

Actividad: Plataforma de Microservicios

Descripción General

En grupos de 4 estudiantes, diseñarán e implementarán una plataforma compuesta por microservicios que se desplegarán utilizando tecnologías basadas en contenedores (Dockerfile o Docker Compose). Esta plataforma deberá conectarse con **Roble**, el sistema de backend provisto por la universidad que ofrece servicios de autenticación y almacenamiento de datos.

La arquitectura debe permitir que los microservicios sean **dinámicos**: se puedan **crear, editar, eliminar y listar** desde una **interfaz web de administración** incluida en la solución.

Objetivo del Proyecto

Construir una plataforma que permita:

- Desplegar microservicios que se conectan con Roble mediante tokens de autenticación.
- Procesar datos de Roble de forma específica en cada microservicio (filtrado, agregación, análisis, etc.).
- Exponer cada microservicio como un **endpoint HTTP** que recibe parámetros y devuelve datos procesados en **formato JSON**.
- Administrar dinámicamente los microservicios a través de una **interfaz web CRUD**.

Requisitos Técnicos

- Cada microservicio debe estar contenido en una imagen Docker propia y ser independiente de los demás.
- El número de microservicios no debe estar predefinido: deben poder **agregarse o eliminarse dinámicamente**.
- Toda la solución debe ser desplegable localmente mediante un solo comando (`docker-compose up`).
- Se debe proveer un **servicio de gestión web (dashboard)** que permita:
 - Registrar un nuevo microservicio con sus parámetros (nombre, tipo de procesamiento, etc.).
 - Editar o eliminar microservicios existentes.
 - Visualizar los endpoints disponibles.

Requisitos Funcionales

- Cada microservicio debe:
 - Exponerse como un **endpoint HTTP**.
 - Recibir:
 - Un **token de autenticación válido** generado por Roble (`Authorization: Bearer <token>`).

- Parámetros adicionales vía query o body (ej. `?userId=123,{ "tipo": "avanzado" }`).
 - Usar el token para consultar Roble en nombre del usuario autenticado
 - Procesar y transformar los datos recibidos según su lógica específica.
 - Retornar una **respuesta en formato JSON**.
- La interfaz web debe permitir:
 - Crear microservicios (por ejemplo, definiendo un tipo o plantilla).
 - Editar configuraciones de microservicios.
 - Eliminar microservicios.
 - Ver el estado actual de los microservicios desplegados.
 - Hacer solicitudes de prueba a los endpoints disponibles.
- La plataforma debe manejar correctamente:
 - Tokens expirados o inválidos → 401 Unauthorized.
 - Acceso denegado → 403 Forbidden.
 - Errores internos → 500 Internal Server Error.

Criterios de Evaluación

- Diseño modular y uso adecuado de contenedores.
- Correcta interacción con la plataforma Roble.
- Documentación clara sobre la arquitectura, endpoints y despliegue.
- Originalidad en el tipo de procesamiento o funcionalidad ofrecida.
- Buenas prácticas de desarrollo (uso de `.env`, manejo de errores, logs, etc.).

Entrega

- Fin de la semana 10
- Código fuente en un repositorio (GitHub o similar).
- Archivo `docker-compose.yml` o instrucciones claras para construir y ejecutar cada contenedor.
- Archivo `README.md` con:
 - Descripción del proyecto.
 - Diagrama de arquitectura.
 - Instrucciones de uso.
 - Ejemplos de solicitudes y respuestas esperadas.

Referencias

- <https://roble.openlab.uninorte.edu.co/>
- <https://roble.openlab.uninorte.edu.co/docs/autenticacion>
- <https://roble.openlab.uninorte.edu.co/docs/database/records>
- https://github.com/openlabun/EJEMPLOS_ROBLE