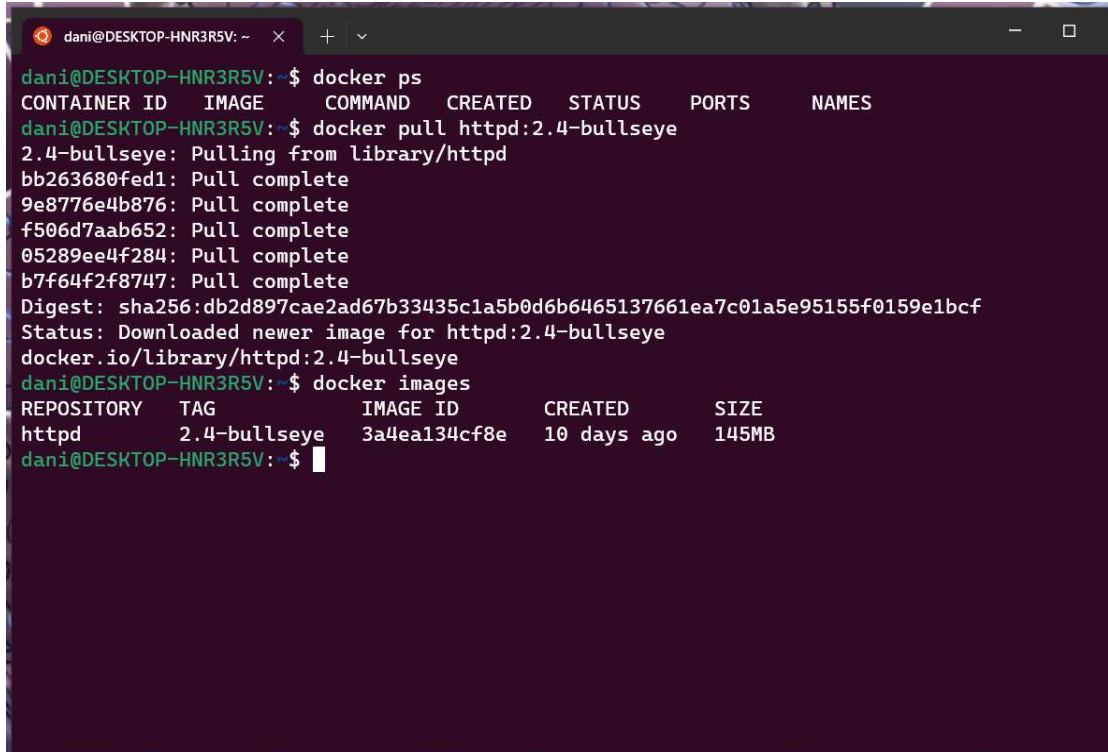


1. Docker. Imágenes

1.Descarga del registro de imágenes por defecto el servidor web apache httpd en su versión 2.4 bullseye. Toma nota del tamaño que ocupa la imagen en disco.

A terminal window with a dark background and light green text. The window title is 'dani@DESKTOP-HNR3R5V: ~'. The user enters 'docker ps' and gets a table with 6 columns: CONTAINER ID, IMAGE, COMMAND, CREATED, STATUS, PORTS, and NAMES. Then the user enters 'docker pull httpd:2.4-bullseye'. The output shows the image being pulled from the library, with several layers being pulled and completed. It then shows the digest (sha256) and status (Downloaded newer image). Finally, the user enters 'docker images' and gets a table with 5 columns: REPOSITORY, TAG, IMAGE ID, CREATED, and SIZE. The table shows 'httpd' with tag '2.4-bullseye', image ID '3a4ea134cf8e', created '10 days ago', and size '145MB'.

```
dani@DESKTOP-HNR3R5V: ~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
dani@DESKTOP-HNR3R5V: ~$ docker pull httpd:2.4-bullseye
2.4-bullseye: Pulling from library/httpd
bb263680fed1: Pull complete
9e8776e4b876: Pull complete
f506d7aab652: Pull complete
05289ee4f284: Pull complete
b7f64f2f8747: Pull complete
Digest: sha256:db2d897cae2ad67b33435c1a5b0d6b6465137661ea7c01a5e95155f0159e1bcf
Status: Downloaded newer image for httpd:2.4-bullseye
docker.io/library/httpd:2.4-bullseye
dani@DESKTOP-HNR3R5V: ~$ docker images
REPOSITORY    TAG       IMAGE ID      CREATED       SIZE
httpd         2.4-bullseye  3a4ea134cf8e  10 days ago  145MB
dani@DESKTOP-HNR3R5V: ~$
```

Ocupa 145 Megabytes

2.¿Cuál es el ID de la imagen que acabas de descargar?¿Y su hash o digest?¿Qué relación tienen entre sí?

El id es ID:3a4ea134cf8e.

```
dani@DESKTOP-HNR3R5V: ~  
dani@DESKTOP-HNR3R5V:~$ docker inspect 3a4ea134cf8e  
[  
  {  
    "Id": "sha256:3a4ea134cf8e081516a776ce184dedc28986f941ed214b9012dc888049480f5a",  
    "RepoTags": [  
      "httpd:2.4-bullseye"  
    ],  
    "RepoDigests": [  
      "httpd@sha256:db2d897cae2ad67b33435c1a5b0d6b6465137661ea7c01a5e95155f0159e1bcf"  
    ],  
    "Parent": "",  
    "Comment": "",  
    "Created": "2023-02-09T04:28:52.362739699Z",  
    "Container": "a921b4bf2b2e520ae53d7202663430537a77b92babaca7279a51207e97a870cd",  
    "ContainerConfig": {  
      "Hostname": "a921b4bf2b2e",  
      "Domainname": "",  
      "User": "",  
      "AttachStdin": false,  
      "AttachStdout": false,  
      "AttachStderr": false,  
      "ExposedPorts": {  
        "80/tcp": {}  
      },  
      "Tty": false,  
    },  
  },  
]
```

El hash está en "RepoDigests" y se encuentra completo después de @.

3.Descarga ahora la versión 3.17 alpine de la imagen ¿Cuánto ocupa en disco?¿Cuántas veces más pequeña o grande es la imagen de alpine respecto de la de bullseye? ¿A qué se debe esta diferencia?

```
Windows PowerShell dani@DESKTOP-HNR3R5V: ~  
dani@DESKTOP-HNR3R5V:~$ docker pull alpine:3.17  
3.17: Pulling from library/alpine  
63b65145d645: Pull complete  
Digest: sha256:69665d02cb32192e52e07644d76bc6f25abeb5410edc1c7a81a10ba3f0efb90a  
Status: Downloaded newer image for alpine:3.17  
docker.io/library/alpine:3.17  
dani@DESKTOP-HNR3R5V:~$ docker images  
REPOSITORY TAG IMAGE ID CREATED SIZE  
alpine 3.17 b2aa39c304c2 10 days ago 7.05MB  
httpd 2.4-bullseye 3a4ea134cf8e 12 days ago 145MB  
dani@DESKTOP-HNR3R5V:~$
```

Ocupa 7.05MB

145/7.05=20,56737588652482

Es 20,5 veces mas grande la version 2.4-Bullseye que la de alpine.

La diferencia se debe a que la version alpine es mas sencilla y contiene menos funcionalidades.

4. Elimina ambas imágenes del registro local

```
dani@DESKTOP-HNR3R5V: ~$ docker pull alpine:3.17
3.17: Pulling from library/alpine
63b65145d645: Pull complete
Digest: sha256:69665d02cb32192e52e07644d76bc6f25abeb5410edc1c7a81a10ba3f0efb90a
Status: Downloaded newer image for alpine:3.17
docker.io/library/alpine:3.17
dani@DESKTOP-HNR3R5V: ~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
alpine        3.17      b2aa39c304c2   10 days ago   7.05MB
httpd        2.4-bullseye 3a4ea134cf8e   12 days ago   145MB
dani@DESKTOP-HNR3R5V: ~$ docker rmi b2aa39c304c2 3a4ea134cf8e
Untagged: alpine:3.17
Untagged: alpine@sha256:69665d02cb32192e52e07644d76bc6f25abeb5410edc1c7a81a10ba3f0efb90a
Deleted: sha256:b2aa39c304c27b96c1fef0c06bee651ac9241d49c4fe34381cab8453f9a89c7d
Deleted: sha256:7cd52847ad775a5ddc4b58326cf884beee34544296402c6292ed76474c686d39
Untagged: httpd:2.4-bullseye
Untagged: httpd@sha256:db2d897cae2ad67b33435c1a5b0d6b6465137661ea7c01a5e95155f0159e1bcf
Deleted: sha256:3a4ea134cf8e081516a776ce184dedc28986f941ed214b9012dc888049480f5a
Deleted: sha256:019e5c44c73d76bc67f1618d02f9535348180094293dc4ddcfe70894209fd9ed
Deleted: sha256:eb991c200c9af34ef15003013e10c8ce8e143991de9780f2d0c5370041f3cf19
Deleted: sha256:127d0bec4c754ad1d28fcb982b114444cbc9aca95a6f5a7d74560e61a109a2fb
Deleted: sha256:c5c018d684454c7d5056c7f72a970aac612a515e70f3c858cba3978039a26248
Deleted: sha256:4695cdfb426a05673a100e69d2fe9810d9ab2b3dd88ead97c6a3627246d83815
dani@DESKTOP-HNR3R5V: ~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
dani@DESKTOP-HNR3R5V: ~$
```

2. Docker. Containers y redes (3.25 pts)


1. Descarga la última versión de la imagen containous/whoami.

```
dani@DESKTOP-HNR3R5V: ~$ docker pull containous/whoami:latest
latest: Pulling from containous/whoami
29015087d73b: Pull complete
0109a00d13bc: Pull complete
d3caffff64d8: Pull complete
Digest: sha256:7d6a3c8f91470a23ef380320609ee6e69ac68d20bc804f3a1c6065fb56cfa34e
Status: Downloaded newer image for containous/whoami:latest
docker.io/containous/whoami:latest
dani@DESKTOP-HNR3R5V: ~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
containous/whoami latest  0f6fbbdd377   2 years ago    7.37MB
dani@DESKTOP-HNR3R5V: ~$
```

2. Levanta un contenedor llamado testserver1 a partir de la imagen anterior asignando un puerto aleatorio del sistema host (opción -P). Ejecutalo en segundo plano.

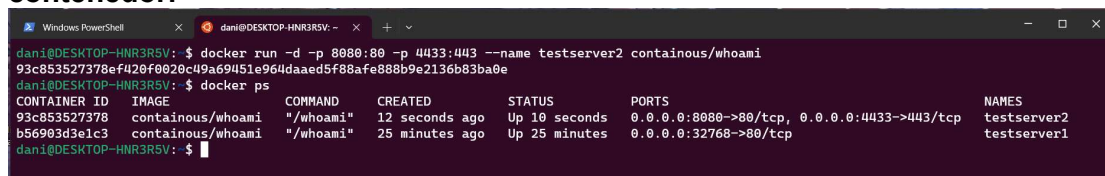
```
dani@DESKTOP-HNR3R5V: ~$ docker run -d -P --name testserver1 containous/whoami
b56903d3e1c36c5fd19ac5f564067066fe9d41dea3fbf303dd2702121a51ad2c
dani@DESKTOP-HNR3R5V: ~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
b56903d3e1c3   containous/whoami  "/whoami"               57 seconds ago Up 56 seconds  0.0.0.0:32768->80/tcp    testserver1
dani@DESKTOP-HNR3R5V: ~$
```

3. Inspecciona el contenedor anterior para determinar el puerto del host que se ha asignado. Accede a la URL <http://localhost:PUERTO> desde tu navegador.



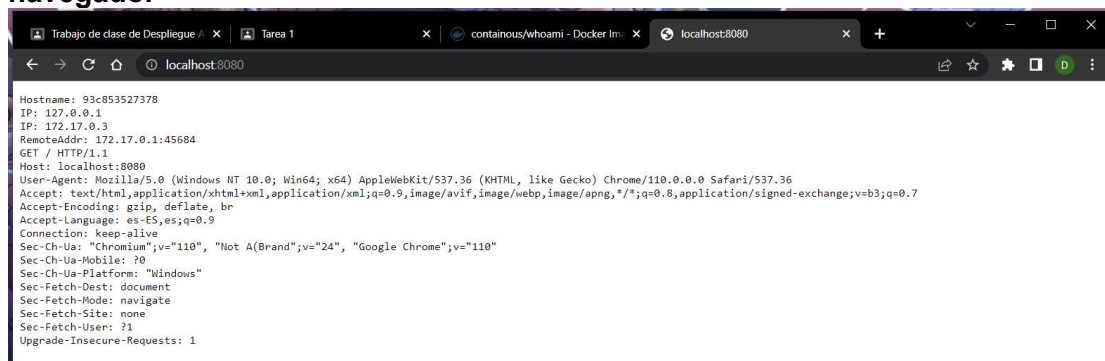
```
Trabajo de clase de Despliegue / x Tarea 1 x Docker x localhost:32768 x +
localhost:32768
GET / HTTP/1.1
Host: localhost:32768
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: es-ES,es;q=0.9
Connection: keep-alive
Sec-Ch-Ua: "Chromium";v="110", "Not A(Brand";v="24", "Google Chrome";v="110"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
```

4. Crea un contenedor idéntico al anterior con el nombre testserver2, pero en esta ocasión enlaza los puertos 8080 y 4433 del host a los puertos 80 y 443 del contenedor.

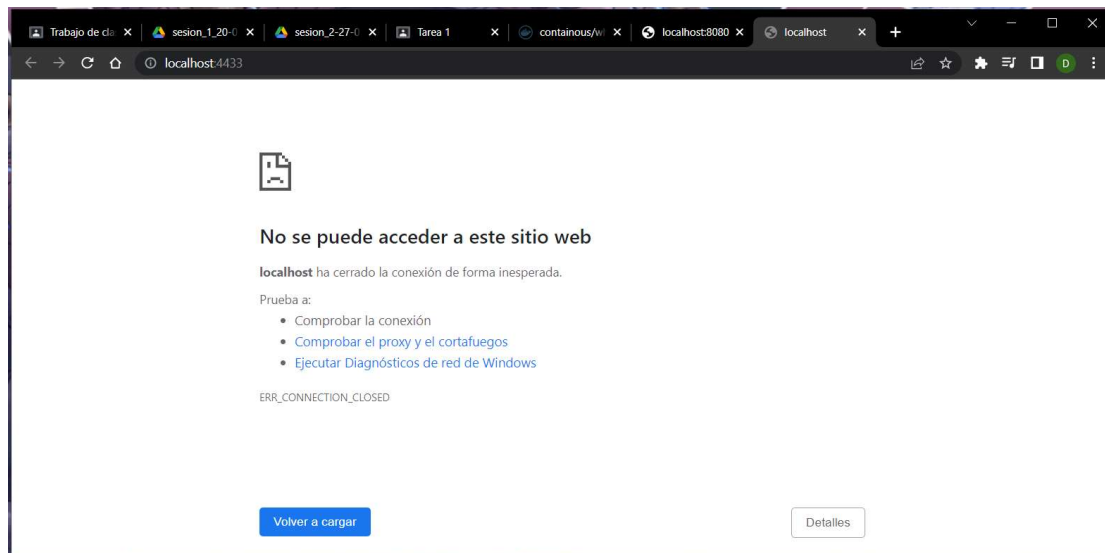


```
Windows PowerShell
dani@DESKTOP-HNR3R5V: ~
dani@DESKTOP-HNR3R5V: $ docker run -d -p 8080:80 -p 4433:443 --name testserver2 containous/whoami
93c853527378ef420f0020c49a69451e964daaed5f88afe888b9e2136b83ba0e
dani@DESKTOP-HNR3R5V: $ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                                                                 NAMES
93c853527378   containous/whoami   "/whoami"               12 seconds ago   Up 10 seconds   0.0.0.0:8080->80/tcp, 0.0.0.0:4433->443/tcp   testserver2
b56903d3e1c3   containous/whoami   "/whoami"               25 minutes ago   Up 25 minutes   0.0.0.0:32768->80/tcp                       testserver1
dani@DESKTOP-HNR3R5V: $
```

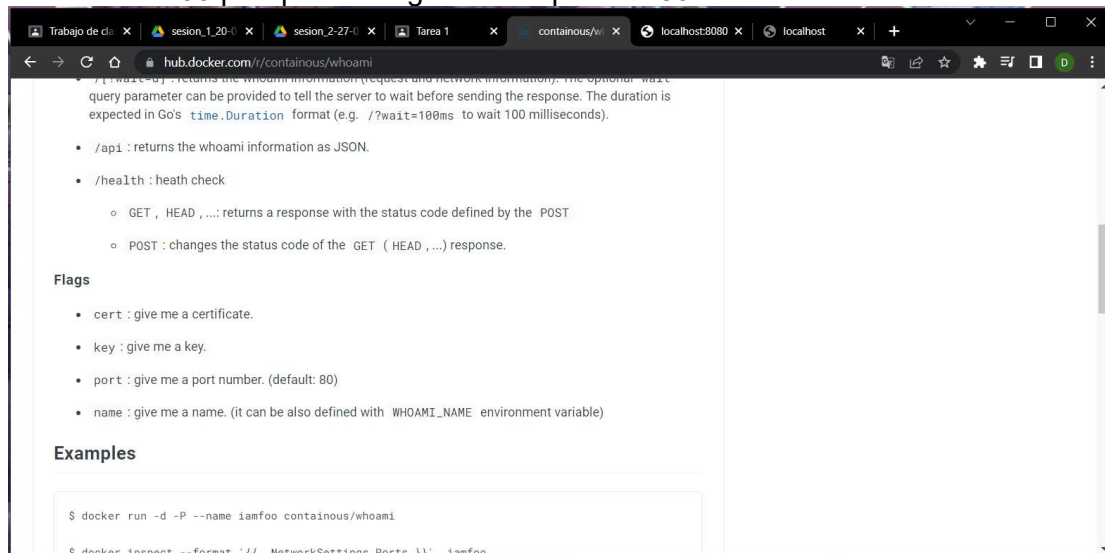
Accede a las URLS <http://localhost:8080> y <https://localhost:4433> desde tu navegador



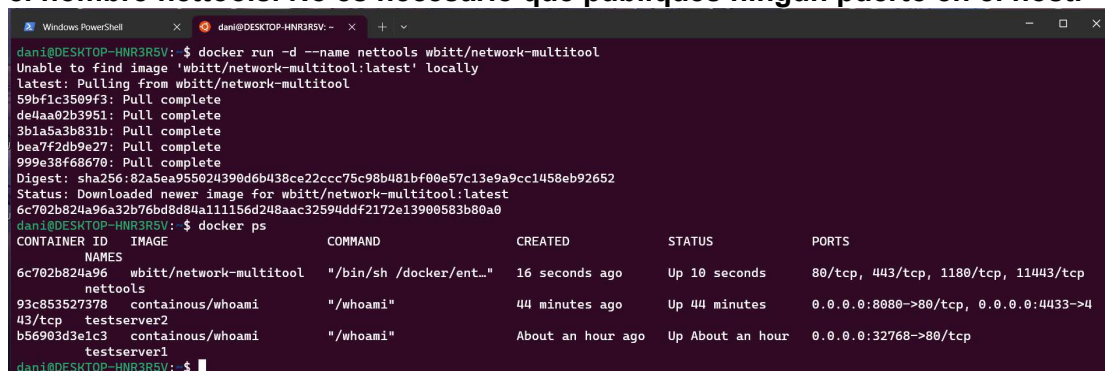
```
Trabajo de clase de Despliegue / x Tarea 1 x containous/whoami - Docker Im: x localhost:8080 x +
localhost:8080
GET / HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: es-ES,es;q=0.9
Connection: keep-alive
Sec-Ch-Ua: "Chromium";v="110", "Not A(Brand";v="24", "Google Chrome";v="110"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
```



No sale el 4433 por que la imagen usa el puersto 80



5. Levanta un contenedor a partir de la imagen `wbitt/network-multitool` con el nombre `nettools`. No es necesario que publiques ningún puerto en el host.



6. Utilizando el comando `docker exec -it nettools bash` accede al

contenedor nettools y ejecuta el comando ping testserver1 ¿Obtienes respuesta?

```
Windows PowerShell
dani@DESKTOP-HNR3R5V: ~
dani@DESKTOP-HNR3R5V: $ docker exec -it nettools bash
bash-5.1# ping testserver1
ping: testserver1: Try again
bash-5.1# ping testserver1
ping: testserver1: Try again
bash-5.1#
```

En caso negativo, ¿por qué no?

Los contenedores estan en una red por defecto que crea docker llamada bridge en la cual no se puede comunicar a traves del nombre. Los contenedores en esta red no se reconocen por su nombre

7.Repite el punto anterior pero en esta ocasión haz ping a la IP del contenedor testserver1 (tendrás que utilizar el comando docker inspect para conocer la IP) ¿Obtienes respuesta? ¿Por qué?

```
Windows PowerShell
dani@DESKTOP-HNR3R5V: ~
dani@DESKTOP-HNR3R5V: $ docker inspect testserver1
[
  {
    "Id": "b56903d3e1c36c5fd19ac5f564067066fe9d41dea3fbf303dd2702121a51ad2c",
    "Created": "2023-02-22T18:33:40.482548253Z",
    "Path": "/whoami",
    "Args": [],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 1722,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2023-02-22T18:33:41.11440119Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    }
  },
]
```

```
Windows PowerShell
dani@DESKTOP-HNR3R5V: ~
dani@DESKTOP-HNR3R5V: $ docker exec -it nettools bash
bash-5.1# ping 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=5.24 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.043 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.047 ms
64 bytes from 172.17.0.2: icmp_seq=4 ttl=64 time=0.045 ms
64 bytes from 172.17.0.2: icmp_seq=5 ttl=64 time=0.051 ms
64 bytes from 172.17.0.2: icmp_seq=6 ttl=64 time=0.054 ms
64 bytes from 172.17.0.2: icmp_seq=7 ttl=64 time=0.044 ms
64 bytes from 172.17.0.2: icmp_seq=8 ttl=64 time=0.049 ms
64 bytes from 172.17.0.2: icmp_seq=9 ttl=64 time=0.053 ms
64 bytes from 172.17.0.2: icmp_seq=10 ttl=64 time=0.045 ms
64 bytes from 172.17.0.2: icmp_seq=11 ttl=64 time=0.072 ms
64 bytes from 172.17.0.2: icmp_seq=12 ttl=64 time=0.049 ms
64 bytes from 172.17.0.2: icmp_seq=13 ttl=64 time=0.045 ms
64 bytes from 172.17.0.2: icmp_seq=14 ttl=64 time=0.055 ms
64 bytes from 172.17.0.2: icmp_seq=15 ttl=64 time=0.050 ms
64 bytes from 172.17.0.2: icmp_seq=16 ttl=64 time=0.057 ms
64 bytes from 172.17.0.2: icmp_seq=17 ttl=64 time=0.050 ms
```

La red tipo bridge si permite la comunicacion mediante ip

8.Crea una red llamada testnet y une los contenedores testserver1 y nettools a la misma. Repite el apartado 6. Explica qué ocurre en esta ocasión

```
Windows PowerShell
dani@DESKTOP-HNR3R5V: ~
dani@DESKTOP-HNR3R5V: $ docker network create testnet
9d4643b5cba7d3bb4880dfab1411edd4b92516af39541606a39230b980429f28
dani@DESKTOP-HNR3R5V: $ docker network ls
NETWORK ID          NAME       DRIVER  SCOPE
8fc2b0c01e8a        bridge    bridge  local
154b5cb7b948        host      host    local
0cbacd326ce3        none      null     local
9d4643b5cba7        testnet   bridge  local
dani@DESKTOP-HNR3R5V: $
```

```
Windows PowerShell
dani@DESKTOP-HNR3R5V: ~
dani@DESKTOP-HNR3R5V: $ docker network connect testnet testserver1
dani@DESKTOP-HNR3R5V: $ docker network connect testnet nettools
dani@DESKTOP-HNR3R5V: $ docker exec -it nettools bash
bash-5.1# ping testserver1
PING testserver1 (172.18.0.2) 56(84) bytes of data:
64 bytes from testserver1.testnet (172.18.0.2): icmp_seq=1 ttl=64 time=0.682 ms
64 bytes from testserver1.testnet (172.18.0.2): icmp_seq=2 ttl=64 time=0.119 ms
64 bytes from testserver1.testnet (172.18.0.2): icmp_seq=3 ttl=64 time=0.048 ms
64 bytes from testserver1.testnet (172.18.0.2): icmp_seq=4 ttl=64 time=0.049 ms
64 bytes from testserver1.testnet (172.18.0.2): icmp_seq=5 ttl=64 time=0.049 ms
64 bytes from testserver1.testnet (172.18.0.2): icmp_seq=6 ttl=64 time=0.049 ms
64 bytes from testserver1.testnet (172.18.0.2): icmp_seq=7 ttl=64 time=0.044 ms
64 bytes from testserver1.testnet (172.18.0.2): icmp_seq=8 ttl=64 time=0.046 ms
64 bytes from testserver1.testnet (172.18.0.2): icmp_seq=9 ttl=64 time=0.114 ms
64 bytes from testserver1.testnet (172.18.0.2): icmp_seq=10 ttl=64 time=0.059 ms
^C
--- testserver1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9345ms
rtt min/avg/max/mdev = 0.044/0.125/0.682/0.187 ms
bash-5.1#
```

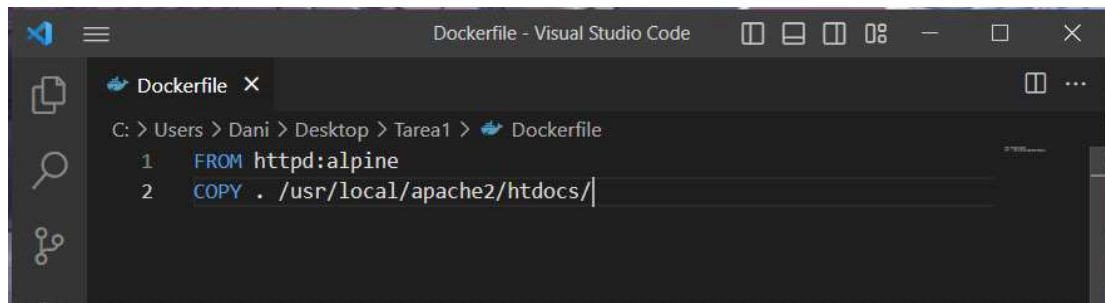
3. Docker. Dockerfile y volúmenes

1. Clona el repo de la asignatura <https://github.com/tcamba-ciclos-montecastelo/daw-22>. En la ruta tareas/tarea-1/ejercicio-5/dist/ encontrarás los estáticos de una página web.

```
MINGW64:/c/Users/Dani/Desktop
Dani@DESKTOP-HNR3R5V MINGW64 ~/Desktop
$ git clone https://github.com/tcamba-ciclos-montecastelo/daw-22.git
Cloning into 'daw-22'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 10 (delta 0), reused 10 (delta 0), pack-reused 0
Receiving objects: 100% (10/10), done.

Dani@DESKTOP-HNR3R5V MINGW64 ~/Desktop
$ |
```

2. Completa el Dockerfile que encontrarás en tareas/tarea-1/ejercicio-5/Dockerfile para crear una imagen basada en la última versión de alpine de httpd que sirva la web anterior



3. Construye la imagen a partir del Dockerfile anterior con el tag {nombre}/web5:latest

```

Windows PowerShell
PS C:\Users\Dani\Desktop\Tarea1> docker build -t "daniel/web5:latest" .
[+] Building 6.1s (8/8) FINISHED
=> [internal] load build definition from Dockerfile                                0.1s
=> => transferring dockerfile: 89B                                                0.0s
=> [internal] load .dockerignore                                                  0.1s
=> => transferring context: 2B                                                    0.0s
=> [internal] load metadata for docker.io/library/httpd:alpine                  2.1s
=> [auth] library/httpd:pull token for registry-1.docker.io                    0.0s
=> [internal] load build context                                                  0.1s
=> => transferring context: 6.30MB                                                0.1s
=> [1/2] FROM docker.io/library/httpd:alpine@sha256:99e7d53182c2edefc8050e93df0c3f9cbb279531a6ce4f5341ebd3988778 3.1s
=> resolve docker.io/library/httpd:alpine@sha256:99e7d53182c2edefc8050e93df0c3f9cbb279531a6ce4f5341ebd3988778 0.0s
=> sha256:2a7233f37399d2b38e180ff4511d403623006bd630acdd3ef8a2277737c0f881 1.26kB / 1.26kB 0.2s
=> sha256:3a49cb9b71696ca3e3d3725dab652a4d5fb940aafadee799cf381592eb238ad4 178B / 178B 0.3s
=> sha256:99e7d53182c2edefc8050e93df0c3f9cbb279531a6ce4f5341ebd39887789c4b 1.65kB / 1.65kB 0.0s
=> sha256:9abb46b5c8db76cc89a94fd4b2ddc4eb4a1ac3edc871f8e39d7e634a12047254 1.57kB / 1.57kB 0.0s
=> sha256:2717589de96486e8f3277f070b2299e2f9841601008ab0474224489440a1c98f 8.74kB / 8.74kB 0.0s
=> sha256:56e6e368b425382c3fc100b05c371fd91cbddc9f4dc53f72b5e971ad11311130 9.87MB / 9.87MB 1.1s

=> => extracting sha256:63b65145d645c1250c391b2d16be53b3747c295ca8ba2fcb6b0cf064a4dc21c 0.3s
=> sha256:be96bdb2a24d7b8d8145289df7413e020083cef5e2817f7e92a2ced2d84565c0 290B / 290B 1.0s
=> extracting sha256:2a7233f37399d2b38e180ff4511d403623006bd630acdd3ef8a2277737c0f881 0.0s
=> extracting sha256:3a49cb9b71696ca3e3d3725dab652a4d5fb940aafadee799cf381592eb238ad4 0.0s
=> extracting sha256:56e6e368b425382c3fc100b05c371fd91cbddc9f4dc53f72b5e971ad11311130 0.7s
=> extracting sha256:7f67bdc40aa69b594ed41bc4c1428934acab1d44ad89d639b84b48400d74e707 0.3s
=> extracting sha256:be96bdb2a24d7b8d8145289df7413e020083cef5e2817f7e92a2ced2d84565c0 0.0s
=> [2/2] COPY . /usr/local/apache2/htdocs/                                       0.5s
=> exporting image                                                                0.1s
=> exporting layers                                                              0.1s
=> writing image sha256:ea8d2956f4f1915e33cedc0a504c830f16bf82c52e115167a4f0ebac5791e1d8 0.0s
=> naming to docker.io/daniel/web5:latest                                       0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
PS C:\Users\Dani\Desktop\Tarea1> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
daniel/web5   latest    ea8d2956f4f1   22 seconds ago 63.2MB
PS C:\Users\Dani\Desktop\Tarea1>

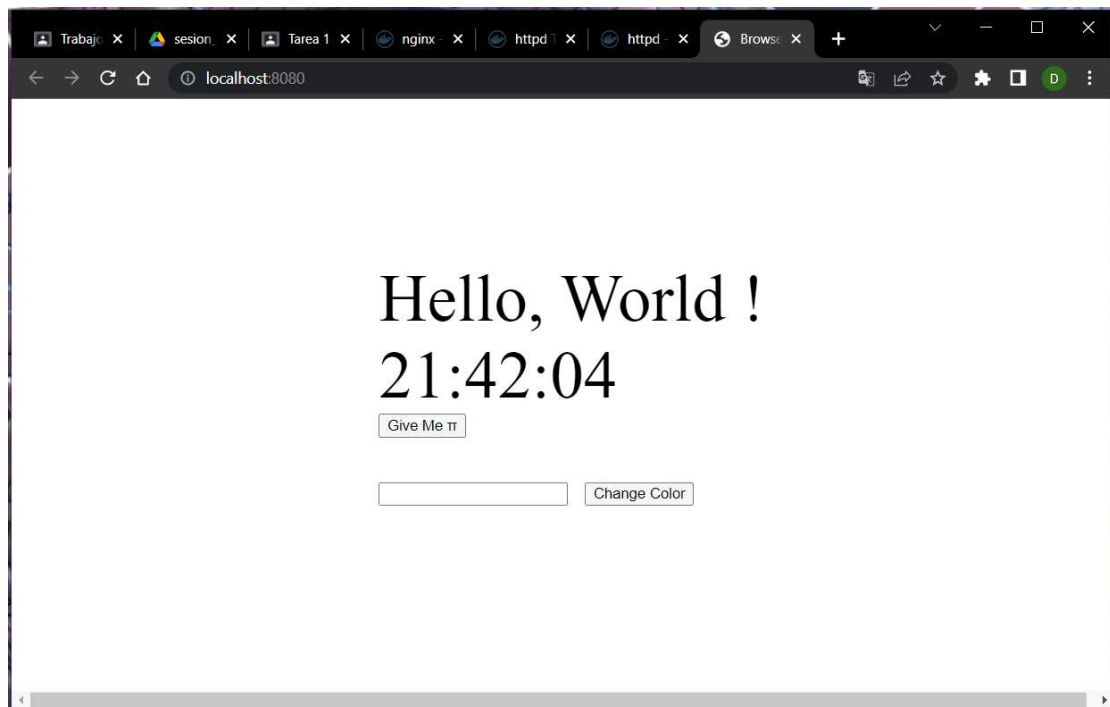
```

4. Levanta un contenedor de la imagen anterior llamado web5test que publique el puerto 80 en el 8080 del host. ¿Puedes acceder a la web en <http://localhost:8080> ? En caso negativo ¿Qué está sucediendo? ¿Cómo lo arreglarías?

```

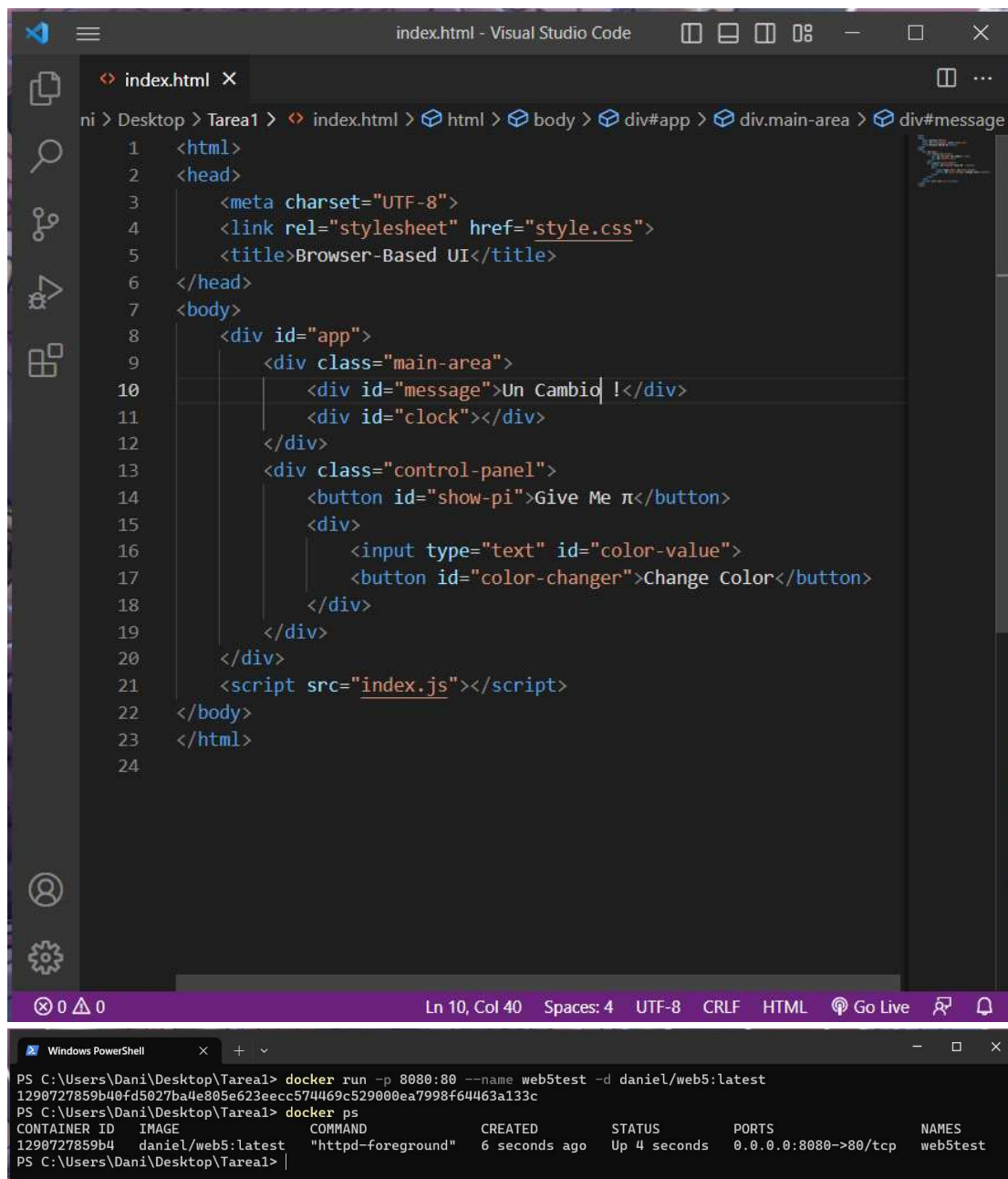
Windows PowerShell
PS C:\Users\Dani\Desktop\Tarea1> docker run -p 8080:80 --name web5test -d daniel/web5:latest
25d97f030c9a03170bf9edd7904a65514b6ae44bffe74f24b7ffdaee1cec8f0
PS C:\Users\Dani\Desktop\Tarea1> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
25d97f030c9a   daniel/web5:latest "httpd-foreground"      6 seconds ago Up 5 seconds   0.0.0.0:8080->80/tcp    web5test
PS C:\Users\Dani\Desktop\Tarea1>

```

5. Detén el contenedor y eliminalo. Haz un cambio cualquiera en el fichero html de la web y vuelve a levantarlo (sin reconstruir la imagen). ¿Se aplicaron los cambios en la web? En caso negativo ¿qué ha sucedido?

```
Windows PowerShell
PS C:\Users\Dani\Desktop\Tarea1> docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORTS    NAMES
995bf48eee56   daniel/web5:latest   "httpd-foreground"      2 minutes ago    Up 2 minutes    0.0.0.0:8080->80/tcp    web5test
PS C:\Users\Dani\Desktop\Tarea1> docker stop web5test
web5test
PS C:\Users\Dani\Desktop\Tarea1> docker rm web5test
web5test
PS C:\Users\Dani\Desktop\Tarea1> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORTS    NAMES
PS C:\Users\Dani\Desktop\Tarea1>
```



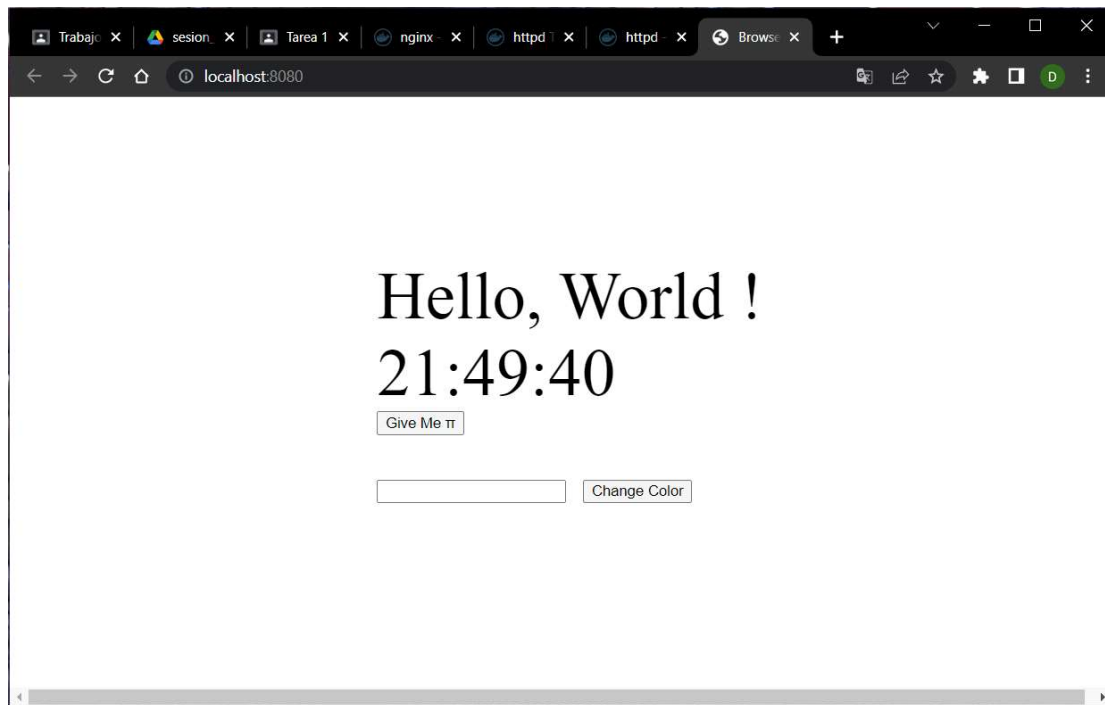
The image shows a Visual Studio Code editor window with a file named `index.html` open. The file path is `ni > Desktop > Tarea1 > index.html`. The code is an HTML document with the following structure:

```
1 <html>
2 <head>
3   <meta charset="UTF-8">
4   <link rel="stylesheet" href="style.css">
5   <title>Browser-Based UI</title>
6 </head>
7 <body>
8   <div id="app">
9     <div class="main-area">
10      <div id="message">Un Cambio !</div>
11      <div id="clock"></div>
12    </div>
13    <div class="control-panel">
14      <button id="show-pi">Give Me  $\pi$ </button>
15      <div>
16        <input type="text" id="color-value">
17        <button id="color-changer">Change Color</button>
18      </div>
19    </div>
20  </div>
21  <script src="index.js"></script>
22 </body>
23 </html>
24
```

The status bar at the bottom of the editor shows: `Ln 10, Col 40 Spaces: 4 UTF-8 CRLF HTML Go Live`.

Below the editor is a Windows PowerShell terminal window. It shows the following commands and output:

```
PS C:\Users\Dani\Desktop\Tarea1> docker run -p 8080:80 --name web5test -d daniel/web5:latest
1290727859b40fd5027ba4e805e623eccc574469c529000ea7998f64463a133c
PS C:\Users\Dani\Desktop\Tarea1> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
1290727859b4   daniel/web5:latest "httpd-foreground"      6 seconds ago Up 4 seconds   0.0.0.0:8080->80/tcp    web5test
PS C:\Users\Dani\Desktop\Tarea1>
```



No se aplicaron los cambios, por que son los archivos que carga al hacer docker build. Para aplicar cambios habria que hacer un docker build con los nuevos cambios para que queden guardados en esa imagen.

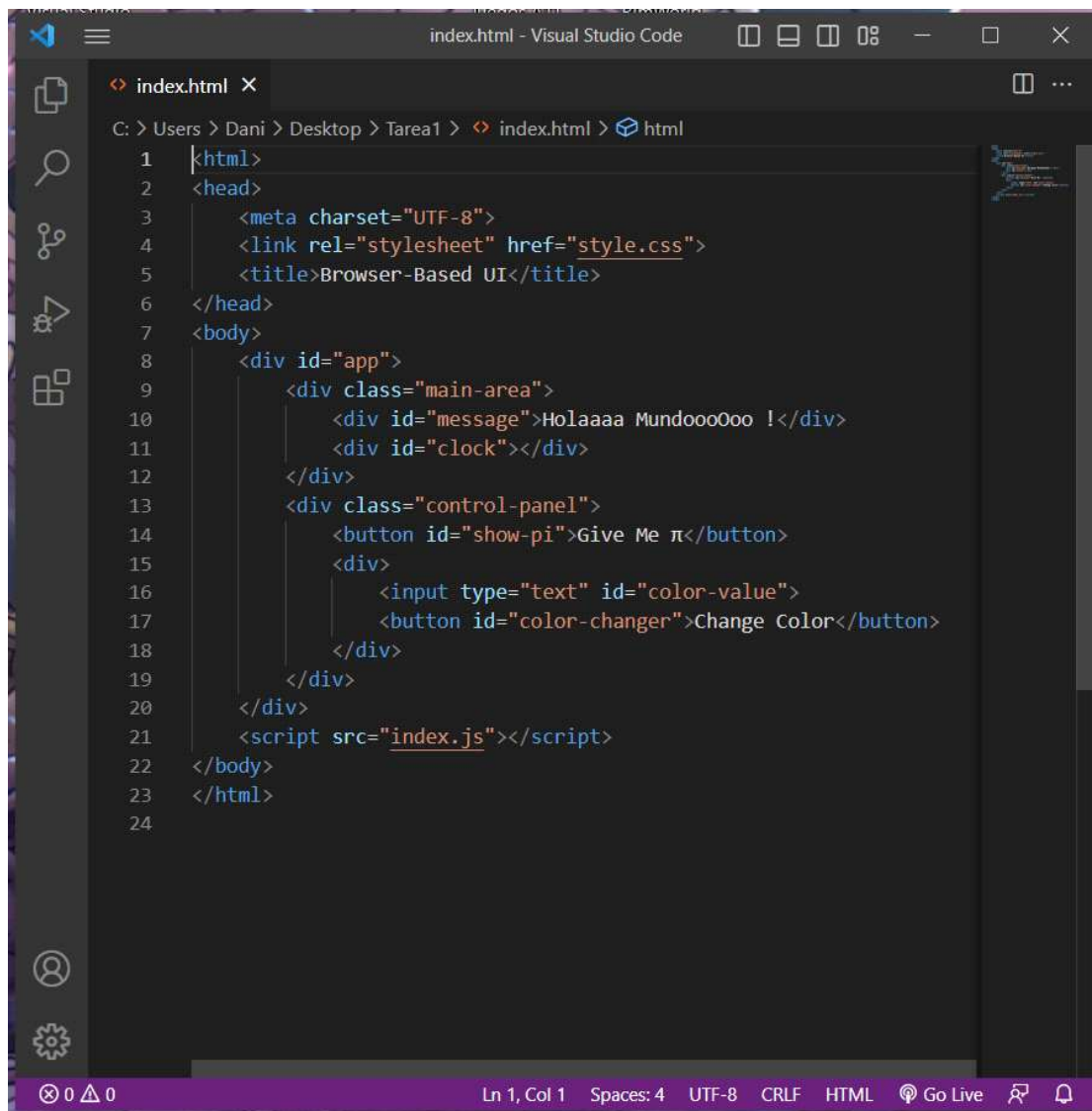
6. A continuación levanta un nuevo contenedor con el mismo nombre (web5test) pero en esta ocasión enlaza o mapea la ruta del host que contiene los estáticos de la web con el directorio del contenedor desde el que se sirven. Este paso lo puedes realizar directamente por consola utilizando el comando docker run con la imagen oficial de httpd (es decir, puedes prescindir de la imagen custom construida a partir del Dockerfile). Comprueba que puedes acceder al sitio web que acabas de servir en localhost

```

Windows PowerShell
PS C:\Users\Dani\Desktop\Tarea1> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS               NAMES
1290727859b4   daniel/web5:latest "httpd-foreground"      3 minutes ago Up 3 minutes   0.0.0.0:8080->80/tcp web5test
PS C:\Users\Dani\Desktop\Tarea1> docker stop web5test
web5test
PS C:\Users\Dani\Desktop\Tarea1> docker rm web5test
web5test
PS C:\Users\Dani\Desktop\Tarea1> docker run -p 8080:80 --name web5test -d -v C:/Users/Dani/Desktop/Tarea1:/usr/local/apache2/htdocs/ daniel/web5:latest
d76c02d0379fa8a45238694be7e053d72eacdf16100d55605dd539bf71e8c9d9
PS C:\Users\Dani\Desktop\Tarea1> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS               NAMES
d76c02d0379f   daniel/web5:latest "httpd-foreground"      5 seconds ago Up 4 seconds   0.0.0.0:8080->80/tcp web5test
PS C:\Users\Dani\Desktop\Tarea1>
  
```



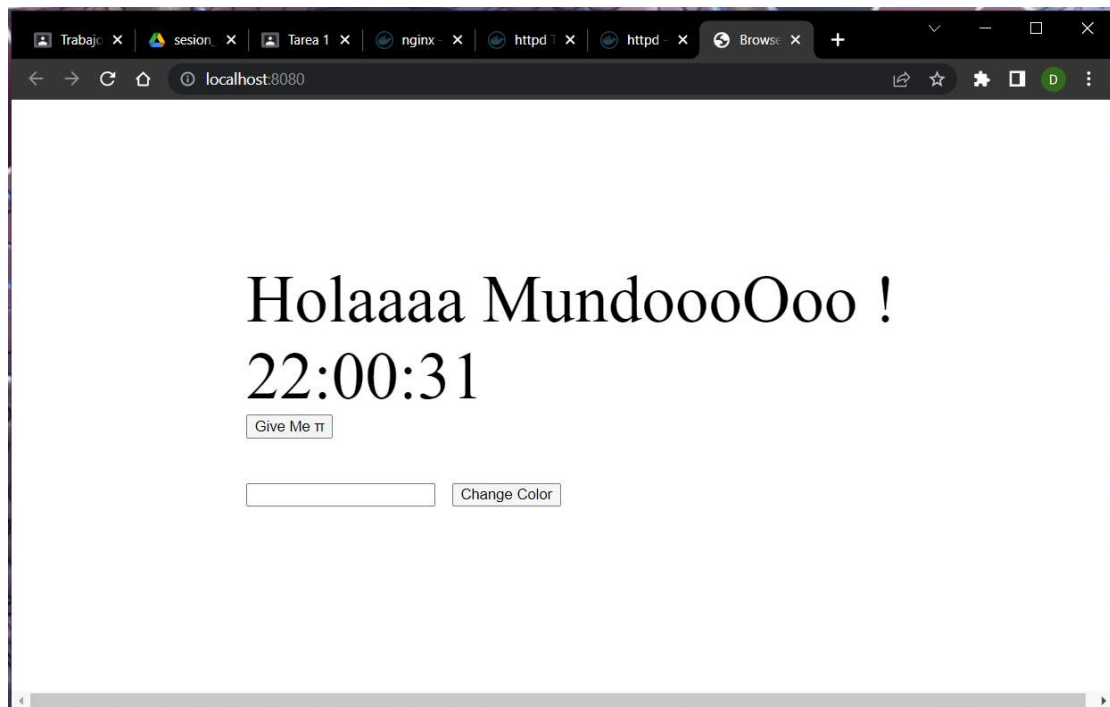
7. (0.25 pts) Sin eliminar o detener el contenedor anterior, modifica el fichero html y verifica que el cambio se aplica en el sitio web (puede ser necesario que actualices la página) ¿Se aplica el cambio? ¿Por qué?



```
1 <html>
2 <head>
3   <meta charset="UTF-8">
4   <link rel="stylesheet" href="style.css">
5   <title>Browser-Based UI</title>
6 </head>
7 <body>
8   <div id="app">
9     <div class="main-area">
10       <div id="message">Holaaaa MundoooOoo !</div>
11       <div id="clock"></div>
12     </div>
13     <div class="control-panel">
14       <button id="show-pi">Give Me  $\pi$ </button>
15       <div>
16         <input type="text" id="color-value">
17         <button id="color-changer">Change Color</button>
18       </div>
19     </div>
20   </div>
21   <script src="index.js"></script>
22 </body>
23 </html>
24
```

0 0

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF HTML Go Live



Por que gracias al mapeo modifica los cambios ocurridos en los archivos de la url especificad

4. Docker Compose

1. (1.5 pts) Escribe el fichero docker-compose.yml que te permita levantar el stack compuesto por:

a. Servicio redis

- i. Imagen oficial de Redis (última versión de alpine)
- ii. Nombre del contenedor: redis
- iii. Puerto 63799 del host enlazado al estándar de Redis
- iv. Sin volúmenes enlazados

b. Servicio redisinsight

- i. Imagen oficial de RedisInsight (última versión de alpine)
- ii. Nombre del contenedor: redisinsight
- iii. Puerto 8111 del host enlazado al estándar de RedisInsight
- iv. Volúmen administrado llamado redisinsight_data enlazado /db

c. Ambos servicios deben estar unidos a una red custom llamada redisnet

```
docker-compose.yml - Visual Studio Code
C:\Users\Dani\Desktop\Tarea1\Compose> docker-compose.yml
1  |--
2
3  networks:
4    redisnet:
5      driver: bridge
6
7  volumes:
8    redisinsight-data:
9
10 services:
11   redis:
12     image: redis:alpine
13     container_name: redis
14     ports:
15       - "63799:6379"
16     networks:
17       - redisnet
18
19   redisinsight:
20     image: redislabs/redisinsight:latest
21     container_name: redisinsight
22     ports:
23       - "8111:8001"
24     volumes:
25       - redisinsight-data:/db
26     networks:
27       - redisnet
28
29
30
31

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF Compose Go Live

Windows PowerShell
PS C:\Users\Dani\Desktop\Tarea1\Compose> docker-compose up
[+] Running 7/23
- redis Pulled 4.3s
- 63b65145d645 Pull complete 0.9s
- 6a83e1b979d3 Pull complete 1.0s
- 33568fda55fd Pull complete 1.4s
- 92c907937b14 Pull complete 1.7s
- ae96d2ab3885 Pull complete 1.8s
- fbd5435e8d0e Pull complete 1.9s
- redisinsight Pulling 6.0s
- bd159e379b3b Extracting [=====] 29.8... 3.6s
- de08aeb7fd50 Download complete 3.6s
- 7091270c8142 Download complete 3.6s
[+] Running 7/236 Download complete 3.6s - redis Pulled
- 63b65145d645 Pull complete 4.3s
- 6a83e1b979d3 Pull complete 0.9s - 6a83e1b979d3 Pull comp
d3 Pull complete 1.0s
- 33568fda55fd Pull complete 1.4s - 92c907937b
14 Pull complete 1.7s
- ae96d2ab3885 Pull complete 1.8s - fbd5435e8d
0e Pull complete 1.9s
[+] Running 7/23Pulling
- redis Pulled 4.3s
- 63b65145d645 Pull complete 0.9s0 - 6a83e1b979d3 Pull comp
lete 1.0s4 - 33568fda55fd Pull complete 1.4s
- 92c907937b14 Pull complete 1.7s0 - ae96d2ab3885 Pull comp
lete 1.7s0
[+] Running 14/23
- redis Pulled 4.3s8 - 63b65145d645 Pull comple
te 0.9s. - 6a83e1b979d3 Pull complete
```

```
Windows PowerShell
[+] Running 23/237b14 Pull complete
- redis Pulled 4.3s
- 63b65145d645 Pull complete 0.9ss
- 6a83e1b979d3 Pull complete 1.0s9
- 33568fda55fd Pull complete 1.4s.
- 92c907937b14 Pull complete 1.7s4
- ae96d2ab3885 Pull complete 1.8s
- fbd5435e8d0e Pull complete 1.9s
- redisinsight Pulled 32.8ss
- bd159e379b3b Pull complete 4.2s
- de08aeb7fd50 Pull complete 4.4ss
- 7091270c8142 Pull complete 5.0s4
- 2f44657947b6 Pull complete 5.1s.
- 55e2b120f544 Pull complete 5.4s5
- c86f5c882457 Pull complete 5.5s2
- 6cc253bd45b3 Pull complete 5.6s
- 70af7c73ffc9 Pull complete 24.7s
- 8d324e600002 Pull complete 24.9ss
- 2dad24227676 Pull complete 25.0s2
- b4fd899b7335 Pull complete 28.7s.
- 4f4fb700ef54 Pull complete 28.8s5
- f5ad3ec4d516 Pull complete 29.8s2
- 7025349d430 Pull complete 30.3s
- 9aba992ce8bd Pull complete 30.4s
[+] Running 2/2 24.9ssete 17.0s
- Container redisinsight Created 4.0s
- Container redis Created 4.0s
Attaching to redis, redisinsight
redis | 1:C 26 Feb 2023 19:59:33.187 # o00o00o00o0o Redis is starting o00o00o00o0o
redis | 1:C 26 Feb 2023 19:59:33.187 # Redis version=7.0.8, bits=64, commit=00000000, modified=0, pid=1, just started

Windows PowerShell
PS C:\Users\Dani> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
a4823a20ee08 redis:alpine "docker-entrypoint.s..." 3 minutes ago Up 3 minutes 0.0.0.0
:63799->6379/tcp redis
ed1dbf279e94 redislabs/redisinsight:latest "bash ./docker-entry..." 3 minutes ago Up 3 minutes 0.0.0.0
:8111->8001/tcp redisinsight
PS C:\Users\Dani> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
redis alpine ae6b92ae6202 2 weeks ago 29.9MB
redislabs/redisinsight latest 76f704dba29e 3 months ago 1.15GB
PS C:\Users\Dani> |
```

2. (0.5 pts) Accede a la URL <http://localhost:8111> para validar que tienes acceso a la GUI del cliente Redis

