



# INFORME DEL PROYECTO

---

Despliegue de Aplicación Web en  
Kubernetes con Minikube

## Redes II

Carrera: Analista en Sistemas  
Institución: Universidad Champagnat  
Profesor: José Scattareggia

Autores: Enzo Golbano - Lucas Sosa  
Año: 2025

## 1. Introducción

El presente trabajo consiste en el despliegue de una aplicación web estática dentro de un entorno de contenedores utilizando **Docker**, **Minikube** y **Kubernetes**.

El objetivo principal es comprender el funcionamiento del ecosistema de orquestación de contenedores, aplicando conceptos como la creación de imágenes, Pods, Deployments y Services.

El proyecto demuestra el flujo completo de despliegue:

1. Construcción de una imagen Docker personalizada
2. Ejecución de la imagen dentro de un clúster Kubernetes local
3. Exposición de la aplicación mediante un Service de tipo NodePort
4. Acceso externo a la aplicación desde el navegador del host

## 2. Tecnologías utilizadas

Tecnología	Función
Docker Desktop	Construcción y gestión de imágenes de contenedores
NGINX	Servidor web que sirve la aplicación estática
Minikube	Clúster Kubernetes local para desarrollo
Kubernetes	Orquestación, escalado y gestión de aplicaciones
kubectl	Herramienta de administración del clúster
PowerShell / VS Code	Consola y entorno de trabajo

### 3. Estructura del Proyecto

```
Proyecto-kubernetes-Enzo-Lucas/
|
├── app/
│   └── index.html           # Página web estática
|
├── Dockerfile               # Imagen Docker personalizada basada en NGINX
|
├── deployment.yaml          # Deployment de Kubernetes (crea el Pod)
|
├── service.yaml             # Service NodePort (exposición externa)
|
└── README.md                # Documentación del proyecto
```

### 4. Desarrollo del Proyecto

#### 4.1 Construcción de la imagen Docker

Se utilizó un **Dockerfile** basado en la imagen oficial de NGINX:

```
FROM nginx:alpine
COPY ./app/index.html /usr/share/nginx/html/index.html
```

Esto genera la imagen personalizada:

```
docker build -t miweb:1.0 .
```

#### 4.2 Despliegue en Minikube

Se activó el daemon de Docker dentro del clúster mediante:

```
minikube docker-env
```

Luego se construyó la imagen dentro del entorno de Minikube para poder utilizarla sin necesidad de subirla a un repositorio externo.

### 4.3 Creación del Deployment

Archivo: `deployment.yaml`

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: miweb
spec:
  replicas: 1
  selector:
    matchLabels:
      app: miweb
  template:
    metadata:
      labels:
        app: miweb
    spec:
      containers:
        - name: miweb
          image: miweb:1.0
          ports:
            - containerPort: 80
```

El Deployment se encarga de:

- Crear y mantener el Pod
- Ejecutar el contenedor
- Asegurar disponibilidad (Self-healing)
- Gestionar réplicas y actualizaciones

Se aplica con:

```
kubectl apply -f deployment.yaml
```

## 4.4 Exposición del servicio

Archivo: `service.yaml`

```
apiVersion: v1
kind: Service
metadata:
  name: miweb-service
spec:
  type: NodePort
  selector:
    app: miweb
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30080
```

Permite acceder a la aplicación desde fuera del clúster.

Se aplica con:

```
kubectl apply -f service.yaml
```

## 4.5 Acceso final a la aplicación

Desde Windows, con driver Docker, la forma correcta es:

```
minikube service miweb-service
```

Esto crea un túnel y abre automáticamente la página en el navegador.

## 5. Funcionamiento General

El flujo del proyecto puede resumirse así:

1. **Docker** empaqueta la aplicación (HTML + NGINX)
2. **Minikube** ejecuta Kubernetes localmente
3. **Kubernetes** crea un Pod mediante un Deployment
4. **Kubernetes** expone la aplicación mediante un Service
5. **El usuario accede a la aplicación web** desde el navegador

## 6. Conclusión

El proyecto permitió comprender y aplicar los conceptos fundamentales de contenedores y orquestación mediante Kubernetes. Se logró desplegar una aplicación funcional, accesible desde el host, validando el flujo completo: **imagen → pod → servicio → acceso externo**.

Esto demuestra el uso correcto de herramientas modernas utilizadas en entornos profesionales de despliegue y DevOps.