

Tarea: Implementación de OpenTelemetry en Microservicios

Objetivo: Implementar OpenTelemetry en los microservicios FutureXCourseApp y FutureXCourseCatalog utilizando tres enfoques diferentes.

Instrucciones:

Sigan las tres guías proporcionadas en orden y capturen evidencia (pantallazos) para cada paso importante. Asegúrense de documentar cualquier problema encontrado y cómo lo resolvieron.

Parte 1: Usando el Agente Java de OpenTelemetry

1. Descarguen el agente Java de OpenTelemetry.
2. Modifiquen el archivo `application.properties` en ambos servicios.
3. Ejecuten el archivo Docker Compose para Jaeger.
4. Ejecuten ambas aplicaciones con el agente Java.
5. Verifiquen las trazas en la UI de Jaeger.

Evidencia requerida:

- Pantallazo del `application.properties` modificado
- Pantallazo de la aplicación ejecutándose con el agente
- Pantallazo de las trazas en la UI de Jaeger

Parte 2: Usando la API de OpenTelemetry

1. Actualicen las dependencias en los archivos `pom.xml`.
2. Creen la clase de configuración `OpenTelemetryConfig`.
3. Actualicen las clases principales de la aplicación.
4. Modifiquen los controladores para usar OpenTelemetry.
5. Ejecuten Jaeger y verifiquen las trazas.

Evidencia requerida:

- Pantallazo de la clase `OpenTelemetryConfig`
- Pantallazo de los controladores modificados
- Pantallazo de las trazas en Jaeger

Parte 3: Usando el Colector de OpenTelemetry

1. Configuren el Colector de OpenTelemetry.
2. Actualicen el archivo `docker-compose.yml`.

3. Modifiquen el archivo `application.properties`.
4. Actualicen la clase `OpenTelemetryConfig`.
5. Ejecuten los contenedores Docker y verifiquen las trazas.

Evidencia requerida:

- Pantallazo del archivo de configuración del Colector (yaml)
- Pantallazo del `application.properties` modificado
- Pantallazo de la clase `OpenTelemetryConfig` actualizada
- Pantallazo de las trazas en Jaeger pasando por el Colector

Entregable final: Un documento con todos los pantallazos organizados por sección, incluyendo breves descripciones de cualquier desafío encontrado durante el proceso y como lo resolvieron.