Mapa

Descripción generada automáticamente

# DOCKER Y KUBERNETES AVANZADO

# Trabajo Final

|  |  |
| --- | --- |
| Grupo 02 Roberto Cabrera Cabell  Javier Velasquez Montero  José Yugar Ladines  Raúl Arrascue Sanchez |  |

|  |
| --- |
| Presentación El presente trabajo tiene por objetivo al finalizar este proyecto, los participantes serán capaces de desarrollar, contenerizar y desplegar una solución basada en microservicios para la gestión de eventos y sus participantes, utilizando Docker y Kubernetes. Esto incluirá la creación de imágenes Docker optimizadas, la configuración de un archivo Docker Compose para el desarrollo local, y el despliegue en un clúster de Kubernetes con configuración de servicios, replicasets, y acceso mediante Ingress, aplicando las mejores prácticas de escalabilidad, gestión de redes y almacenamiento en contenedores. |
|  |

|  |
| --- |
| Tecnologías y Herramientas utilizadas:  * Lenguaje de programación: * Java 21 * Frameworks y librerías: * Spring Cloud Kubernetes * Spring Boot 3.3.x * Spring Cloud Gateway * Spring Boot Actuator * Spring Data JPA * Spring Web * Spring Cloud OpenFeign * Lombok * Oracle Database JDBC * Apache Maven * Springdoc OpenAPI * Base de datos: * Oracle EM Express 21 * Contenerización: * Docker * Docker Compose * Orquestación de contenedores: * Kubernetes * IDES * Intellij IDEA (Community Edition) * STS 4 * Plataforma de Desarrollo Colaborativo * GitHub * Herramienta de Documentación y Diseño de APIs * Swagger |

# Lista de Entregables

* **Informe del trabajo final** (Informe Grupo 05.docx).
* **Ruta GitHub del Proyecto final**

<https://github.com/rmcabrera/gestor-eventos-participantes.git>

* **Ruta GitHub de los entregables**

<https://github.com/raulkkonen1/repo-equipo05.git>

* **Código fuente de la aplicación**
* Código fuente de los microservicios
* Estructura de los proyectos
* Ruta GIT del proyecto final
* JARs generados
* **Artefactos contenerizados**
* Dockerfiles
* Docker compose
* Imágenes docker
* **Configuración Kubernetes**

Archivos YAML agrupados por:

* Namespace
* Volumenes
* Configmaps
* Secrets
* Roles
* Clusterroles
* Deployments
* Services
* Ingress
* **Base de datos**
* Scripts de inicialización
* volumenes
* **Documentación**
* Manual de despliegue
* Diagrama de arquitectura
* Documentación de APIs con Swagger
* **Resultados**
* Endpoints de Postman (json)
* Imágenes en docker hub
* Validación del despliegue en Kubernetes
* Validación de PODs con Postman