en métodos de recuperación	Se revisaron y corrigieron las firmas de métodos para manejar Long de forma consistente entre entidad, repositorio y servicio.
Fallo por falta de importación de librería de color en iText	Se importó explícitamente com.itextpdf.kernel.color.Color para evitar conflictos con java.awt.Color.
No mostrar botón de descarga en el mapa de administrador	Se corrigió la plantilla de Thymeleaf mapa_admin.html para incluir el botón de descarga en la ventana emergente de cada marcador.

Métricas de calidad aplicadas al diseño

Responsabilidad única (SRP): Cada clase y método está enfocado en una única responsabilidad concreta.

Modularidad: Se preservó la separación clara entre Controladores, Servicios y Repositorios, facilitando la mantenibilidad.

Reutilización de código: El servicio SismoService centraliza las consultas, evitando duplicación en los controladores.

Seguridad: Las rutas sensibles están protegidas de acuerdo al rol del usuario, minimizando exposición de endpoints no autorizados.

Estilo de codificación: Se mantuvo coherencia en convenciones de nombrado, anotaciones, manejo de excepciones y uso de `@Autowired`, `@Service`, `@Controller`.