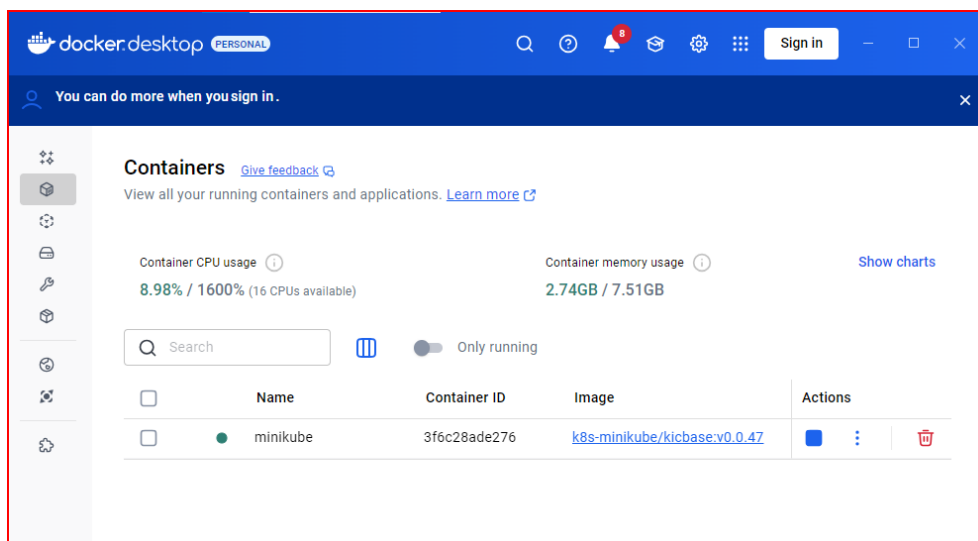


Despliegue del Proyecto en Kubernetes - MINIKUBE

1. Habilitar Minikube en Docker desktop

minikube start --driver=docker

```
Administrador: Windows PowerShell
PS C:\Windows\system32> minikube start --driver=docker
* minikube v1.36.0 en Microsoft Windows 10 Pro 10.0.19045.2364 Build 19045.2364
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.47 ...
* Restarting existing docker container for "minikube" ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparando Kubernetes v1.33.1 en Docker 28.1.1...
* Verifying Kubernetes components...
- Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Complementos habilitados: storage-provisioner, default-storageclass
* Done! kubect1 is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Windows\system32>
```



2. Repositorio de charts

<https://rmcabrera.github.io/neuromotion-charts/>

Chart	Versión	Imagen	Descripción
ms-prerequisites	1.0.0	—	Crea el namespace ms-app y el Secret con credenciales DB
mysql-doctores	0.1.0	8.0	Base de datos MySQL para el microservicio ms-doctores
mysql-usuarios	0.1.0	8	Base de datos MySQL para el microservicio ms-usuarios
ms-doctores	1.0.0	v2	Microservicio Spring Boot ms-doctores
ms-usuarios	1.0.0	v2	Microservicio Spring Boot ms-usuarios
neuromotion-frontend	0.1.0	v1	Frontend de Neuromotion (Angular)

3. Despliegue en Kubernetes

Paso 1: Agregar el repositorio Helm

```
helm repo add neuromotion https://rmcabrera.github.io/neuromotion-charts/  
helm repo update
```

Paso 2: Instalar los pre-requisitos

Instala el chart `ms-prerequisites` para crear el namespace `ms-app` y el Secret con las credenciales de la base de datos.

```
helm install ms-prerequisites neuromotion/ms-prerequisites
```

Paso 3: Instalar las bases de datos

Instala las bases de datos MySQL para los microservicios `ms-doctores` y `ms-usuarios`.

```
helm install mysql-doctores neuromotion/mysql-doctores -n ms-app  
helm install mysql-usuarios neuromotion/mysql-usuarios -n ms-app
```

Paso 4: Instalar los microservicios

Instala los microservicios `ms-doctores` y `ms-usuarios`.

```
helm install ms-doctores neuromotion/ms-doctores -n ms-app  
helm install ms-usuarios neuromotion/ms-usuarios -n ms-app
```

Paso 5: Instalar el Frontend

Instala el frontend de Neuromotion.

```
helm install neuromotion-frontend neuromotion/neuromotion-frontend -n ms-app
```

Paso 6: Verificación de PODs

```
kubectl get pods -n ms-app
```

```
Administrador: Windows PowerShell
PS C:\Windows\system32> kubectl get pods -n ms-app
NAME                                READY   STATUS    RESTARTS   AGE
ms-doctores-966665547-gsm4m         1/1     Running   2 (52m ago) 4h16m
ms-doctores-966665547-mpvbw         1/1     Running   2 (52m ago) 4h16m
ms-usuarios-6944b88458-14d6b        1/1     Running   2 (52m ago) 4h14m
ms-usuarios-6944b88458-tnd6c        1/1     Running   2 (52m ago) 4h14m
mysql-doctores-84467cdb59-ttdf2     1/1     Running   2 (52m ago) 4h17m
mysql-usuarios-54fb7bd54f-48zb4     1/1     Running   2 (52m ago) 4h16m
neuromotion-frontend-6b4c456669-x715x 1/1     Running   4 (51m ago) 3h12m
PS C:\Windows\system32>
```

kubectl get deployments -n ms-app

```
Administrador: Windows PowerShell
PS C:\Windows\system32> kubectl get deployments -n ms-app
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
ms-doctores          2/2     2             2           4h18m
ms-usuarios          2/2     2             2           4h17m
mysql-doctores       1/1     1             1           4h20m
mysql-usuarios       1/1     1             1           4h19m
neuromotion-frontend 1/1     1             1           3h14m
PS C:\Windows\system32>
```

Paso 7: Acceder al Frontend

Una vez desplegado el frontend, puedes acceder a él a través de Minikube:

```
minikube service neuromotion-frontend-service -n ms-app
```

Paso 8: Prueba de Endpoints

ms-doctores

```
# Port forward
kubectl port-forward -n ms-app svc/ms-doctores-service 8082:8082

# Listar especialidades
curl http://localhost:8082/api/especialidades

# Obtener especialidad por ID
curl http://localhost:8082/api/especialidades/1

# Crear especialidad
curl -X POST -H "Content-Type: application/json" -d '{"nombre\":"Cardiología\"}' http://localhost:8082/api/especialidades

# Actualizar especialidad
curl -X PUT -H "Content-Type: application/json" -d '{"nombre\":"Cardiología Avanzada\"}' http://localhost:8082/api/especialidades/1

# Eliminar especialidad
```

```
curl -X DELETE http://localhost:8082/api/especialidades/1

# Listar doctores
curl http://localhost:8082/api/doctores

# Obtener doctor por ID
curl http://localhost:8082/api/doctores/1

# Crear doctor
curl -X POST -H "Content-Type: application/json" -d '{"nombre": "Juan Perez", "licencia": "12345", "email": "juan@example.com", "especialidad": {"id": 1}}' http://localhost:8082/api/doctores

# Actualizar doctor
curl -X PUT -H "Content-Type: application/json" -d '{"nombre": "Juan Perez", "licencia": "54321", "email": "juanp@example.com", "especialidad": {"id": 1}}' http://localhost:8082/api/doctores/1

# Eliminar doctor
curl -X DELETE http://localhost:8082/api/doctores/1
```

ms-usuarios

```
# Port forward
kubectl port-forward -n ms-app svc/ms-usuarios-service 8081:8081

# Listar usuarios
curl http://localhost:8081/api/usuarios

# Obtener usuario por ID
curl http://localhost:8081/api/usuarios/1

# Crear usuario
curl -X POST -H "Content-Type: application/json" -d '{"nombre": "Alexander", "apellido": "Gonzales", "email": "as.dsas@neuromotion.pe", "telefono": "987654321"}' http://localhost:8081/api/usuarios

# Actualizar usuario
curl -X PUT -H "Content-Type: application/json" -d '{"nombre": "Alexander R.", "apellido": "Gonzales", "email": "agonzales@neuromotion.pe", "telefono": "987654321"}' http://localhost:8081/api/usuarios/1

# Eliminar usuario
curl -X DELETE http://localhost:8081/api/usuarios/1

# Listar citas
curl http://localhost:8081/api/citas

# Obtener cita por ID
curl http://localhost:8081/api/citas/1

# Crear cita
curl -X POST -H "Content-Type: application/json" -d '{"fechaHora": "2025-06-25T15:30:00", "usuarioId": 1, "doctorId": 2, "motivo": "Consulta de terapia cognitiva"}' http://localhost:8081/api/citas
```

```
# Actualizar cita
curl -X PUT -H "Content-Type: application/json" -d "{\"fechaHora\": \"2025-06-26T16:00:00\", \"usuarioId\": 1, \"doctorId\": 2, \"motivo\": \"Seguimiento de terapia\"}" http://localhost:8081/api/citas/1

# Eliminar cita
curl -X DELETE http://localhost:8081/api/citas/1
```

Paso 9: Desinstalación

Para desinstalar todos los charts y limpiar los recursos:

```
helm uninstall ms-doctores -n ms-app
helm uninstall ms-usuarios -n ms-app
helm uninstall neuromotion-frontend -n ms-app
helm uninstall mysql-doctores -n ms-app
helm uninstall mysql-usuarios -n ms-app
kubectl delete pvc --all -n ms-app
kubectl delete pv --all
helm uninstall ms-prerequisites
```

Limpieza adicional (Opcional)

Para limpiar el repositorio Helm local y las imágenes de Docker en Minikube:

```
helm repo remove neuromotion
```

Para limpiar las imágenes de Docker en Minikube:

```
minikube ssh
docker images | grep ms-usuarios # Puedes ajustar el filtro según las
imágenes que quieras ver
docker rmi mcabrerac/ms-doctores:v2
docker rmi mcabrerac/ms-usuarios:v2
docker rmi mcabrerac/neuromotion-frontend:v1
exit
```