

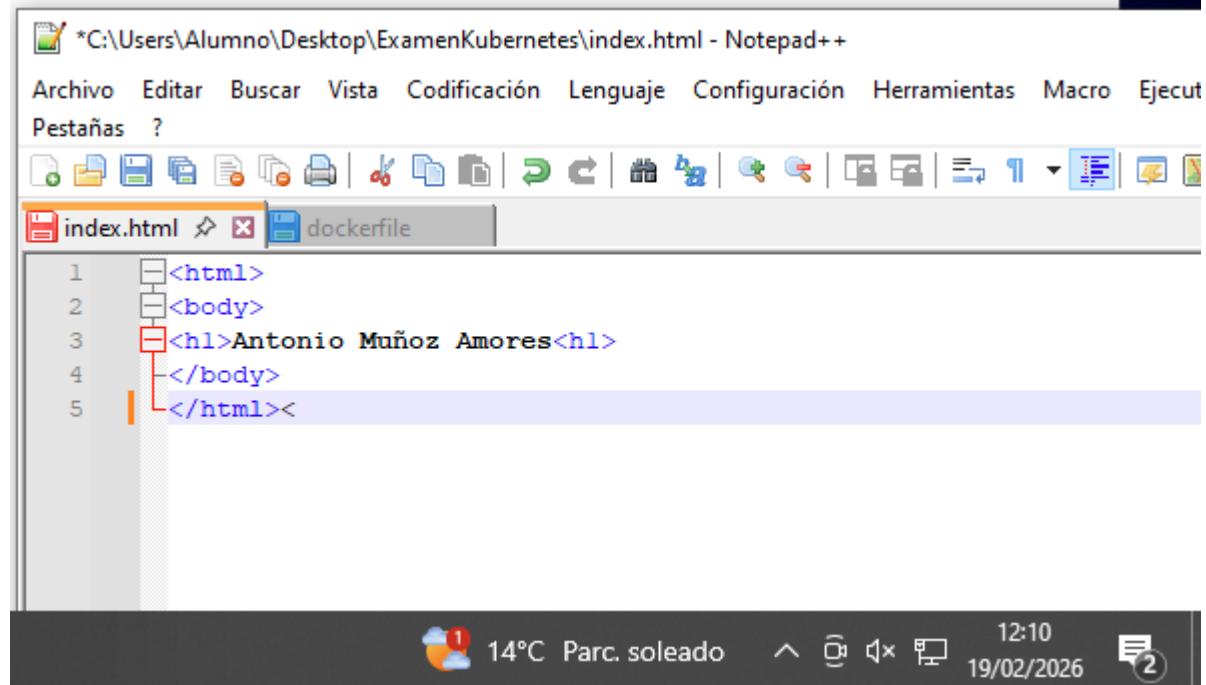
EXAMEN FINAL DOCKER

C A M P A Ñ A P R I N C I P A L

Antonio Muñoz Amores

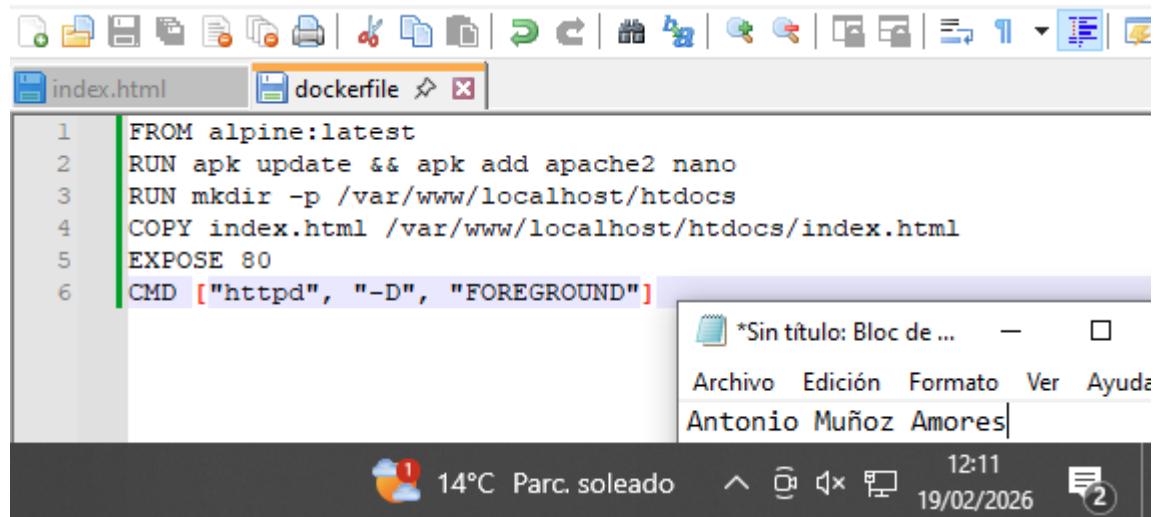
1. Preparación del entorno y de la imagen

Primero creamos el html



```
*C:\Users\Alumno\Desktop\ExamenKubernetes\index.html - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecut
Pestañas ?
index.html dockerfile
1 <html>
2 <body>
3 <h1>Antonio Muñoz Amores</h1>
4 </body>
5 </html>
```

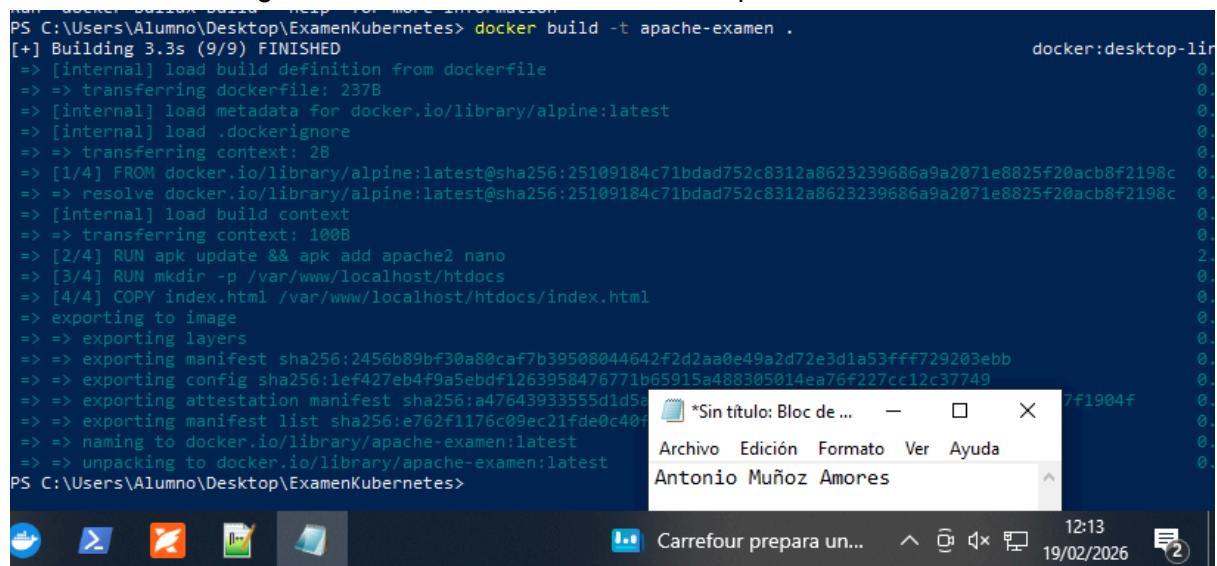
Después creamos el siguiente dockerfile



```
FROM alpine:latest
RUN apk update && apk add apache2 nano
RUN mkdir -p /var/www/localhost/htdocs
COPY index.html /var/www/localhost/htdocs/index.html
EXPOSE 80
CMD ["httpd", "-D", "FOREGROUND"]
```

Construimos la imagen con el comando docker build -t apache-examen .

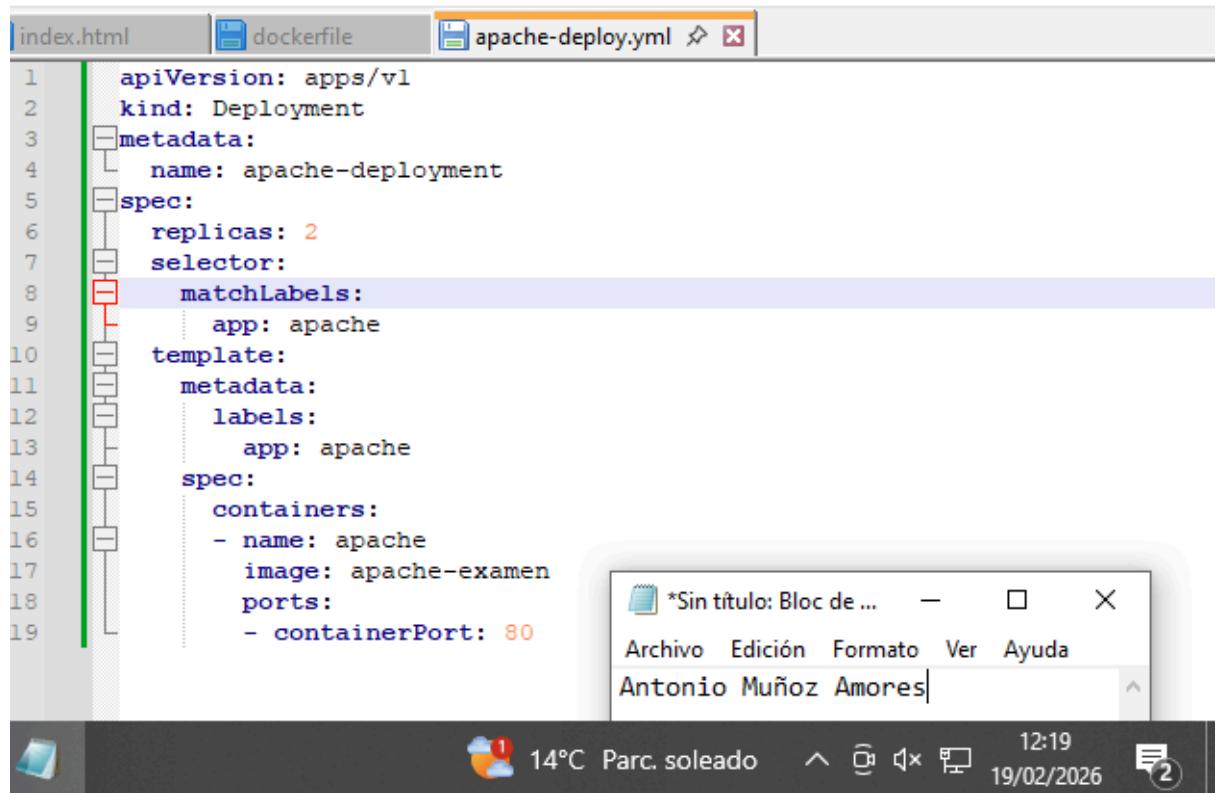
```
PS C:\Users\Alumno\Desktop\ExamenKubernetes> docker build -t apache-examen .
[+] Building 3.3s (9/9) FINISHED
  => [internal] load build definition from dockerfile
  => => transferring dockerfile: 237B
  => [internal] load metadata for docker.io/library/alpine:latest
  => [internal] load .dockerignore
  => => transferring context: 28
  => [1/4] FROM docker.io/library/alpine:latest@sha256:25109184c71bdad752c8312a8623239686a9a2071e8825f20acb8f2198c
  => => resolve docker.io/library/alpine:latest@sha256:25109184c71bdad752c8312a8623239686a9a2071e8825f20acb8f2198c
  => [internal] load build context
  => => transferring context: 100B
  => [2/4] RUN apk update && apk add apache2 nano
  => [3/4] RUN mkdir -p /var/www/localhost/htdocs
  => [4/4] COPY index.html /var/www/localhost/htdocs/index.html
  => exporting to image
  => => exporting layers
  => => exporting manifest sha256:2456b89bf30a80caf7b39508044642f2d2aa0e49a2d72e3d1a53fff729203ebb
  => => exporting config sha256:1ef427eb4f9a5ebdf1263958476771b65915a488305014ea76f227cc12c37749
  => => exporting attestation manifest sha256:a47643933555d1d5a
  => => exporting manifest list sha256:e762f1176c09ec21fde0c40f
  => => naming to docker.io/library/apache-examen:latest
  => => unpacking to docker.io/library/apache-examen:latest
PS C:\Users\Alumno\Desktop\ExamenKubernetes>
```



2. Despliegue de kubernetes

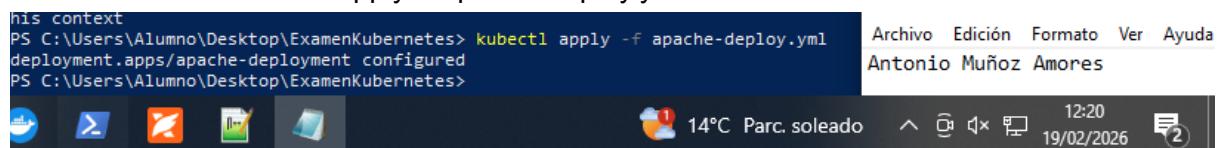
Creamos el siguiente archivo de despliegue

```
index.html dockerfile apache-deploy.yml ✘ x
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: apache-deployment
5  spec:
6    replicas: 2
7    selector:
8      matchLabels:
9        app: apache
10       template:
11         metadata:
12           labels:
13             app: apache
14         spec:
15           containers:
16             - name: apache
17               image: apache-examen
18               ports:
19                 - containerPort: 80
```

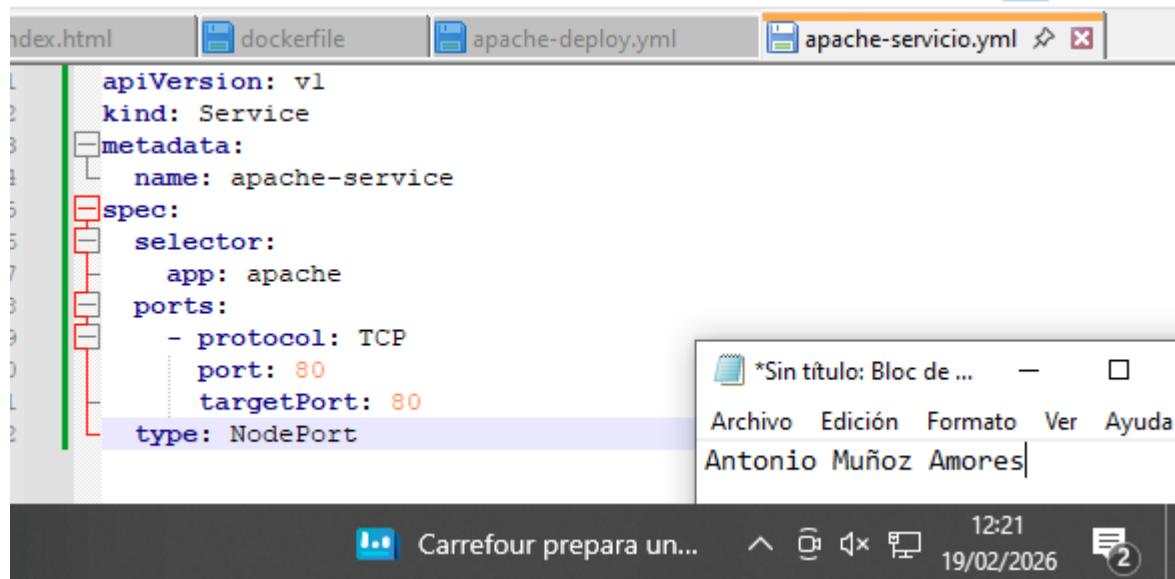


Lo lanzamos con kubectl apply -f apache-deploy.yml

```
PS C:\Users\Alumno\Desktop\ExamenKubernetes> kubectl apply -f apache-deploy.yml
deployment.apps/apache-deployment configured
PS C:\Users\Alumno\Desktop\ExamenKubernetes>
```

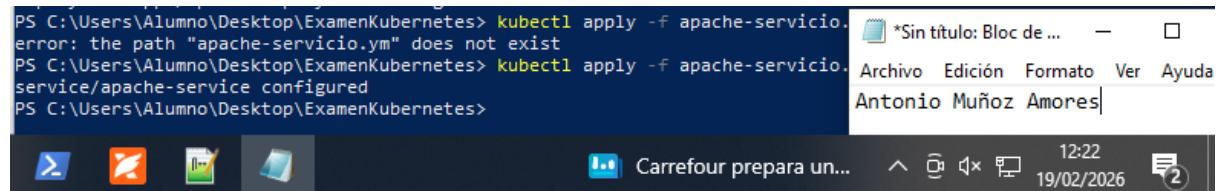


Creamos el siguiente archivo para el servicio de apache



```
apiVersion: v1
kind: Service
metadata:
  name: apache-service
spec:
  selector:
    app: apache
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: NodePort
```

Lo lanzamos con kubectl apply -f apache-servicio.yml

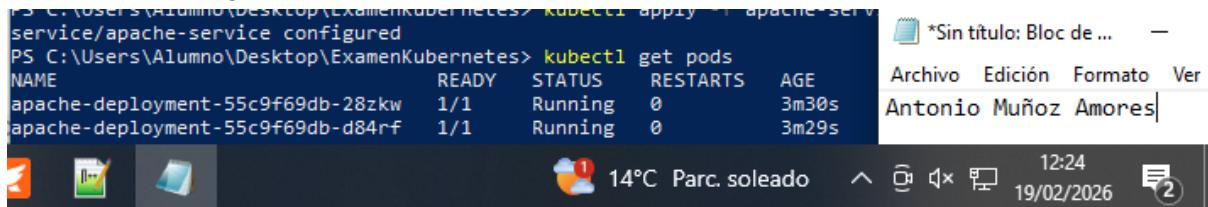


```
PS C:\Users\Alumno\Desktop\ExamenKubernetes> kubectl apply -f apache-servicio.yml
error: the path "apache-servicio.yml" does not exist
PS C:\Users\Alumno\Desktop\ExamenKubernetes> kubectl apply -f apache-servicio.yml
service/apache-service configured
PS C:\Users\Alumno\Desktop\ExamenKubernetes>
```

3. Comprobación del funcionamiento:

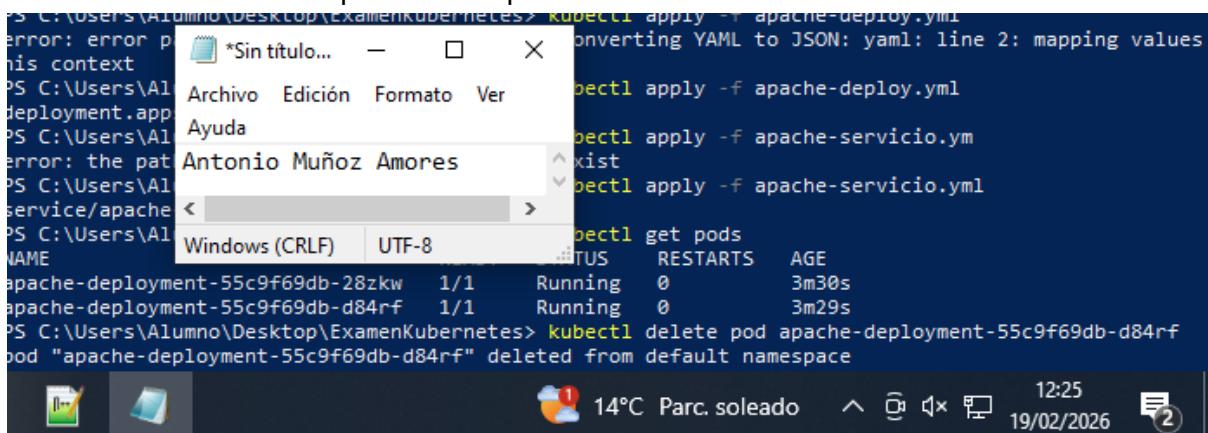
Con el comando get pods vemos que tenemos los dos pods creados anteriormente

```
C:\Users\Alumno\Desktop\ExamenKubernetes> kubectl apply -f apache-servi
service/apache-service configured
PS C:\Users\Alumno\Desktop\ExamenKubernetes> kubectl get pods
NAME READY STATUS RESTARTS AGE
apache-deployment-55c9f69db-28zkw 1/1 Running 0 3m30s
apache-deployment-55c9f69db-d84rf 1/1 Running 0 3m29s
```



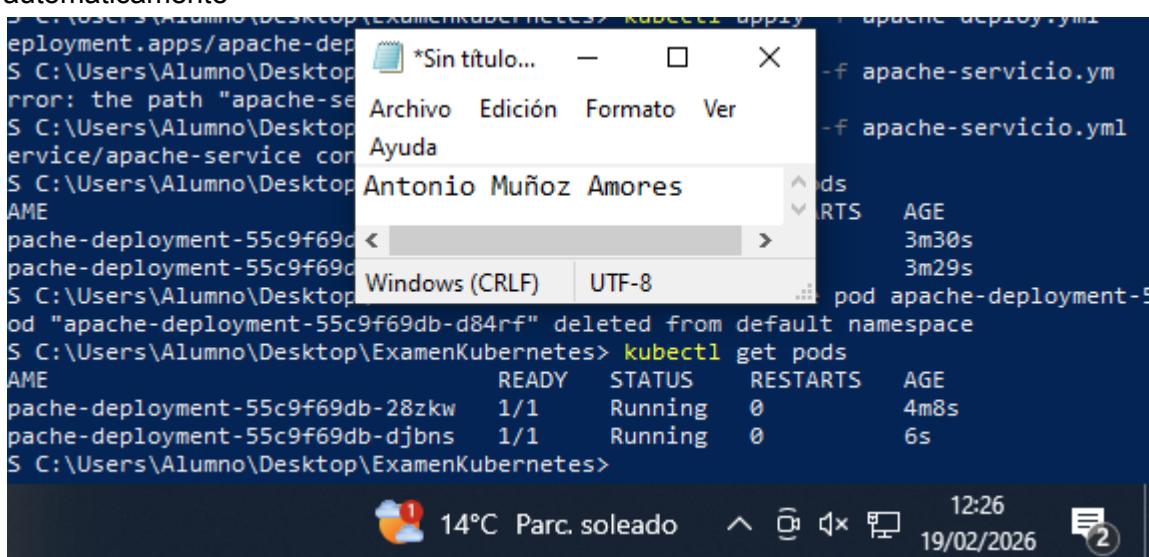
Borramos uno con delete-pod nombre-pod

```
C:\Users\Alumno\Desktop\ExamenKubernetes> kubectl apply -f apache-deploy.yml
error: error P
his context
PS C:\Users\Al
deployment.app
PS C:\Users\Al
error: the pat
PS C:\Users\Al
service/apache
PS C:\Users\Al
NAME
apache-deployment-55c9f69db-28zkw 1/1 Running 0 3m30s
apache-deployment-55c9f69db-d84rf 1/1 Running 0 3m29s
PS C:\Users\Alumno\Desktop\ExamenKubernetes> kubectl delete pod apache-deployment-55c9f69db-d84rf
pod "apache-deployment-55c9f69db-d84rf" deleted from default namespace
```

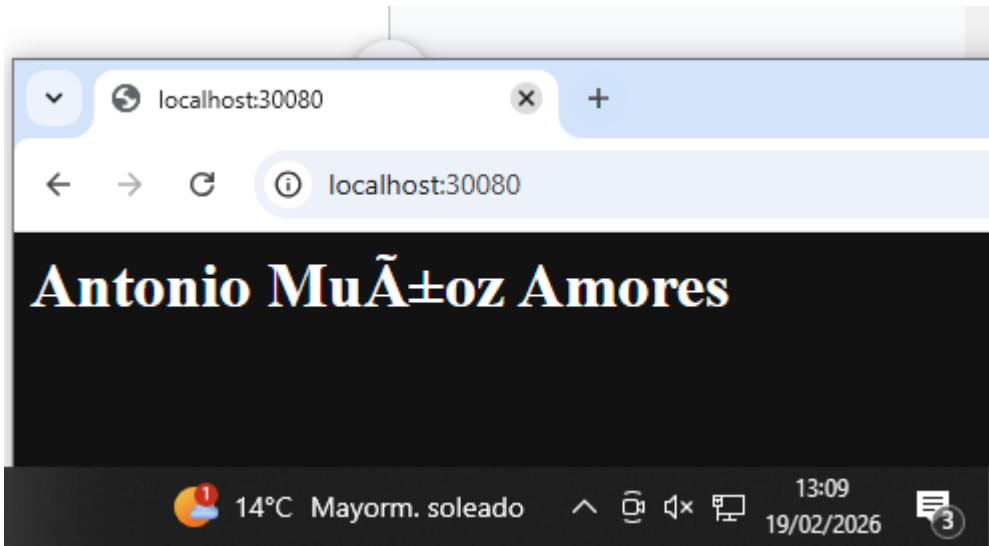


Ahora si volvemos a mirar los pods disponibles comprobamos que se ha creado otro automáticamente

```
C:\Users\Alumno\Desktop\ExamenKubernetes> kubectl apply -f apache-deploy.yml
deployment.apps/apache-dep
PS C:\Users\Alumno\Desktop
rror: the path "apache-se
PS C:\Users\Alumno\Desktop
ervice/apache-service cor
PS C:\Users\Alumno\Desktop
AME
apache-deployment-55c9f69d
apache-deployment-55c9f69d
PS C:\Users\Alumno\Desktop
od "apache-deployment-55c9f69db-d84rf" deleted from default namespace
PS C:\Users\Alumno\Desktop\ExamenKubernetes> kubectl get pods
NAME READY STATUS RESTARTS AGE
apache-deployment-55c9f69db-28zkw 1/1 Running 0 4m8s
apache-deployment-55c9f69db-djbns 1/1 Running 0 6s
PS C:\Users\Alumno\Desktop\ExamenKubernetes>
```



Comprobamos los logs de cuando se realiza una petición
Abrimos nuestra pagina buscando <http://localhost:30080> en el navegador



Y con el comando `kubectl exec -it nombre-contenedor -- tail -f /var/log/apache2/access.log` podemos ver las peticiones al servidor