



DOCKER Y KUBERNETES INTERMEDIO

Proyecto Final del Curso

Integrantes

Quintana Cubas Lenin Alexander
Ponte Arica Anthony Rosemberg
Ramirez Vasquez Jonathan
Silva Ramos Juan Armando

Contenido

Proyecto Curso Docker y Kubernetes	2
Entregables del Proyecto Final: Gestión de Eventos y Participantes	2
1. Código Fuente	2
1.1. Repositorios de Microservicios:	2
2. Artefactos Contenerizados.....	4
2.1. Imágenes Docker:	4
2.2. Docker Compose:	8
3. Configuración para Kubernetes	10
3.1. Archivos YAML:.....	10
4. Microservicios.	16
4.1. Manifiestos completos:	17
5. Bases de Datos	18
5.1. Scripts de Inicialización:.....	18
5.2. Volúmenes:.....	25
6. Documentación	25
6.1. Manual de Despliegue:	25
6.2. Diagrama de Arquitectura:	27
7. Resultados	27
7.1. Validación del Despliegue:.....	27
7.2. Registro de Imágenes:	33

Proyecto Curso Docker y Kubernetes

Entregables del Proyecto Final: Gestión de Eventos y Participantes

Los entregables se organizan en categorías que abarcan desde el código fuente hasta la documentación y los artefactos de despliegue.

Se creó el siguiente repositorio Git para todos los entregables del presente proyecto: https://github.com/jonathan0284/Curso_Docker_K8S.git

1. Código Fuente

1.1. Repositorios de Microservicios:

- Código fuente de los dos Micro servicios: gestión de eventos y gestión de inscripciones (participantes).

ms-eventos:

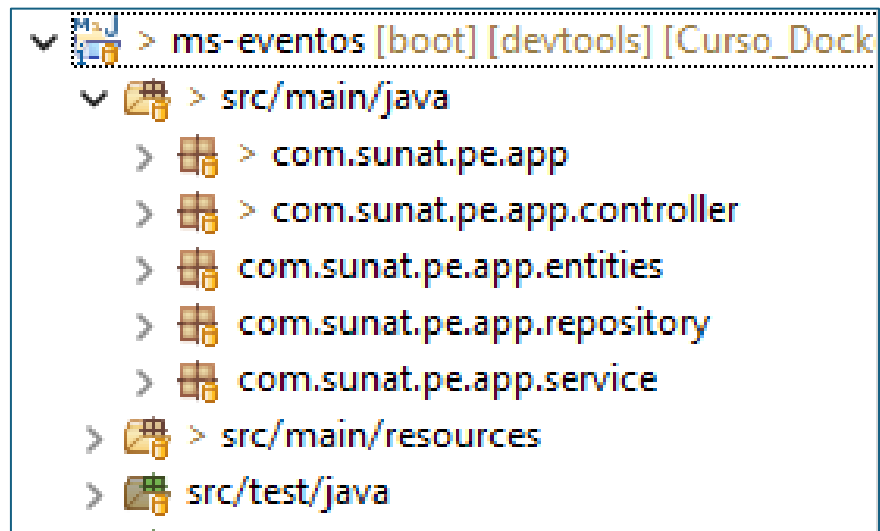
https://github.com/jonathan0284/Curso_Docker_K8S/tree/main/Microservicios/ms-eventos

ms-participantes:

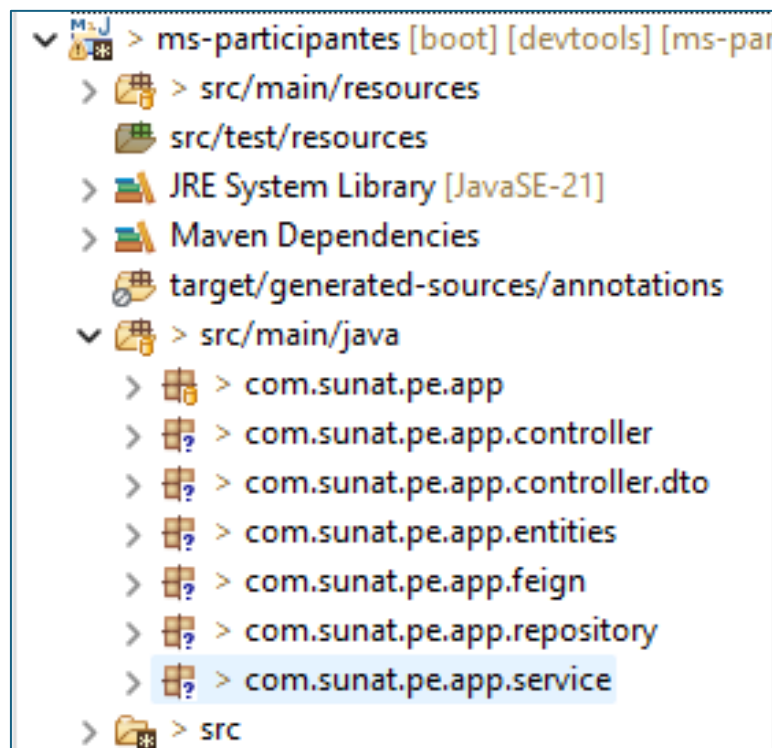
https://github.com/jonathan0284/Curso_Docker_K8S/tree/main/Microservicios/ms-participantes

- Estructura del proyecto conforme a las mejores prácticas (separación en capas: controlador, servicio, repositorio, entidades, etc.).

ms-eventos



ms-participantes



- Uso de control de versiones (Git) con un historial claro de commits:
https://github.com/jonathan0284/Curso_Docker_K8S/activity

2. Artefactos Contenerizados

2.1. Imágenes Docker:

- Imágenes Docker de ambos microservicios, construidas y publicadas en un repositorio como Docker Hub.



Comandos Docker.txt

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/contenedores%20docker/Comandos%20Docker.txt

ms-eventos

<https://hub.docker.com/repository/docker/jonathan0284/ms-eventos>

ms-participantes

<https://hub.docker.com/repository/docker/jonathan0284/ms-participantes/general>

```

Comandos Docker.txt U X
2-Entregables > Contenedores Docker > Comandos Docker.txt



1 Paso 1. Crear Dockerfile para ms-participantes
2 =====
3 FROM openjdk:21-jdk-slim
4 WORKDIR /app
5 COPY target/ms-participantes-0.0.1-SNAPSHOT.jar app.jar
6 EXPOSE 7075
7 ENTRYPOINT ["java", "-jar", "app.jar"]
8
9 Paso 2. Crear Dockerfile para ms-eventos
10 =====
11 FROM openjdk:21-jdk-slim
12 WORKDIR /app
13 COPY target/ms-eventos-0.0.1-SNAPSHOT.jar app.jar
14 EXPOSE 9090
15 ENTRYPOINT ["java", "-jar", "app.jar"]
16
17 Paso 3. Comando para compilar los Microservicios
18 =====
19 .\mvnw clean install
20
21 Paso 4. Crear las imágenes Docker
22 =====
23 docker build -t ms-eventos:1.5 .
24 docker build -t ms-participantes:1.5 .
25
26 Paso 5. Crear y levantar los contenedores
27 =====
28 docker run -d --name ms-participantes --network dki-network -p 7075:7075 ms-participantes:1.5
29 docker run -d --name ms-eventos --network dki-network -p 9090:9090 ms-eventos:1.5
30
31 Paso 6. Subir Imágenes a Docker HUB
32 =====
33 docker tag ms-participantes:1.5 jonathan0284/trabajo-final:ms-participantes-1.5
34 docker push jonathan0284/trabajo-final:ms-participantes-1.5
35
36 docker tag ms-eventos:1.5 jonathan0284/trabajo-final:ms-eventos-1.5
37 docker push jonathan0284/trabajo-final:ms-eventos-1.5
38



```

[Repositories](#) / [ms-eventos](#) / [General](#)

jonathan0284/ms-eventos

Last pushed 5 days ago • Repository size: 325.9 MB



[Add a description](#)  

[Add a category](#)  

General Tags Image Management BETA Collaborators

Tags

This repository contains 1 tag(s).



Tag	OS	Type
 1.5		Image



[See all](#)

[Repositories](#) / [ms-participantes](#) / [General](#)

jonathan0284/ms-participantes

Last pushed 5 days ago • Repository size: 311.4 MB



[Add a description](#)  

[Add a category](#)  

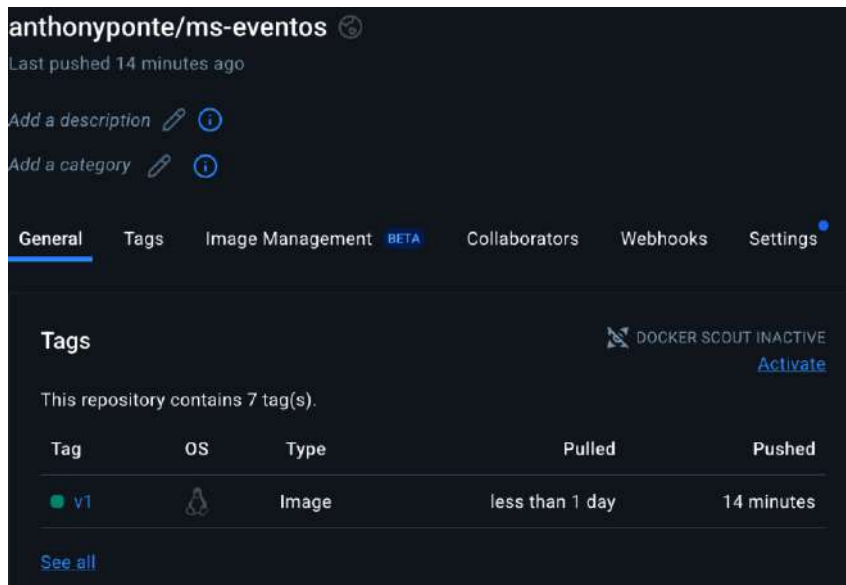
General Tags Image Management BETA Collaborators

Tags

This repository contains 1 tag(s).

Tag	OS	Type
 1.5		Image

[See all](#)



- Imágenes optimizadas según las mejores prácticas (Dockerfile eficiente).

ms-eventos



Dockerfile

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/contenedores%20docker/ms-eventos/Dockerfile

```

Dockerfile U X
2-Entregables > Contenedores Docker > ms-eventos > Dockerfile
1 FROM openjdk:21-jdk-slim
2 WORKDIR /app
3 COPY target/ms-eventos-0.0.1-SNAPSHOT.jar app.jar
4 EXPOSE 9090
5 ENTRYPOINT ["java", "-jar", "app.jar"]
    
```


ms-participantes



Dockerfile

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/contenedores%20docker/ms-participantes/Dockerfile

```
Dockerfile U x
2-Entregables > Contenedores Docker > ms-participantes > Dockerfile
1 FROM openjdk:21-jdk-slim
2 WORKDIR /app
3 COPY target/ms-participantes-0.0.1-SNAPSHOT.jar app.jar
4 EXPOSE 7075
5 ENTRYPOINT ["java", "-jar", "app.jar"]
```

2.2. Docker Compose:

- Archivo docker-compose.yml para levantar ambos microservicios junto con sus bases de datos en un entorno de desarrollo local.



docker-compose.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/docker-compose/docker-compose.yml

```

3-Microservicios > docker > docker-compose.yml
1  services:
2    oracle-db:
3      image: container-registry.oracle.com/database/express:21.3.8-xe
4      hostname: oracle-db
5      container_name: oracle-db-container
6      environment:
7        ORACLE_PWD: "dkpassword"
8        ORACLE_SID: "XEPOB1"
9        ORACLE_PDB: "XEPOB1"
10       SYSDBA_USER: "sys"
11       SYSDBA_PASSWORD: "dkpasswords"
12       ORUSER: "dkuser"
13       ORUSER_PASSWORD: "dkpassword"
14       ORACLE_CHARACTERSET: AL32UTF8
15       NLS_LANG: AMERICAN_AMERICA.AL32UTF8
16     ports:
17       - "1521:1521"
18       - "5500:5500"
19     volumes:
20       - ms-volume:/opt/oracle/oradata
21       - ../oracle-db/scripts:/opt/oracle/scripts/startup
22     networks:
23       - ms-network
24     healthcheck:
25       test: ["CMD", "sh", "-c", "echo 'SELECT 1 FROM DUAL;' | sqlplus system/dkpassword@localhost:1521/XE"]
26       interval: 30s
27       timeout: 10s
28       retries: 5
29       start_period: 60s
30
31   ms-eventos:
32     image: jonathan0284/ms-eventos:1.5
33     container_name: ms-eventos
34     build:
35       context: ../ms-eventos
36     ports:
37       - "9090:9090"
38     networks:
39       - ms-network
40     depends_on:
41       oracle-db:
42         condition: service_healthy
43     healthcheck:
44       test: ["CMD", "curl", "-f", "http://127.0.0.1:9090/eventos"]
45       interval: 30s
46       timeout: 15s
47       retries: 5
48       start_period: 30s
49
50   ms-participantes:
51     image: jonathan0284/ms-participantes:1.5
52     container_name: ms-participantes
53     build:
54       context: ../ms-participantes
55     ports:
56       - "7075:7075"
57     environment:
58       - MS_EVENTOS_URL=http://ms-eventos:9090.
59     networks:
60       - ms-network
61     depends_on:
62       oracle-db:
63         condition: service_healthy
64       ms-eventos:
65         condition: service_healthy
66     healthcheck:
67       test: ["CMD", "curl", "-f", "http://127.0.0.1:7075/participante"]
68       interval: 30s
69       timeout: 15s
70       retries: 5
71       start_period: 30s
72
73   volumes:
74     ms-volume:
75
76   networks:
77     ms-network:
78       driver: bridge

```

3. Configuración para Kubernetes

3.1. Archivos YAML:

- **Deployments:** Configuración de pods y réplicas para ambos microservicios.

ms-eventos-configmap.yml



ms-eventos-configmap.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/k8s/configmap/ms-eventos-configmap.yml

```
! ms-eventos-configmap.yml U X
2-Entregables > k8s > configmap > ! ms-eventos-configmap.yml
1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4    name: ms-eventos-configmap
5    namespace: default
6  data:
7    application.yml: |
8      spring:
9        application:
10         name: ms-eventos
11        datasource:
12         url: jdbc:oracle:thin:@oracle-db-service:1521/XEPDB1
13         username: ${DB_USERNAME}
14         password: ${DB_PASSWORD}
15         driver-class-name: oracle.jdbc.OracleDriver
16      server:
17        port: 9090
18      management:
19        endpoints:
20          web:
21            exposure:
22              include: "*"
23        endpoint:
24          health:
25            show-details: always
26            probes:
27              enabled: true
28      health:
29        livenessState:
30          enabled: true
31        readinessState:
32          enabled: true
```

ms-participantes-configmap.yaml



ms-participantes-configmap.yaml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/k8s/configmap/ms-participantes-configmap.yaml

```
! ms-participantes-configmap.yaml U X
2-Entregables > k8s > configmap > ! ms-participantes-configmap.yaml
1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4    name: ms-participantes-config
5  data:
6    application.yml: |
7      server:
8        port: 7075
9
10     spring:
11       application:
12         name: ms-participantes
13
14     eventos:
15       service:
16         url: http://ms-eventos:9090
17
18     management:
19       endpoint:
20         health:
21           probes:
22             enabled: true
23         health:
24           livenessstate:
25             enabled: true
26           readinessstate:
27             enabled: true
28       endpoints:
29         web:
30           exposure:
31             include: health
32
```

ms-eventos-deployment.yaml



ms-eventos-deployment.yaml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/k8s/deployment/ms-eventos-deployment.yaml

```

1 ms-eventos-deployment.yml X
2 Entregables > k8s > deployment > ms-eventos-deployment.yml
3
4 apiVersion: apps/v1
5 kind: Deployment
6 metadata:
7   name: ms-eventos-deployment
8 spec:
9   replicas: 1
10  selector:
11    matchLabels:
12      app: ms-eventos
13  template:
14    metadata:
15      labels:
16        app: ms-eventos
17    spec:
18      containers:
19        - name: ms-eventos-container
20          image: anthonyponce/ms-eventos:v1
21          args:
22            - "--spring.config.additional-location-file:/config/"
23          ports:
24            - containerPort: 9090
25          livenessProbe:
26            httpGet:
27              path: /actuator/health/liveness
28              port: 9090
29              initialDelaySeconds: 400
30              periodSeconds: 30
31              timeoutSeconds: 10
32          readinessProbe:
33            httpGet:
34              path: /actuator/health/readiness
35              port: 9090
36              initialDelaySeconds: 105
37              periodSeconds: 10
38              timeoutSeconds: 10
39          env:
40            - name: DB_USERNAME
41              valueFrom:
42                secretKeyRef:
43                  name: grupo2-secret
44                  key: DB_USERNAME
45            - name: DB_PASSWORD
46              valueFrom:
47                secretKeyRef:
48                  name: grupo2-secret
49                  key: DB_PASSWORD
50            - name: POD_NAME
51              valueFrom:
52                fieldRef:
53                  fieldPath: metadata.name
54            - name: POD_ID
55              valueFrom:
56                fieldRef:
57                  fieldPath: status.podIP
58          resources:
59            requests:
60              cpu: "100m"
61              memory: "400Mi"
62            limits:
63              cpu: "200m"
64              memory: "600Mi"
65          volumeMounts:
66            - name: ms-eventos-volume
67              mountPath: /config/
68              subPath: application.yml
69      volumes:
70        - name: ms-eventos-volume
71          configMap:
72            name: ms-eventos-configmap
73            items:
74              - key: application.yml
75                path: application.yml

```

ms-participantes-deployment.yml



ms-participantes-deployment.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/k8s/deployment/ms-participantes-deployment.yml

! ms-participantes-deployment.yml U ●

2-Entregables > k8s > deployment > ! ms-participantes-deployment.yml

```

1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: ms-participantes
5    labels:
6      app: ms-participantes
7  spec:
8    replicas: 1
9    selector:
10     matchLabels:
11       app: ms-participantes
12    template:
13      metadata:
14        labels:
15          app: ms-participantes
16      spec:
17        containers:
18          - name: ms-participantes
19            image: anthonyponce/ms-participantes:v1
20            args:
21              - "--spring.config.additional-location=file:/config/"
22            ports:
23              - containerPort: 7075
24
25        livenessProbe:
26          httpGet:
27            path: /actuator/health/liveness
28            port: 7075
29            initialDelaySeconds: 100
30            periodSeconds: 30
31            timeoutSeconds: 10
32
33        readinessProbe:
34          httpGet:
35            path: /actuator/health/readiness
36            port: 7075
37            initialDelaySeconds: 105
38            periodSeconds: 15
39            timeoutSeconds: 10
40
41        env:
42          - name: POD_NAME
43            valueFrom:
44              fieldRef:
45                fieldPath: metadata.name
46          - name: POD_ID
47            valueFrom:
48              fieldRef:
49                fieldPath: status.podIP
50
51        resources:
52          requests:
53            cpu: "100m"
54            memory: "300Mi"
55          limits:
56            cpu: "200m"
57            memory: "500Mi"
58
59        volumeMounts:
60          - name: config-volume
61            mountPath: /config/
62            subPath: application.yml
63
64        volumes:
65          - name: config-volume
66            configMap:
67              name: ms-participantes-config
68              items:
69                - key: application.yml
70                  path: application.yml
71

```

- **Services:** Definición de servicios ClusterIP o NodePort para los

ms-eventos-service.yml



ms-eventos-service.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/k8s/service/ms-eventos-service.yml

```
! ms-eventos-service.yml U X
2-Entregables > k8s > service > ! ms-eventos-service.yml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: ms-eventos-service
5  spec:
6    selector:
7      app: ms-eventos
8    ports:
9      - protocol: TCP
10        port: 9090
11        targetPort: 9090
12        nodePort: 30090
13    type: NodePort
```

ms-participantes-service.yml



ms-participantes-service.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/k8s/service/ms-participantes-service.yml


```
! ms-participantes-service.yml U X
2-Entregables > k8s > service > ! ms-participantes-service.yml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: ms-participantes-service
5  spec:
6    selector:
7      app: ms-participantes
8    ports:
9      - protocol: TCP
10        port: 7075
11        targetPort: 7075
12        nodePort: 30075
13    type: NodePort
```

4. Microservicios.

- **Ingress:** Configuración de rutas basadas en contexto para exponer las APIs externamente (Opcional, este tema se cubrió en en nivel Básico)
- **ConfigMaps y Secrets:** Manejo de variables de entorno y datos sensibles.

secret.yml



secret.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/k8s/secret.yml

```
! secret.yml U X
2-Entregables > k8s > ! secret.yml
1  apiVersion: v1
2  kind: Secret
3  metadata:
4    name: grupo2-secret
5  type: Opaque
6  data:
7    DB_USERNAME: ZGt1c2Vy
8    DB_PASSWORD: ZGtwYXNzd29yZA==
```

- **Probes:** Liveness y readiness probes configuradas.

4.1. Manifiestos completos:

- Conjunto de manifiestos YAML organizados y documentados en carpetas para cada microservicio.

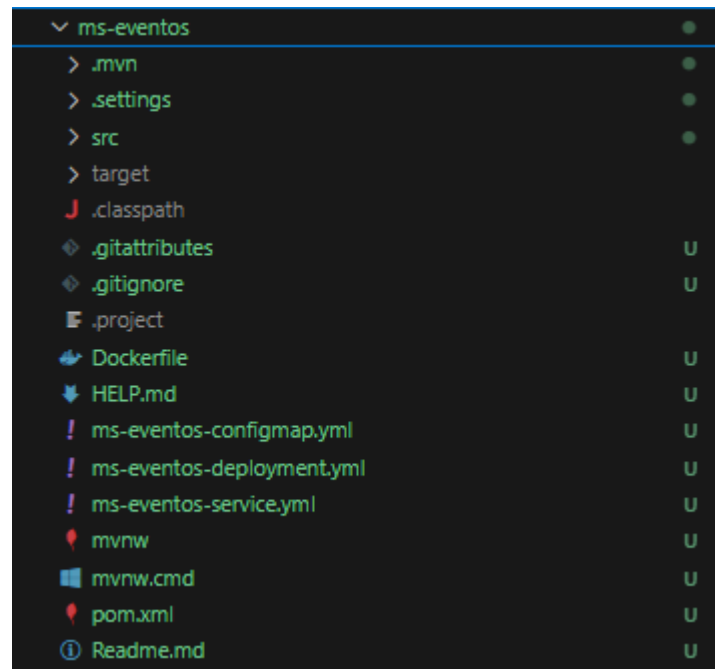
ms-participantes

```

v ms-participantes
├── .mvn
├── .settings
├── src
├── target
├── .classpath
├── .gitattributes
├── .gitignore
├── .project
├── Dockerfile
├── HELP.md
├── ms-participantes-configmap.yml
├── ms-participantes-deployment.yml
├── ms-participantes-service.yml
├── mvnw
├── mvnw.cmd
├── pom.xml
├── README.md
├── .gitignore
└── README.md

```

ms-eventos



5. Bases de Datos

5.1. Scripts de Inicialización:

- Scripts SQL para inicializar las bases de datos con tablas necesarias y datos de prueba.

oracle-db-configmap.yml



oracle-db-configmap.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/oracle-db/oracle-db-configmap.yml

```

! oracle-db-configmap.yml U X
2-Entregables > oracle-db > ! oracle-db-configmap.yml
1  apiVersion: v1
2  kind: ConfigMap
3  metadata:
4    name: oracle-db-configmap
5  data:
6    init.sql: |-
7      SELECT 'Oracle initialization complete' FROM dual;
8    dki_01.sql: |-
9      SELECT SYS_CONTEXT('USERENV', 'CON_NAME') AS CURRENT_CONTAINER FROM DUAL;
10     ALTER SESSION SET CONTAINER = XEPDB1;
11
12     CREATE USER dkuser IDENTIFIED BY dkpassword;
13     GRANT CONNECT, RESOURCE TO dkuser;
14     ALTER USER dkuser QUOTA UNLIMITED ON users;
15
16     CREATE TABLE dkuser.eventos (
17       id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
18       codigo VARCHAR2(125) NOT NULL,
19       nombre VARCHAR2(255) NOT NULL,
20       descripcion VARCHAR2(255) NOT NULL,
21       fecha DATE NOT NULL,
22       ubicacion VARCHAR2(255) NOT NULL,
23       capacidad_max INT NOT NULL
24     );
25
26     CREATE TABLE dkuser.participantes (
27       id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
28       id_evento NUMBER NOT NULL,
29       dni VARCHAR2(8) NOT NULL,
30       nombres_apellidos VARCHAR2(255) NOT NULL,
31       fecha_registro DATETIME NOT NULL
32     );
33
34     ALTER TABLE dkuser.participantes
35     ADD CONSTRAINT fk_participantes_eventos
36     FOREIGN KEY (id_evento)
37     REFERENCES dkuser.eventos(id);
38    dki_02.sql: |-
39      ALTER SESSION SET CONTAINER=CDB$ROOT;
40      ALTER DATABASE OPEN;
41      CREATE PLUGGABLE DATABASE XEPDB2 FROM XEPDB1;
42      ALTER PLUGGABLE DATABASE XEPDB2 OPEN;

```

oracle-db-deployment.yml



oracle-db-deployment.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/oracle-db/oracle-db-deployment.yml

```

! oracle-db-deployment.yml U X
2-Entregables > oracle-db > ! oracle-db-deployment.yml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: oracle-db-deployment
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: oracle-db
10   template:
11     metadata:
12       labels:
13         app: oracle-db
14     spec:
15       containers:
16       - name: oracle-db-container
17         image: container-registry.oracle.com/database/express:21.3.0-xe
18         ports:
19         - containerPort: 1521
20         env:
21         - name: ORACLE_SID
22           value: "XE"
23         - name: ORACLE_PDB
24           value: "XEPDB1"
25         - name: ORACLE_PWD
26           value: "dkpassword"
27         - name: ORACLE_CHARACTERSET
28           value: "AL32UTF8"
29         - name: NLS_LANG
30           value: "AMERICAN_AMERICA.AL32UTF8"
31       volumeMounts:
32       - name: oracle-db-data
33         mountPath: /opt/oracle/oradata
34       - name: oracle-db-scripts
35         mountPath: /opt/oracle/scripts/startup
36       volumes:
37       - name: oracle-db-data
38         persistentVolumeClaim:
39           claimName: oracle-db-pvc
40       - name: oracle-db-scripts
41         configMap:
42           name: oracle-db-configmap

```

oracle-db-pv.yml



oracle-db-pv.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/oracle-db/oracle-db-pv.yml

```
! oracle-db-pv.yml U X
2-Entregables > oracle-db > ! oracle-db-pv.yml
1  apiVersion: v1
2  kind: PersistentVolume
3  metadata:
4    name: oracle-db-pv
5  spec:
6    capacity:
7      storage: 50Gi
8    accessModes:
9      - ReadWriteOnce
10   hostPath:
11     path: /data/oracle
```

oracle-db-pvc.yml



oracle-db-pvc.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/oracle-db/oracle-db-pvc.yml

```
! oracle-db-pvc.yml U X
2-Entregables > oracle-db > ! oracle-db-pvc.yml
1  apiVersion: v1
2  kind: PersistentVolumeClaim
3  metadata:
4    name: oracle-db-pvc
5  spec:
6    accessModes:
7      - ReadWriteOnce
8    resources:
9      requests:
10     storage: 50Gi
```

oracle-db-service.yml



oracle-db-service.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/oracle-db/oracle-db-service.yml

```
! oracle-db-service.yml u x
2-Entregables > oracle-db > ! oracle-db-service.yml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: oracle-db-service
5  spec:
6    type: NodePort
7    selector:
8      app: oracle-db
9    ports:
10     - name: oracle
11       protocol: TCP
12       port: 1521
13       targetPort: 1521
14       nodePort: 30011
```

dki_01.sql



dki_01.sql

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/oracle-db/scripts/dki_01.sql


```

dki_01.sql U X
2-Entregables > oracle-db > scripts > dki_01.sql
1  SELECT SYS_CONTEXT('USERENV', 'CON_NAME') AS CURRENT_CONTAINER FROM DUAL;
2  ALTER SESSION SET CONTAINER = XEPDB1;
3
4  CREATE USER dkuser IDENTIFIED BY dkpassword;
5  GRANT CONNECT, RESOURCE TO dkuser;
6  ALTER USER dkuser QUOTA UNLIMITED ON users;
7
8  CREATE TABLE dkuser.eventos (
9      id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
10     codigo VARCHAR2(125) NOT NULL,
11     nombre VARCHAR2(255) NOT NULL,
12     descripcion VARCHAR2(255) NOT NULL,
13     fecha DATE NOT NULL,
14     ubicacion VARCHAR2(255) NOT NULL,
15     capacidad_max INT NOT NULL
16 );
17
18 CREATE TABLE dkuser.participantes (
19     id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
20     id_evento NUMBER NOT NULL,
21     dni VARCHAR2(8) NOT NULL,
22     nombres_apellidos VARCHAR2(255) NOT NULL,
23     fecha_registro DATETIME NOT NULL
24 );
25
26 ALTER TABLE dkuser.participantes
27 ADD CONSTRAINT fk_participantes_eventos
28 FOREIGN KEY (id_evento)
29 REFERENCES dkuser.eventos(id);

```

dki_02.sql



dki_02.sql

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/oracle-db/scripts/dki_02.sql

```

dki_02.sql U X
2-Entregables > oracle-db > scripts > dki_02.sql
1 ALTER SESSION SET CONTAINER=CDB$ROOT;
2 ALTER DATABASE OPEN;
3 CREATE PLUGGABLE DATABASE XEPDB2 FROM XEPDB1;
4 ALTER PLUGGABLE DATABASE XEPDB2 OPEN;

```

5.2. Volúmenes:

- Configuración de volúmenes en Kubernetes para garantizar la persistencia de los datos.

6. Documentación

6.1. Manual de Despliegue:

- Pasos detallados para:
 - Contenerizar y construir las imágenes:
 - Verificar la existencia de la imagen dki-volume:*
 - docker volume ls*
 - Creación de la imagen dki-volume:*
 - docker volume create dki-volume*
 - Desplegar en Kubernetes utilizando los manifiestos YAML.
 - Configurar Docker Compose para desarrollo local.



docker-compose-local.yml

https://github.com/jonathan0284/Curso_Docker_K8S/blob/main/2-Entregables/docker-compose/docker-compose-local.yml

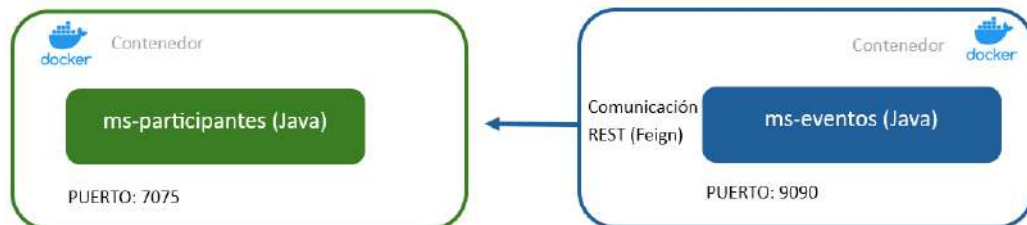
```

2-Entregables > docker-compose > docker-compose-local.yml
1  ms-participantes:
2    image: ms-participantes:1.2
3    container_name: ms-participantes
4    ports:
5      - "7075:7075"
6  environment:
7    USER_DEMO: Netec
8  networks:
9    - dki-network
10 depends_on:
11   dki-oradb:
12     condition: service_healthy
13 healthcheck:
14   test: ["CMD", "curl", "-f", "http://127.0.0.1:9081/participantes"]
15   interval: 30s
16   timeout: 15s
17   retries: 5
18
19 ms-eventos:
20   image: ms-eventos:1.2
21   container_name: ms-eventos
22   ports:
23     - "9090:9090"
24   environment:
25     MS_PARTICIPANTES_URL: http://ms-productos:7075/participantes
26   networks:
27     - dki-network
28   depends_on:
29     ms-participantes:
30       condition: service_healthy
31   healthcheck:
32     test: ["CMD", "curl", "-f", "http://127.0.0.1:9090/eventos"]
33     interval: 30s
34     timeout: 15s
35     retries: 5
36     start_period: 30s
37
38 volumes:
39   dki-volume:
40
41 networks:
42   dki-network:
43     external: true
44 services:
45   dki-oradb:
46     container_name: dki-oradb
47     image: container-registry.oracle.com/database/express:21.3.0-xe
48     environment:
49       ORACLE_PWD: Netec_123
50       ORACLE_SID: XE
51       ORACLE_PDB: XEPDB1
52       ORACLE_CHARACTERSET: AL32UTF8
53     ports:
54       - "1521:1521"
55       - "5500:5500"
56     volumes:
57       - dki-volume:/opt/oracle/oradata
58       - ./scripts:/opt/oracle/scripts/startup
59     networks:
60       - dki-network
61     healthcheck:
62       test: ["CMD", "sh", "-c", "echo 'SELECT 1 FROM DUAL;' | sqlplus system/Netec_123@localhost:1521/XE"]
63       interval: 30s
64       timeout: 10s
65       retries: 5
66       start_period: 60s
67

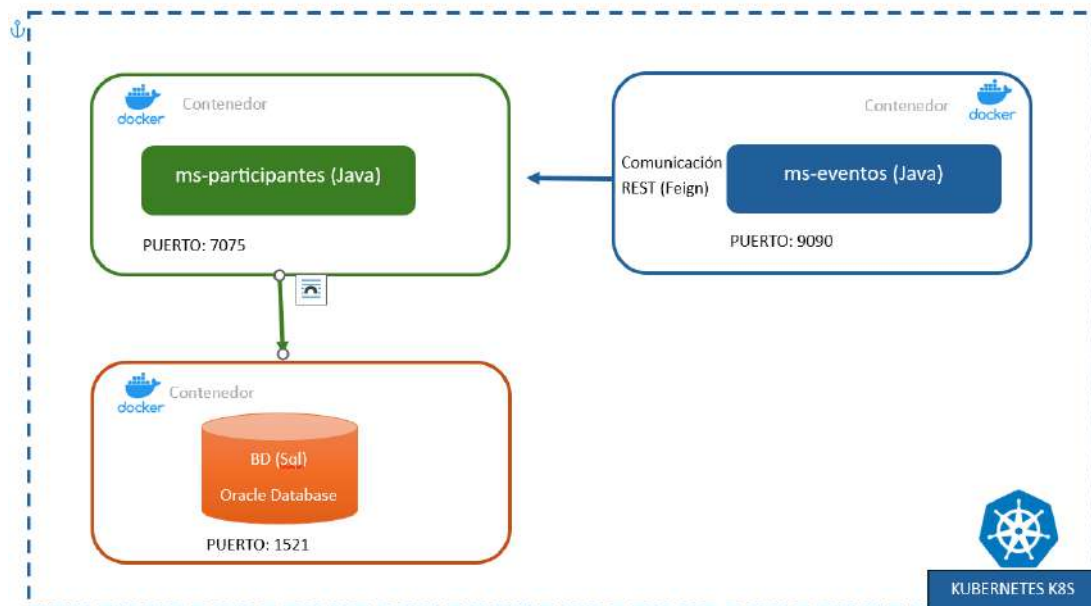
```

6.2. Diagrama de Arquitectura:

- Representación gráfica de la solución, incluyendo:
 - Estructura de microservicios:



- Comunicación entre ellos.
- Despliegue en Kubernetes:



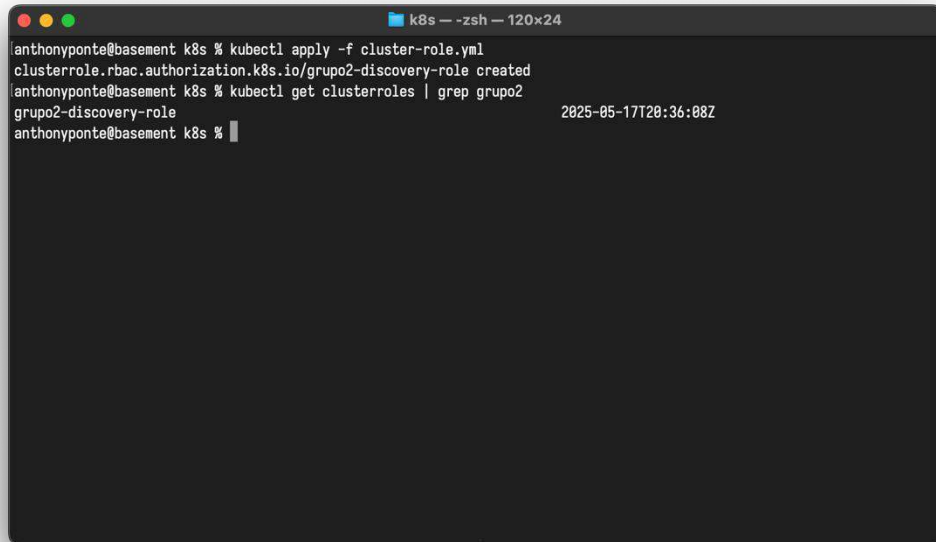
7. Resultados

7.1. Validación del Despliegue:

- Evidencia del despliegue exitoso en Kubernetes (por ejemplo, capturas de pantalla o logs que muestren los pods corriendo).

```
kubectl apply -f cluster-role.yml
```

```
kubectl get clusterroles | grep grupo2
```

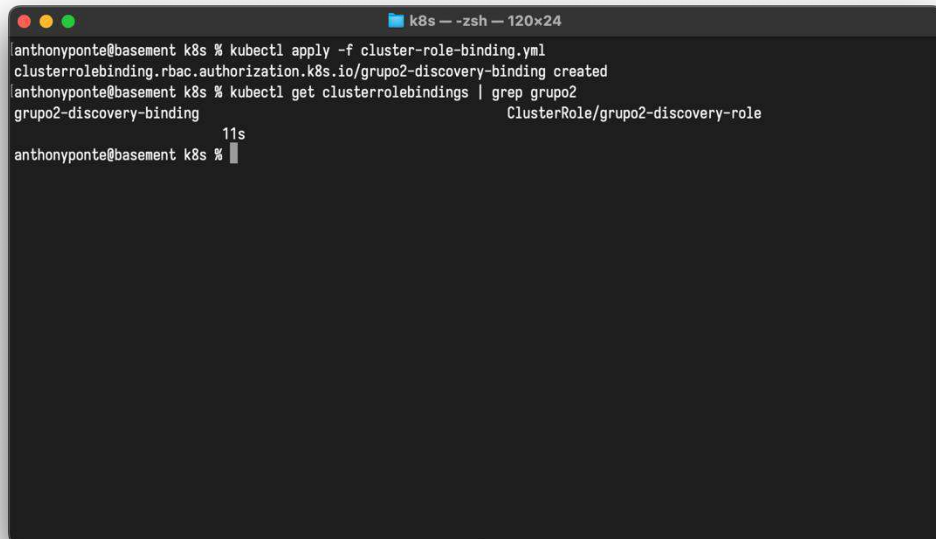


```

k8s — zsh — 120x24
anthonyponte@basement k8s % kubectl apply -f cluster-role.yml
clusterrole.rbac.authorization.k8s.io/grupo2-discovery-role created
anthonyponte@basement k8s % kubectl get clusterroles | grep grupo2
grupo2-discovery-role                                2025-05-17T20:36:08Z
anthonyponte@basement k8s %
  
```

```
kubectl apply -f cluster-role-binding.yml
```

```
kubectl get clusterrolebindings | grep grupo2
```



```

k8s — zsh — 120x24
anthonyponte@basement k8s % kubectl apply -f cluster-role-binding.yml
clusterrolebinding.rbac.authorization.k8s.io/grupo2-discovery-binding created
anthonyponte@basement k8s % kubectl get clusterrolebindings | grep grupo2
grupo2-discovery-binding                            ClusterRole/grupo2-discovery-role
11s
anthonyponte@basement k8s %
  
```

```
kubectl apply -f secret.yml
```

```
kubectl get secrets
```

```
k8s -- zsh -- 120x24
anthonyponte@basement k8s % kubectl apply -f secret.yml
secret/grupo2-secret created
anthonyponte@basement k8s % kubectl get secrets
NAME          TYPE      DATA   AGE
grupo2-secret  Opaque    2       7s
anthonyponte@basement k8s %
```

```
kubectl apply -f ms-eventos-configmap.yml
kubectl apply -f ms-eventos-deployment.yml
kubectl apply -f ms-eventos-service.yml
```

```
ms-eventos -- zsh -- 120x24
anthonyponte@basement ms-eventos % kubectl apply -f ms-eventos-configmap.yml
configmap/ms-eventos-configmap created
anthonyponte@basement ms-eventos % kubectl apply -f ms-eventos-deployment.yml
deployment.apps/ms-eventos-deployment created
anthonyponte@basement ms-eventos % kubectl apply -f ms-eventos-service.yml
service/ms-eventos-service created
anthonyponte@basement ms-eventos %
```

```
kubectl get configmaps | grep ms
kubectl get deployments | grep ms
kubectl get service | grep ms
```

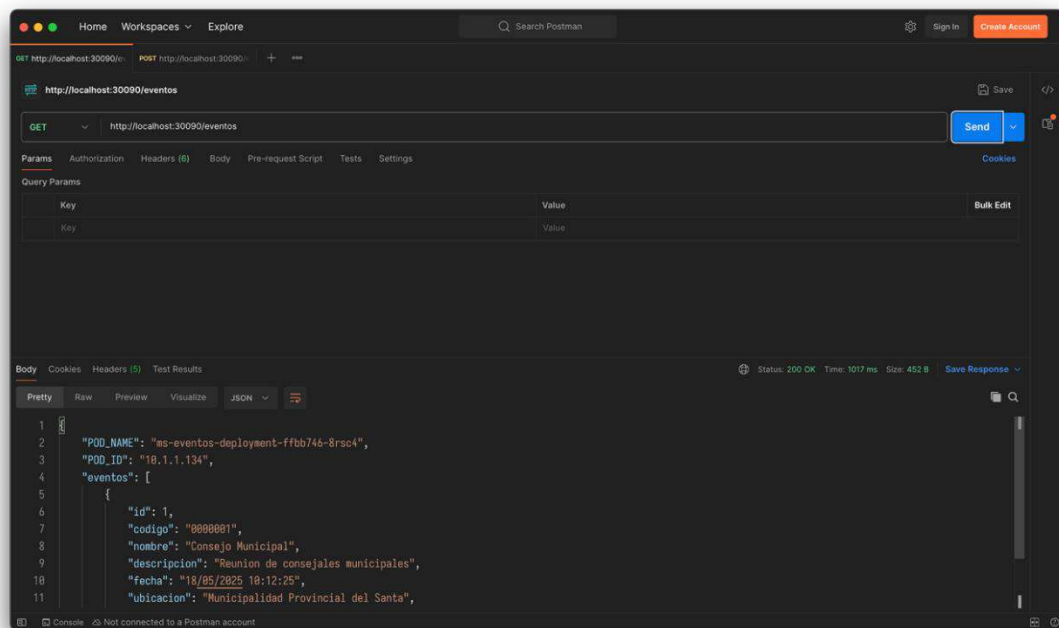
```
ms-eventos - zsh - 120x24
anthonyponte@basement ms-eventos % kubectl get configmaps | grep ms
ms-eventos-configmap 1 40s
anthonyponte@basement ms-eventos % kubectl get deployments | grep ms
ms-eventos-deployment 0/1 1 0 40s
anthonyponte@basement ms-eventos % kubectl get service | grep ms
ms-eventos-service NodePort 10.100.151.125 <none> 9090:30090/TCP 44s
anthonyponte@basement ms-eventos %
```

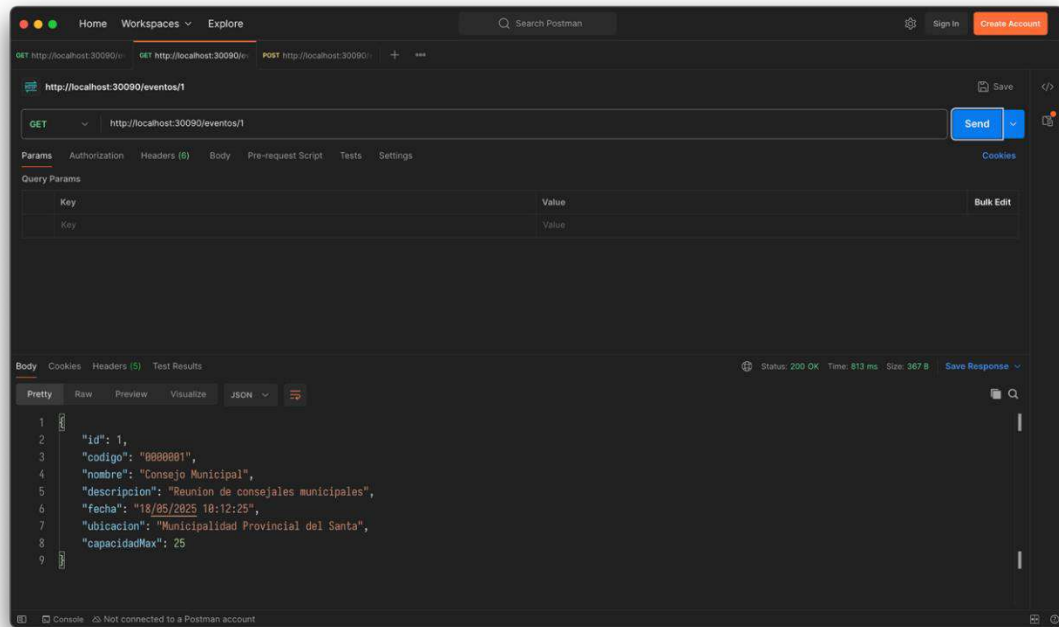
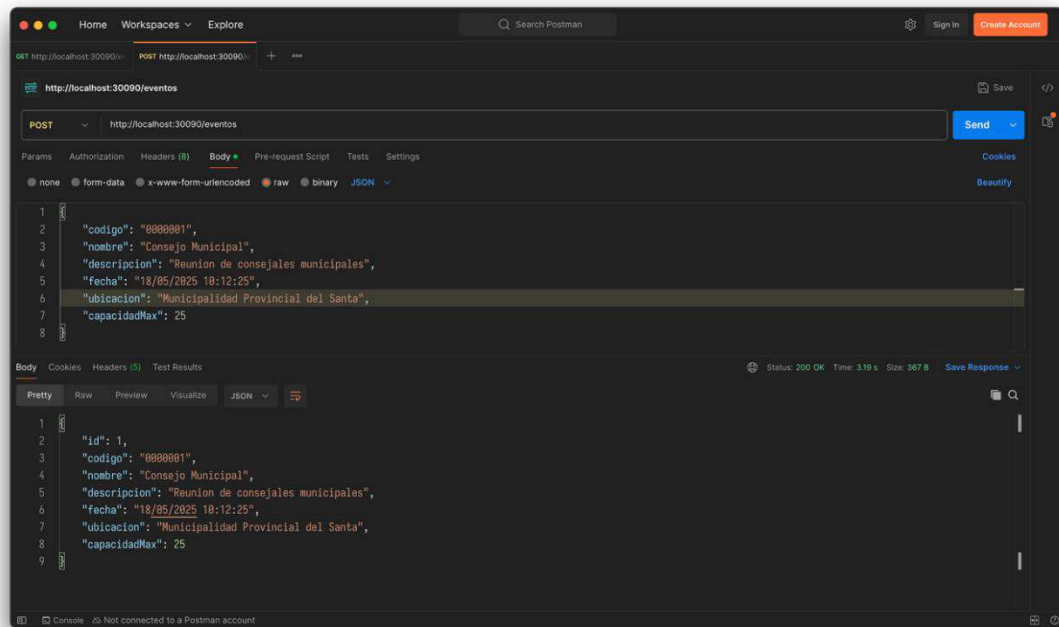
`kubectl get pods | grep ms`

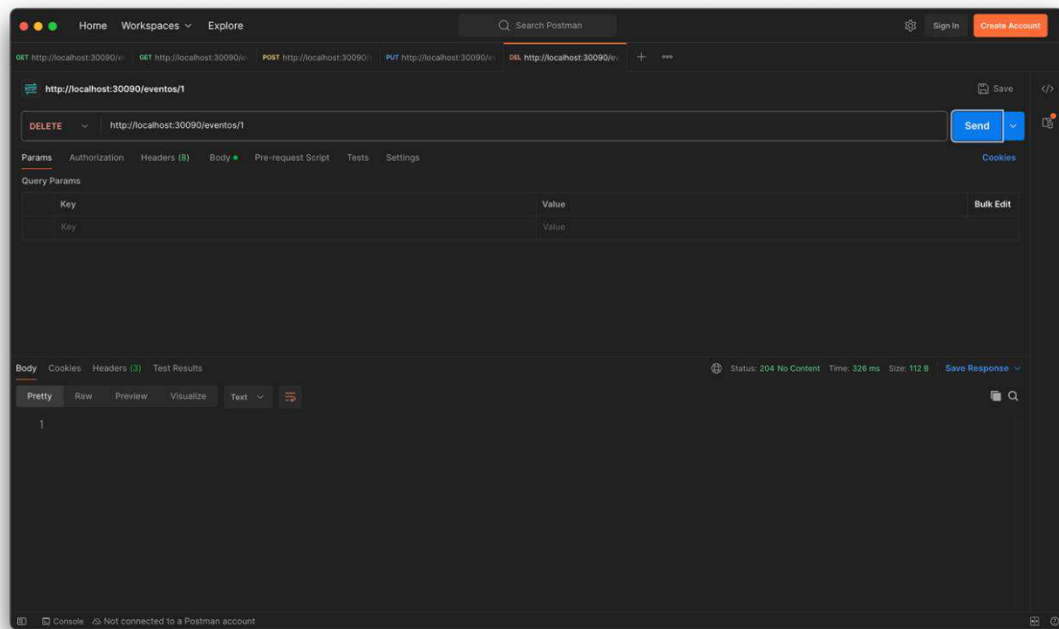
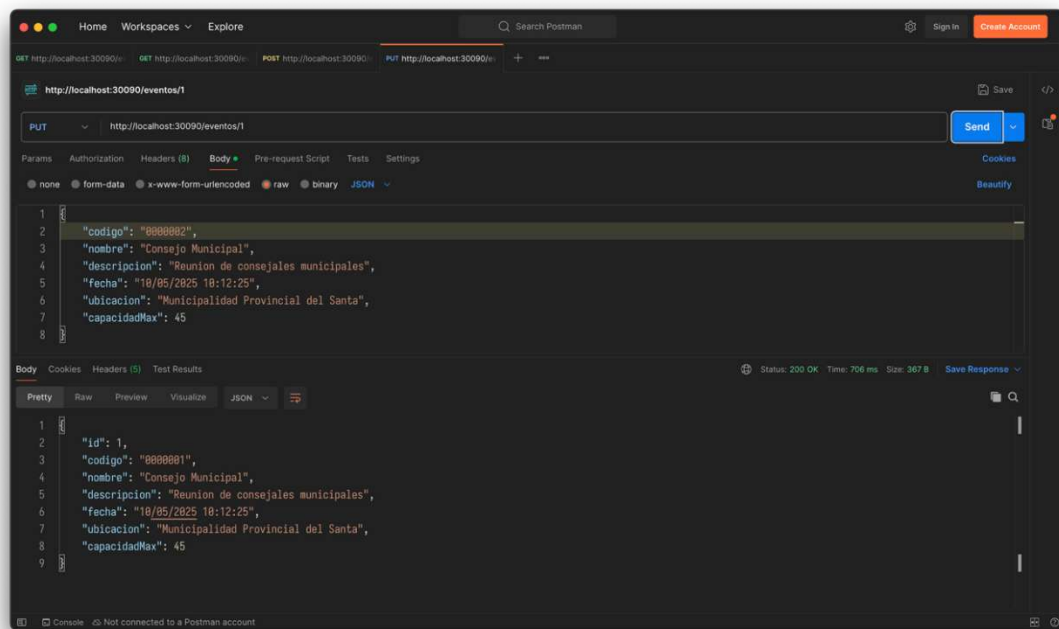
```
ms-eventos - zsh - 120x24
anthonyponte@basement ms-eventos % kubectl get pods | grep ms
ms-eventos-deployment-64f4b5fdb-jbssv 1/1 Running 0 2m38s
anthonyponte@basement ms-eventos %
```

```
mnadmin@Master: ~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
ms-deseos-77f568748-9sht4           1/1     Running   1 (52m ago) 5d3h
ms-eventos-5c5ddb6b8f-1fqjm         1/1     Running   3 (49m ago) 5d2h
ms-participantes-59c47f7598-x7nlp   1/1     Running   3 (49m ago) 5d2h
ms-productos-766bd8595f-8dhtx       1/1     Running   5 (49m ago) 5d3h
nginx-bf5d5cf98-h5664               1/1     Running   0           5d3h
oracle-db-cbf654876-8c8jm           1/1     Running   1 (52m ago) 5d3h
mnadmin@Master:~$
```

- Pruebas realizadas en los servicios mediante Postman o CURL.







7.2. Registro de Imágenes:

- URL de las imágenes Docker publicadas (Docker Hub).

ms-eventos:

<https://hub.docker.com/repository/docker/jonathan0284/ms-eventos/general>

ms-participantes:

<https://hub.docker.com/repository/docker/jonathan0284/ms-participantes/general>