

DOCKER

1. INSTALACIÓN

a. <https://docs.docker.com/engine/install/>

2. PROBLEMAS CON DOCKER

a. <https://www.youtube.com/watch?v=R5FoxpxVukI>

b. Powershell (ejecutado como admin): `wsl --unregister docker-desktop`

3. COMPROBACIÓN DOCKER

a. Docker Desktop - Engine Started (verde)

b. docker -v

c. Explicar qué hace este comando

Comando para ver la versión de Docker instalada, este comando es útil para confirmar que se ha instalado correctamente y saber la versión que estamos usando.

d. docker run hello-world

e. Explicar qué hace este comando

Es una prueba básica de que Docker está funcionando correctamente.

```
PS C:\Windows\system32> docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
ea52d2000f90: Download complete
Digest: sha256:f7931603f70e13dbd844253370742c4fc4202d290c80442b2e68706d8f33ce26
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

4. TERMINAL INTERACTIVA DE UBUNTU

a. docker run -it ubuntu

b. Explicar qué hace este comando

Se abre una terminal dentro del contenedor Ubuntu. Ahora se pueden ejecutar comandos Linux

c. exit (para el contenedor)

Sale de la terminal de Ubuntu y nos devuelve a la terminal de PowerShell

5. OPERAR CON CONTENEDORES

a. docker run -it --name tu_nombre ubuntu

b. CTRL + P + Q => vuelve a la máquina principal

c. docker attach tu_nombre

d. Explicar qué hace este comando

Cuando creamos el contenedor con nuestro nombre, al usar CTRL + P + Q, nos devuelve a la terminal principal (PowerShell), dejando el contenedor ejecutándose en segundo plano. Con el comando `docker attach "nombre"` recupera la terminal de Ubuntu.

e. docker ps

f. Explicar qué hace este comando

Muestra todos los contenedores que están corriendo

g. docker rm -f tu_nombre

h. Explicar qué hace este comando

Elimina el contenedor por nombre

```
PS C:\Windows\system32> docker run it --name javi ubuntu
Unable to find image 'it:latest' locally
docker: Error response from daemon: pull access denied for it, repository does not exist or may require 'docker login'

Run 'docker run --help' for more information
PS C:\Windows\system32> docker run -it --name javi ubuntu
root@d5b9a41967f3:/#
PS C:\Windows\system32> docker attach javi
root@d5b9a41967f3:/# docker ps
bash: docker: command not found
root@d5b9a41967f3:/# read escape sequence
PS C:\Windows\system32> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS   NAMES
d5b9a41967f3   ubuntu   "/bin/bash"             7 minutes ago Up 7 minutes   javi
PS C:\Windows\system32> docker rm -f javi
javi
PS C:\Windows\system32> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS   NAMES
PS C:\Windows\system32>
```

6. CONTENEDOR CON PÁGINA WEB

a. Clonar <https://github.com/franguerrerosanchez/nginx-web-docker.git>

b. Explicar qué contiene este proyecto

Contiene un sitio web preparado para usarlo con NGINX dentro de un contenedor de Docker.

c. Explicar el contenido del fichero Dockerfile

Este fichero crea una imagen Docker que contiene una página web y la sirve con NGINX

d. docker build -t nginx-web-docker .

e. Explicar qué hace este comando

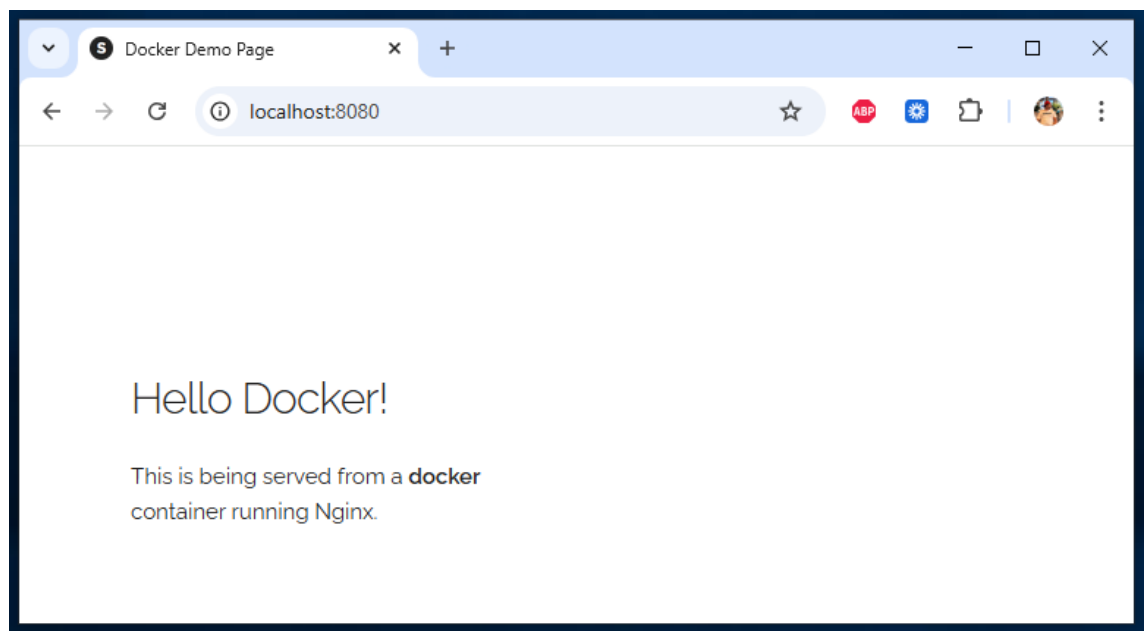
Ha buscado el archivo Dockerfile en la carpeta clonada, ha descargado la imagen, copiado el proyecto dentro de la imagen y ha creado una imagen Docker nueva.

f. docker run -d -p 80:80 --name mi_web nginx-web-docker

g. Explicar qué hace este comando

Ejecuta en segundo plano y expone el puerto 80 del contenedor en nuestro 8080 local

h. Navegar a <https://localhost:80>

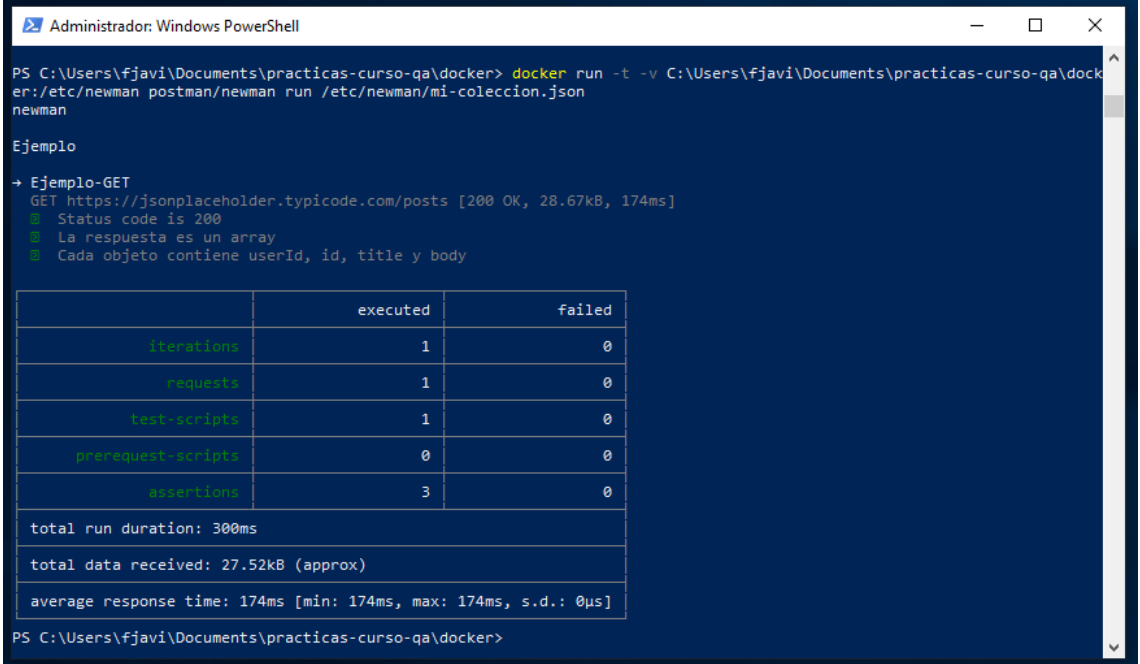


7. CONTENEDOR CON NEWMAN PARA EJECUTAR PRUEBAS DE POSTMAN

a. docker run -t postman/newman run <link de tu coleccion de postman>

b. Explicar qué hace este comando

Arranca un contenedor con Newman y descarga mi colección de Postman a través del archivo proporcionado (lo he hecho con el archivo guardado en mi local y no con el link de Postman). Ejecuta los requests y muestra los resultados en la terminal.



```
PS C:\Users\fjavi\Documents\practicas-curso-qa\docker> docker run -t -v C:\Users\fjavi\Documents\practicas-curso-qa\docker:/etc/newman postman/newman run /etc/newman/mi-coleccion.json
newman

Ejemplo

→ Ejemplo-GET
GET https://jsonplaceholder.typicode.com/posts [200 OK, 28.67kB, 174ms]
  Status code is 200
  La respuesta es un array
  Cada objeto contiene userId, id, title y body
```

	executed	failed
iterations	1	0
requests	1	0
test-scripts	1	0
prerequisite-scripts	0	0
assertions	3	0
total run duration: 300ms		
total data received: 27.52kB (approx)		
average response time: 174ms [min: 174ms, max: 174ms, s.d.: 0µs]		

```
PS C:\Users\fjavi\Documents\practicas-curso-qa\docker>
```

8. CREAR CONTENEDOR PARA TEST DE CYPRESS PARA LA WEB

a. Clonar <https://github.com/franguerrerosanchez/cypress-web-docker.git>

b. Explicar qué tiene este proyecto

Contiene un proyecto de cypress

c. Explicar qué hace el test

Abre una página web y comprueba que el título sea exactamente igual a “Docker Demo Page”

d. docker run --network=host -it -v \$PWD:/e2e -w /e2e cypress/included:12.5.1

./node_modules/.bin/cypress run --spec "cypress/e2e/spec.cy.js"

e. Explicar qué hace este comando

El comando hace todo el flujo de pruebas de Cypress dentro de Docker.

```
Administrador: Windows PowerShell Francisco Javier Jimenez (javi.jim7@hotmail.com)

    at Promise._settlePromise0 (http://localhost:34659/__cypress/runner/cypress_runner.js:6951:10)
    at Promise._settlePromises (http://localhost:34659/__cypress/runner/cypress_runner.js:7027:18)
    at _drainQueueStep (http://localhost:34659/__cypress/runner/cypress_runner.js:3621:12)
    at _drainQueue (http://localhost:34659/__cypress/runner/cypress_runner.js:3614:9)
    at ../../node_modules/bluebird/js/release/async.js.Async._drainQueues (http://localhost:34659/__cypress/runner/cyp
ress_runner.js:3630:5)
    at Async.drainQueues (http://localhost:34659/__cypress/runner/cypress_runner.js:3500:14)
From Your Spec Code:
    at Context.eval (webpack:///./cypress/e2e/spec.cy.js:3:7)

From Node.js Internals:
  Error: connect ECONNREFUSED 127.0.0.1:80
    at TCPConnectWrap.afterConnect [as oncomplete] (node:net:1187:16)

(Results)

Tests:      1
Passing:    0
Failing:    1
Pending:    0
Skipped:    0
Screenshots: 1
Video:      true
Duration:   0 seconds
Spec Ran:   spec.cy.js

(Screenshots)

- /e2e/cypress/screenshots/spec.cy.js/Web Server -- should have the correct title (1280x720)
  (failed).png

(Video)

- Started processing: Compressing to 32 CRF
- Finished processing: 0 seconds

- Video output: /e2e/cypress/videos/spec.cy.js.mp4

=====

(Run Finished)

Spec                                Tests  Passing  Failing  Pending  Skipped
  0 spec.cy.js                      960ms    1        -        1        -        -
  0 1 of 1 failed (100%)            960ms    1        -        1        -        -

PS C:\Users\fjavi\Documents\practicass-curso-qa\docker>
```

9. DOCKER-COMPOSE

a. Clonar <https://github.com/cypress-io/cypress-example-docker-compose>

b. Explicar qué tiene este proyecto

Es un ejemplo de como usar Docker y Docker-compose para pruebas e2e con Cypress

c. docker-compose build

d. Explicar qué hace este comando

Prepara todas las imágenes Docker del proyecto sin iniciar los contenedores. Construye imágenes de forma automática y reproducible y permite poder levantar después los contenedores con el siguiente comando

e. docker-compose up

f. Explicar qué hace este comando

Crea la red interna del proyecto. Levanta los contenedores de la app y los tests. Ejecuta Cypress y muestra los resultados en la consola.

```
Administrador: Windows PowerShell
PS C:\Users\fjavi\Documents\practicass-curso-qa\docker\cypress-example-docker-compose> docker-compose up
time="2025-12-10T12:17:11+01:00" level=warning msg="C:\Users\fjavi\Documents\practicass-curso-qa\docker\cypress-example-docker-compose\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"

Container apache Recreated
Attaching to apache, cypress
apache | [Wed Dec 10 11:17:11.844835 2025] [mpm_event:notice] [pid 1:tid 1] AH00489: Apache/2.4.66 (Unix) configure
d -- resuming normal operations
apache | [Wed Dec 10 11:17:11.845151 2025] [core:notice] [pid 1:tid 1] AH00094: Command line: 'httpd -D FOREGROUND'

cypress | [31:1210/111713.915760:ERROR:object_proxy.cc(590)] Failed to call method: org.freedesktop.portal.Settings
.Read: object_path= /org/freedesktop/portal/desktop: unknown error type:

cypress | DevTools listening on ws://127.0.0.1:34239/devtools/browser/13446d8d-3e98-4c92-88ab-8fcf65897af3
cypress | Couldn't determine Mocha version
cypress |
cypress | =====
cypress | (Run Starting)
cypress |
cypress | Cypress:      13.6.4
cypress | Browser:      Electron 114 (headless)
cypress | Node Version: v16.14.2 (/usr/local/bin/node)
cypress | Specs:         1 found (spec.cy.js)
cypress | Searched:     cypress/e2e/**/*.cy.{js,jsx,ts,tsx}
cypress |
cypress | =====
cypress | Running: spec.cy.js (1 of 1)
cypress | 172.18.0.3 - - [10/Dec/2025:11:17:19 +0000] "GET / HTTP/1.1" 200 30
cypress | (Results)
cypress |
cypress | Tests:        1
cypress | Passing:      1
cypress | Failing:       0
cypress | Pending:       0
cypress | Skipped:       0
cypress | Screenshots:  0
cypress | Video:        false
cypress | Duration:     0 seconds
cypress | Spec Ran:     spec.cy.js
cypress |
cypress | =====
cypress | (Run Finished)
cypress |
cypress | Spec                                Tests  Passing  Failing  Pending  Skipped
cypress | ---                                -
cypress | @ spec.cy.js                        167ms   1         1       -        -        -
cypress | @ All specs passed!                  167ms   1         1       -        -        -
cypress |
cypress | cypress exited with code 0

View in Docker Desktop View Config Enable Watch
```