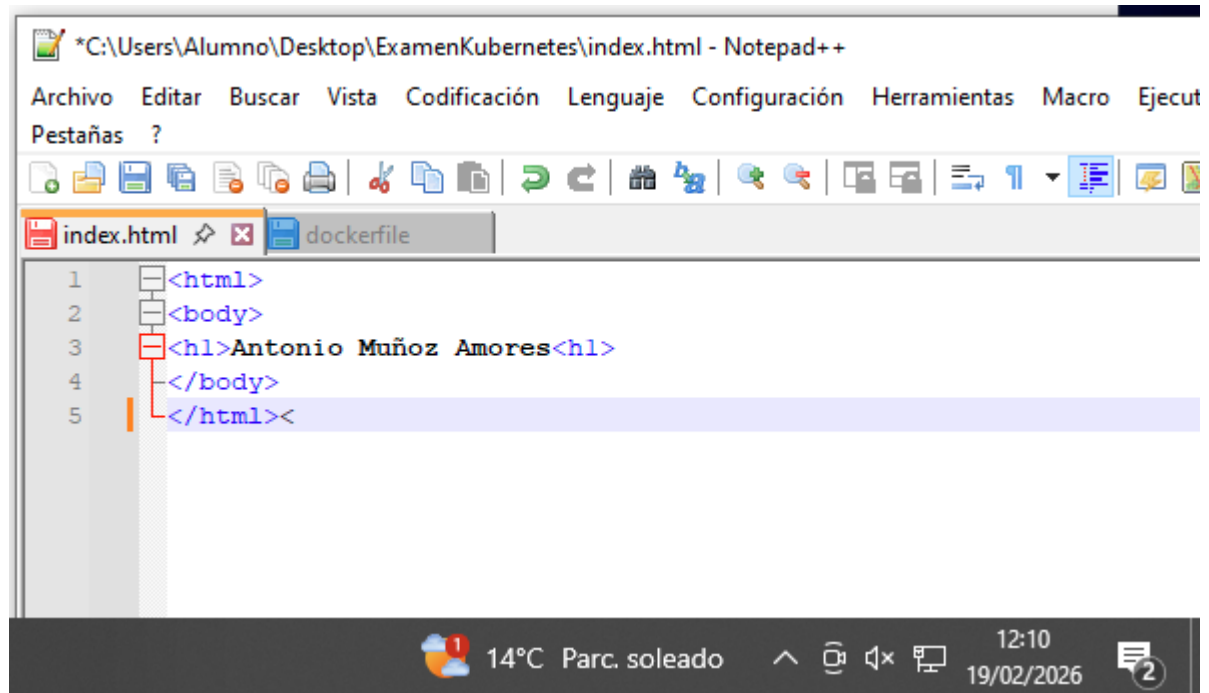


1. Preparación del entorno y de la imagen

Primero creamos el html

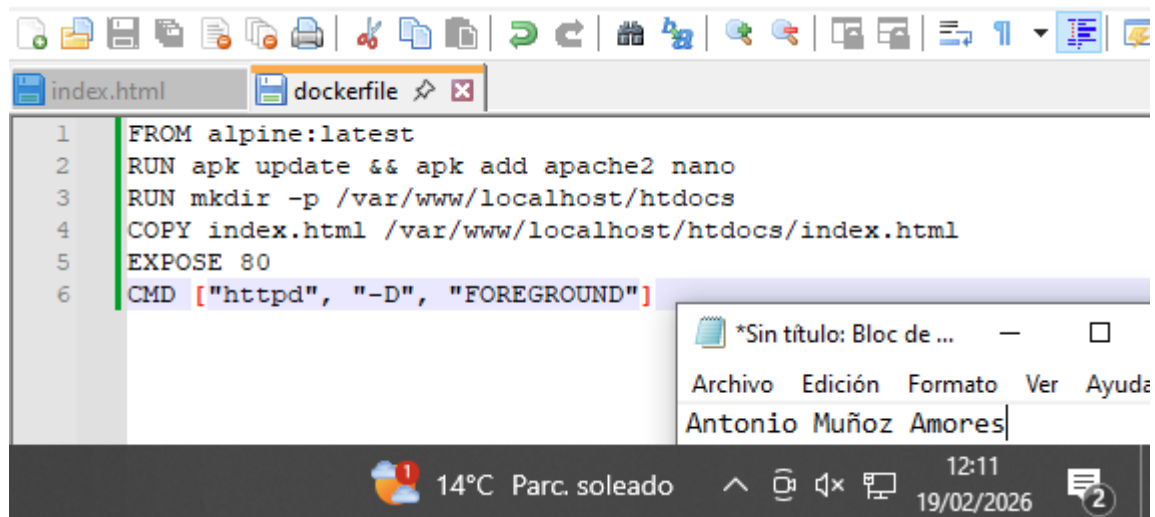


A screenshot of the Notepad++ application. The title bar reads '*C:\Users\Alumno\Desktop\ExamenKubernetes\index.html - Notepad++'. The menu bar includes Archivo, Editar, Buscar, Vista, Codificación, Lenguaje, Configuración, Herramientas, Macro, and Ejecutar. The toolbar contains various icons for file operations. The tab bar shows 'index.html' and 'dockerfile'. The editor area contains the following HTML code:

```
1 <html>
2 <body>
3 <h1>Antonio Muñoz Amores</h1>
4 </body>
5 </html><
```

The Windows taskbar at the bottom shows the weather as 14°C, 'Parc. soleado', and the date/time as 12:10 on 19/02/2026.

Después creamos el siguiente dockerfile



A screenshot of the Notepad++ application with the 'dockerfile' tab active. The editor area contains the following Dockerfile content:

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

A context menu is open over the text 'Antonio Muñoz Amores', showing options: Archivo, Edición, Formato, Ver, Ayuda, and the text 'Antonio Muñoz Amores'.

The Windows taskbar at the bottom shows the weather as 14°C, 'Parc. soleado', and the date/time as 12:11 on 19/02/2026.

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: 2B
=> [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:a47643933555d1d5
=> => exporting manifest list sha256:e762f1176c09ec21fde0c40
=> => 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

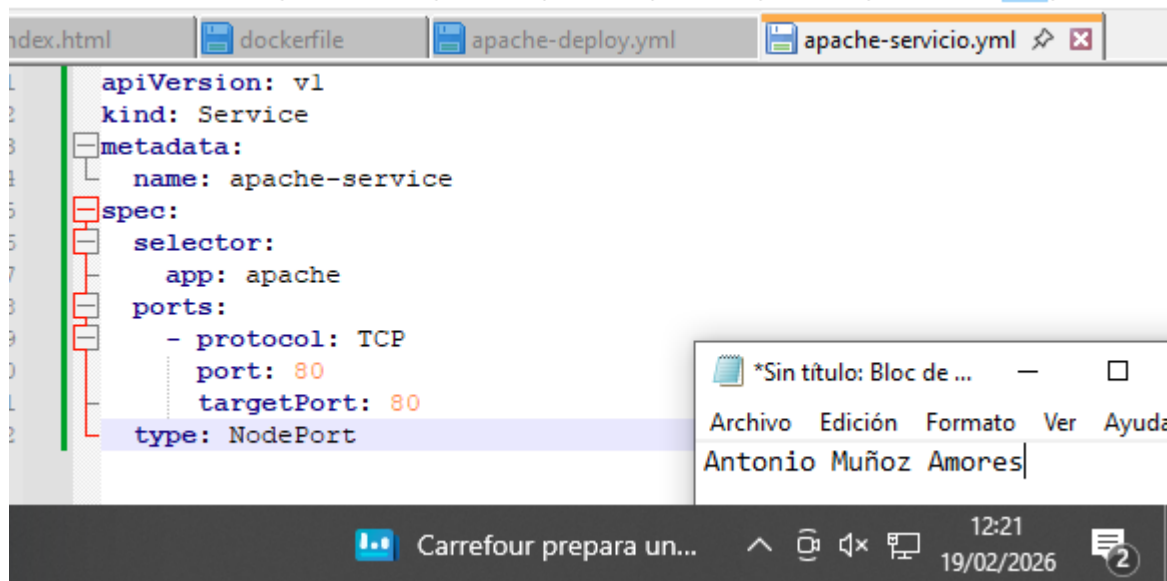
Creamos el siguiente archivo de despliegue

```
index.html | dockerfile | apache-deploy.yml
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`

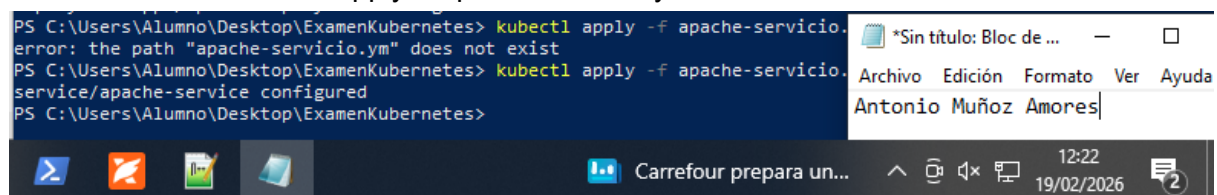
```
PS C:\Users\Alumno\Desktop\ExamenKubernetes> kubectl apply -f apache-deploy.yml
deployment.apps/apache-deployment configured
PS C:\Users\Alumno\Desktop\ExamenKubernetes>
```

Creamos el siguiente archivo para el servicio de apache



```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: apache-service
5 spec:
6   selector:
7     app: apache
8   ports:
9     - protocol: TCP
10       port: 80
11       targetPort: 80
12   type: NodePort
```

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

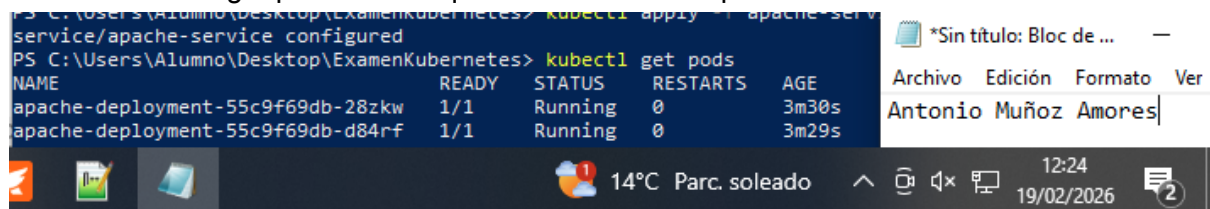


```
PS C:\Users\Alumno\Desktop\ExamenKubernetes> kubectl apply -f apache-servicio.
error: the path "apache-servicio.ym" does not exist
PS C:\Users\Alumno\Desktop\ExamenKubernetes> kubectl apply -f apache-servicio.
service/apache-service configured
PS C:\Users\Alumno\Desktop\ExamenKubernetes>
```

3. Comprobación del funcionamiento:

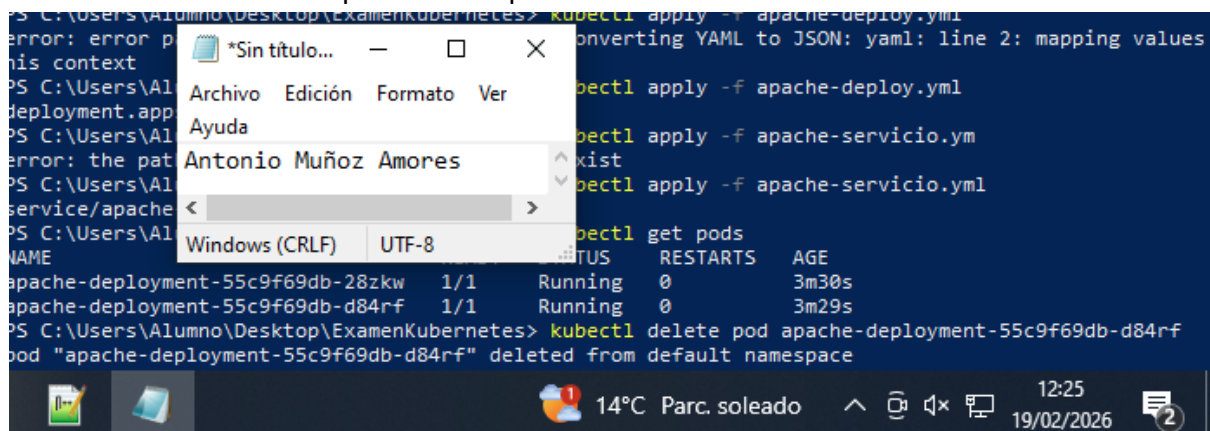
Con el comando get pods vemos que tenemos los dos pods creados anteriormente

```
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

```
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

```
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
```

