



# INSTITUTO TECNOLÓGICO SUPERIOR DE JEREZ

Libramiento Fresnillo-Tepetongo, Fracc. Los Cardos,  
99863 Jerez de García Salinas, Zac.

GARCIA FLORES ANAHI

anahi.gf@tecjerez.edu.mx

INGENIERÍA EN SISTEMAS  
COMPUTACIONALES

DevSecOps - Desarrollo, Seguridad y Operaciones de  
Software

Prof. Salvador Acevedo

9° SEMESTRE

EVALUACION PARCIAL 2

Fecha:10/10/25

# EVALUACIÓN PARCIAL 2

Index

**Objetivo.....2**

**Desarrollo..... 3**

    Despliegue de servicios con Kubernetes.....3

    Ejecución en Kubernetes..... 3

    Resultados.....4

**Conclusiones..... 5**

## Objetivo

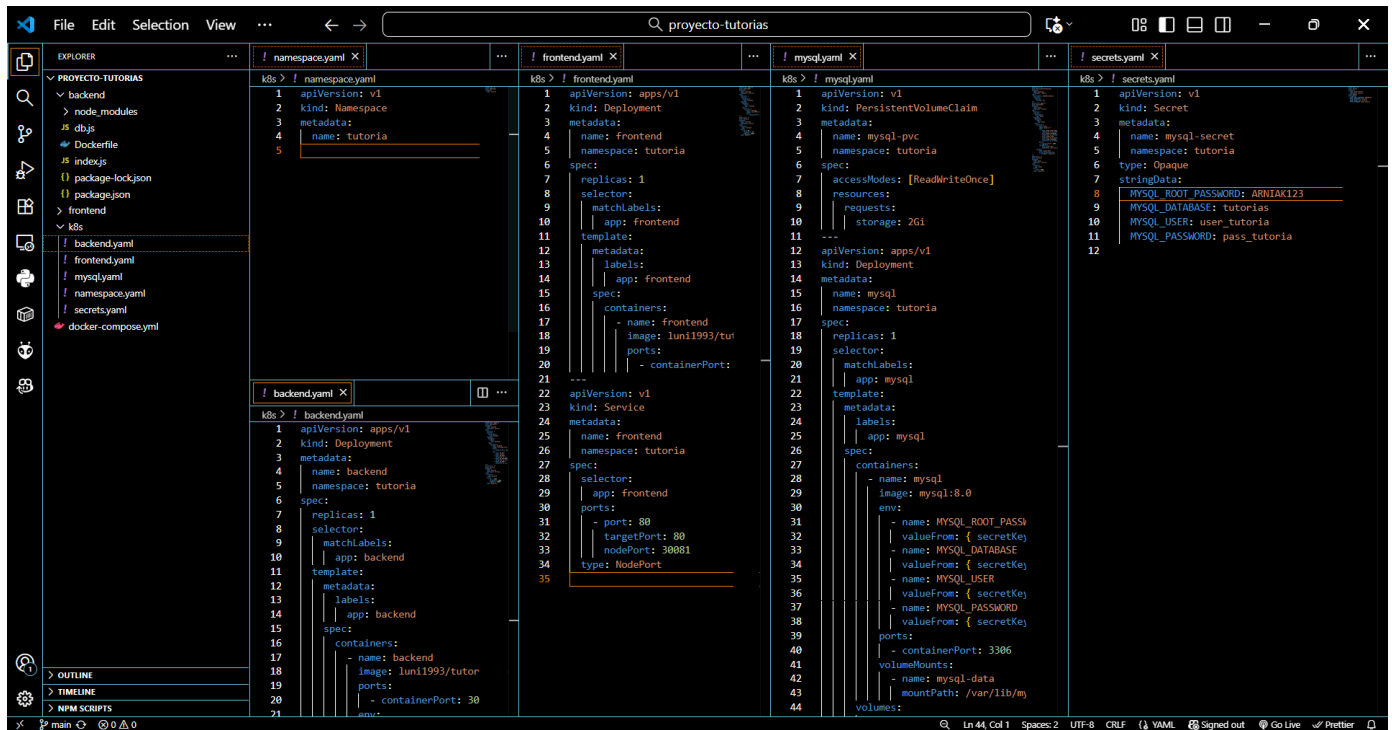
Realizar la orquestación de servicios para una aplicación web institucional del Área de Tutorías del ITSJ, implementando contenedores para el backend, frontend y base de datos, desplegados mediante Kubernetes (Minikube).

# Desarrollo

Link github: <https://github.com/anhigf/Proyecto.git>

## Despliegue de servicios con Kubernetes

Se orquestaron tres servicios: MySQL, Backend, y Frontend, dentro del namespace tutoriala.



The screenshot shows a VS Code editor with five Kubernetes manifest files open in the 'namespace tutoriala' namespace. The Explorer sidebar on the left shows the project structure: PROYECTO-TUTORIAS, backend, frontend, and k8s. The k8s folder contains the following files: namespace.yaml, frontend.yaml, mysql.yaml, backend.yaml, and secrets.yaml. The main editor area displays the content of these files. The namespace.yaml file defines the namespace 'tutoriala'. The frontend.yaml file defines a Deployment for the 'frontend' service. The mysql.yaml file defines a PersistentVolumeClaim for 'mysql-pvc' and a Deployment for the 'mysql' service. The backend.yaml file defines a Deployment for the 'backend' service. The secrets.yaml file defines secrets for 'MYSQL\_ROOT\_PASSWORD', 'MYSQL\_DATABASE', 'MYSQL\_USER', and 'MYSQL\_PASSWORD'.

```
namespace.yaml
1 apiVersion: v1
2 kind: Namespace
3 metadata:
4   name: tutoriala

frontend.yaml
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: frontend
5 namespace: tutoriala
6 spec:
7   replicas: 1
8   selector:
9     matchLabels:
10      app: frontend
11   template:
12     metadata:
13       labels:
14         app: frontend
15     spec:
16       containers:
17         - name: frontend
18           image: lun1993/tutor
19           ports:
20             - containerPort: 3000

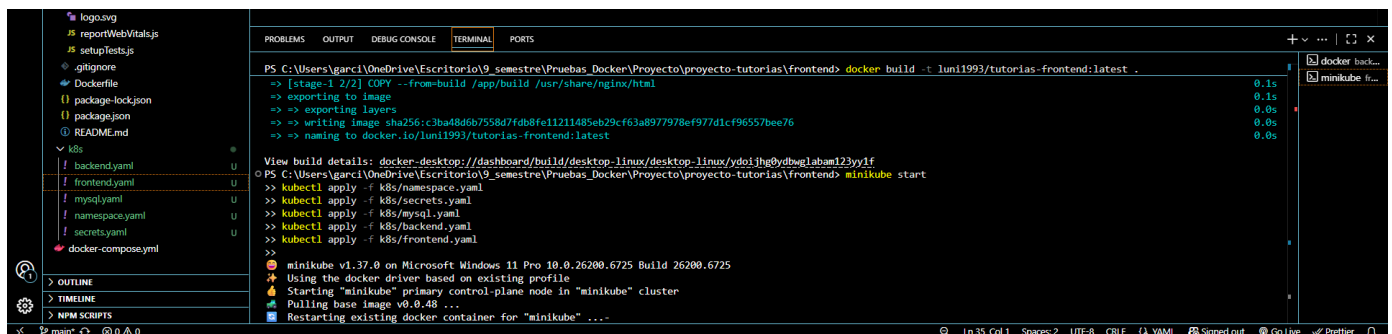
mysql.yaml
1 apiVersion: v1
2 kind: PersistentVolumeClaim
3 metadata:
4   name: mysql-pvc
5 namespace: tutoriala
6 spec:
7   accessModes: [ReadWriteOnce]
8   resources:
9     requests:
10      storage: 2Gi
11 ---
12 apiVersion: apps/v1
13 kind: Deployment
14 metadata:
15   name: mysql
16 namespace: tutoriala
17 spec:
18   replicas: 1
19   selector:
20     matchLabels:
21      app: mysql
22   template:
23     metadata:
24       labels:
25         app: mysql
26     spec:
27       containers:
28         - name: mysql
29           image: mysql:8.0
30           env:
31             - name: MYSQL_ROOT_PASSWORD
32               valueFrom: { secretKey: MYSQL_ROOT_PASSWORD }
33             - name: MYSQL_DATABASE
34               valueFrom: { secretKey: MYSQL_DATABASE }
35             - name: MYSQL_USER
36               valueFrom: { secretKey: MYSQL_USER }
37             - name: MYSQL_PASSWORD
38               valueFrom: { secretKey: MYSQL_PASSWORD }
39           ports:
40             - containerPort: 3306
41       volumeMounts:
42         - name: mysql-data
43           mountPath: /var/lib/mysql
44       volumes:

backend.yaml
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: backend
5 namespace: tutoriala
6 spec:
7   replicas: 1
8   selector:
9     matchLabels:
10      app: backend
11   template:
12     metadata:
13       labels:
14         app: backend
15     spec:
16       containers:
17         - name: backend
18           image: lun1993/tutor
19           ports:
20             - containerPort: 3000

secrets.yaml
1 apiVersion: v1
2 kind: Secret
3 metadata:
4   name: mysql-secret
5 namespace: tutoriala
6 type: Opaque
7 stringData:
8   MYSQL_ROOT_PASSWORD: ANHIAK123
9   MYSQL_DATABASE: tutoriala
10  MYSQL_USER: user.tutoriala
11  MYSQL_PASSWORD: pass.tutoriala
```

## Ejecución en Kubernetes

1. Se aplicaron los YAML.
2. Se verificó el estado de los pods.
3. Se verificaron los servicios y puertos.



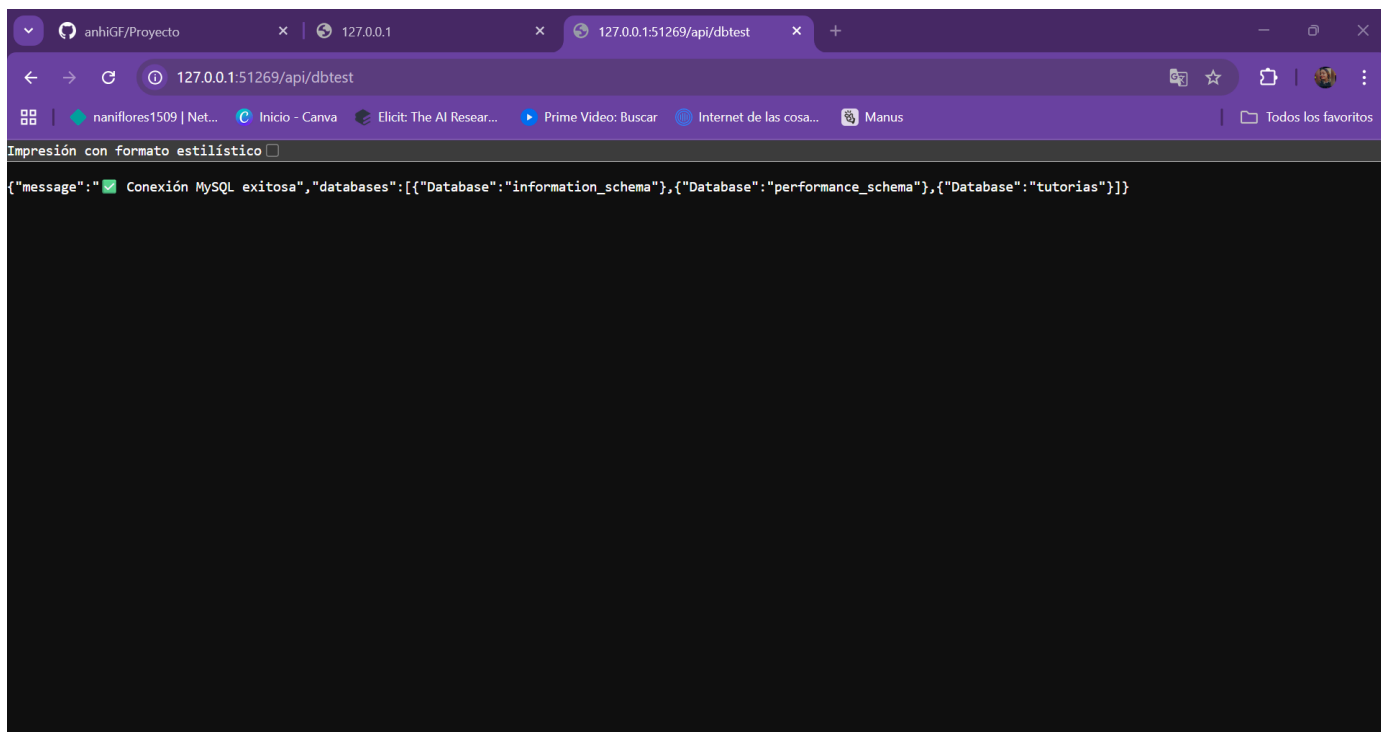
The screenshot shows a VS Code editor with a terminal window open. The terminal displays the output of the following commands: `docker build -t lun1993/tutorias-frontend:latest .`, `minikube start`, and `kubectl apply -f k8s/namespace.yaml`, `kubectl apply -f k8s/secrets.yaml`, `kubectl apply -f k8s/mysql.yaml`, `kubectl apply -f k8s/backend.yaml`, and `kubectl apply -f k8s/frontend.yaml`. The terminal also shows the output of `minikube status` and `minikube logs`. The Explorer sidebar on the left shows the project structure: PROYECTO-TUTORIAS, backend, frontend, and k8s. The k8s folder contains the following files: namespace.yaml, frontend.yaml, mysql.yaml, backend.yaml, and secrets.yaml. The main editor area displays the content of these files.

```
PS C:\Users\garcia\OneDrive\Escritorio\9 semestre\Pruebas Docker\Proyecto\proyecto-tutorias\frontend> docker build -t lun1993/tutorias-frontend:latest .
=> [stage-1 2/2] COPY --from=build /app/build /usr/share/nginx/html
=> exporting to image
=> exporting layers
=> writing image sha256:c3ba48d6b7558d7fdb8fe11211485eb29cf63a8977978ef977d1cf96557bee76
=> naming to docker.io/lun1993/tutorias-frontend:latest

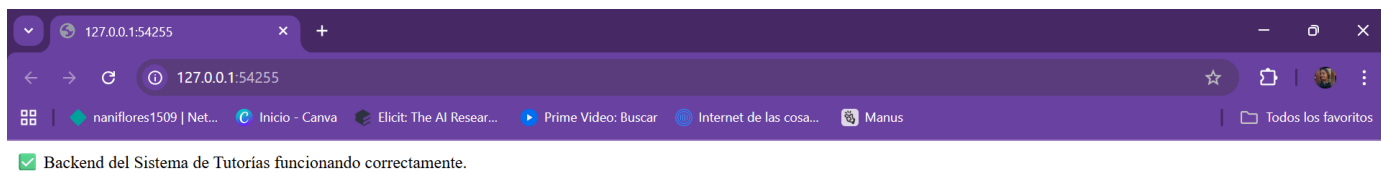
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/ydoi1hg0ydbwlabam123yy1f
PS C:\Users\garcia\OneDrive\Escritorio\9 semestre\Pruebas Docker\Proyecto\proyecto-tutorias\frontend> minikube start
>> kubectl apply -f k8s/namespace.yaml
>> kubectl apply -f k8s/secrets.yaml
>> kubectl apply -f k8s/mysql.yaml
>> kubectl apply -f k8s/backend.yaml
>> kubectl apply -f k8s/frontend.yaml
>> minikube v1.37.0 on Microsoft Windows 11 Pro 10.0.22000.6725 Build 26200.6725
>> Using the docker driver based on existing profile
>> Starting "minikube" primary control-plane node in "minikube" cluster
>> Pulling base image v0.0.48 ...
>> Restarting existing docker container for "minikube" ....
```

## Resultados

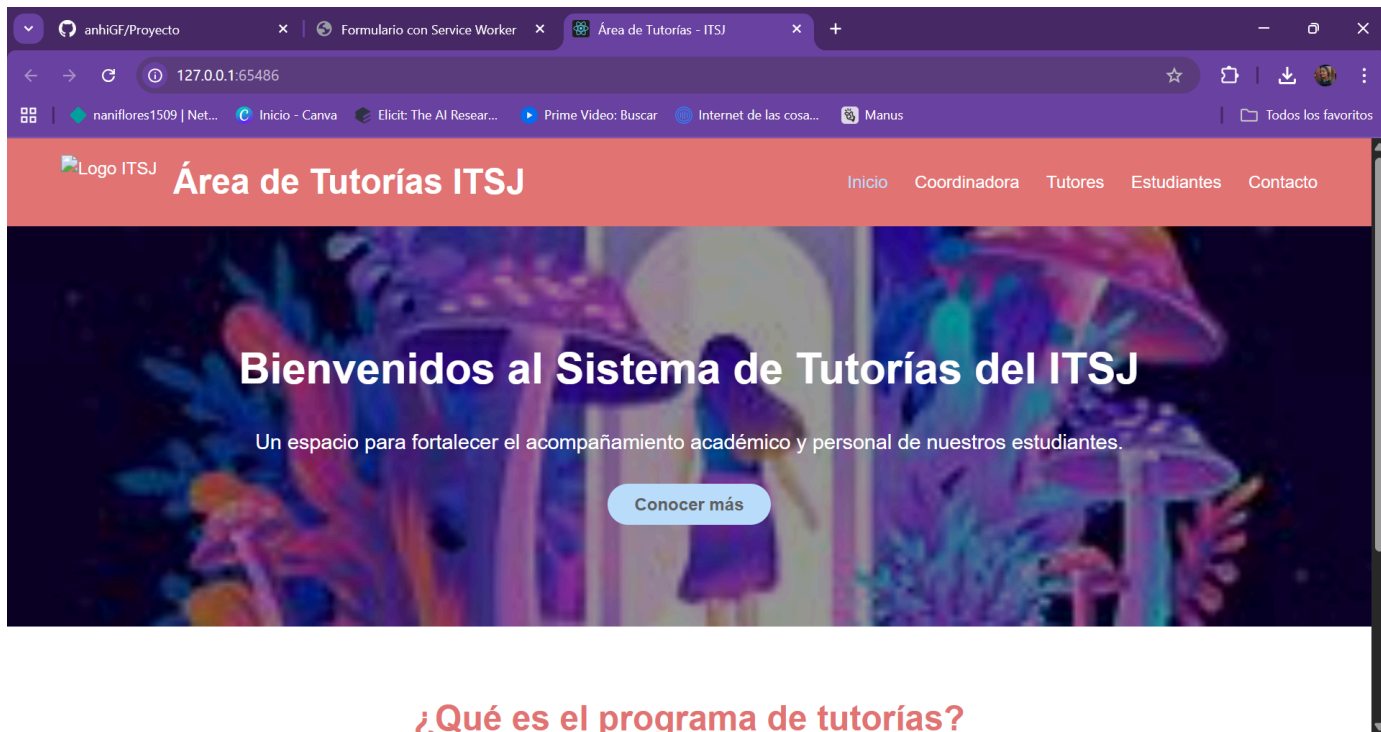
- SQL



- Backend



- Frontend: Acceso exitoso a la aplicación



## Conclusiones

- Se logró desplegar exitosamente una arquitectura multicapa: frontend, backend y base de datos.
- Se fortalecieron las habilidades de configuración de Dockerfiles, YAMLs de despliegue, y administración de pods y servicios en Kubernetes.
- Minikube resultó una herramienta ideal para simular un entorno real de clúster en un entorno local.
- Finalmente, se comprobó la comunicación efectiva entre los servicios y la persistencia de datos mediante MySQL dentro del clúster.