

Parcial 3: Creación, Pruebas y Despliegue de Microservicios en Kubernetes con Spring Boot y MariaDB

Objetivo

El estudiante desarrollará cuatro microservicios independientes en Spring Boot, cada uno implementando un endpoint específico del CRUD (Crear, Leer, Actualizar, Eliminar) sobre una entidad de su elección, garantizando su correcto funcionamiento mediante pruebas unitarias e integración. Además, deberá desplegar estos microservicios utilizando Kubernetes, incluyendo todos los componentes necesarios como Pods, Deployments, Services, ReplicaSets e Ingress, utilizando MariaDB como base de datos.

Requerimientos Técnicos

1. Desarrollo del Microservicio

El microservicio debe cumplir con los siguientes requisitos:

- Endpoint específico del CRUD (Crear, Leer, Actualizar, Eliminar).
 - Arquitectura modular:
 - Controlador (Endpoint REST).
 - Servicio (Capa de negocio).
 - Repositorio o DAO para interacción con MariaDB.
 - Configuración del archivo `application.properties` con conexión a MariaDB, asegurando la resolución DNS en Kubernetes.
 - Uso de Spring Data JPA para base de datos.
-

2. Configuración de la Base de Datos

- MariaDB ejecutada en Kubernetes utilizando Persistent Volumes (PV) y Persistent Volume Claims (PVC).

- Se debe realizar una copia de seguridad de la base de datos, almacenándola con un nombre que incluya la fecha del respaldo.
-

3. Pruebas Unitarias y de Integración

- Implementar pruebas unitarias para cada endpoint utilizando JUnit y Mockito.
 - Implementar pruebas de integración para verificar la correcta interacción con la base de datos y otras capas del sistema.
 - Se debe generar un reporte de cobertura de código usando JaCoCo.
-

4. Documentación y Pruebas con Postman

- Se debe crear una colección en Postman con todos los endpoints del microservicio.
 - Incluir pruebas básicas para verificar el correcto funcionamiento del CRUD.
 - Exportar la colección y entregarla como parte del trabajo final.
-

5. Contenerización del Microservicio

- Crear un Dockerfile utilizando el concepto de Multi-Stage Build:
 - Primera etapa: Compilar la aplicación con Maven/Gradle.
 - Segunda etapa: Ejecutar la aplicación con una imagen optimizada.
- Implementar buenas prácticas en la construcción del Dockerfile:
 - Minimizar capas.
 - Evitar archivos innecesarios.
 - Usar **.dockerignore**.
 - Seguir buenas prácticas de seguridad.
- Construir la imagen y probar el contenedor antes de la publicación.



Referencias:

- [Cómo escribir un Dockerfile](#)
 - [Buenas prácticas para Dockerfile](#)
 - [Guía para contenerizar un microservicio en Spring Boot](#)
-

6. Publicación en Docker Hub

- Subir la imagen del microservicio a Docker Hub.

- Se recomienda que el nombre del repositorio en Docker Hub siga el siguiente formato:
 - **[primera letra del nombre][apellido][segunda letra del apellido]**
 - **Ejemplo:**
 - Nombre: Juan Pérez
 - Usuario en Docker Hub: jperezp
-

7. Despliegue en Kubernetes

- Archivos YAML para:
 - Pods
 - Deployments
 - Services
 - ReplicaSets
 - Ingress
 - Persistent Volumes (PV)
 - Persistent Volume Claims (PVC)
 - Configuración correcta de redes y accesos externos con ingress.
-

8. Diagrama de Infraestructura

Diagrama detallado mostrando:

- Componentes (Pods, Deployments, Services, ReplicaSets, Ingress, PV, PVC).
- Configuración de contenedores.
- IPs asignadas.
- Redes y seguridad (Firewall o WAF).

8. Automatización del Proceso de Construcción

Script obligatorio en Bash o PowerShell para:

- Construcción automática de imágenes.
- Despliegue automático tras cambios en código.
- Verificación automática de endpoints con comandos **curl**.

Entregables

- Código fuente en GitHub/GitLab.
- Dockerfiles estructurados.
- Archivos YAML de Kubernetes (Pods, Deployments, Services, ReplicaSets, Ingress, PV, PVC).
- Colección de Postman exportada con pruebas.
- Reportes de pruebas unitarias e integración.
- Reporte de cobertura con JaCoCo.
- Comando para copias de seguridad de MariaDB.
- Capturas/logs de:
 - CRUD funcionando en Postman.
 - Persistencia con PV y PVC.
 - Imágenes publicadas en Docker Hub.
- Diagrama de infraestructura.
- Script de automatización documentado y funcional.

Criterios de Evaluación

Criterio	Puntos
Implementación modular del CRUD (4 microservicios)	15 pts
Configuración y conexión correcta a MariaDB	10 pts
Persistencia y respaldo en MariaDB usando PV y PVC	10 pts
Colección Postman creada y exportada	10 pts
Pruebas unitarias e integración con cobertura	15 pts

Dockerfiles con Multi-Stage Build y buenas prácticas	10 pts
Recursos Kubernetes configurados correctamente	10 pts
Publicación en Docker Hub	5 pts
Diagrama de infraestructura	5 pts
Organización y documentación en repositorio	5 pts
Automatización funcional incluyendo verificación con curl	5 pts
Total	100 pts