

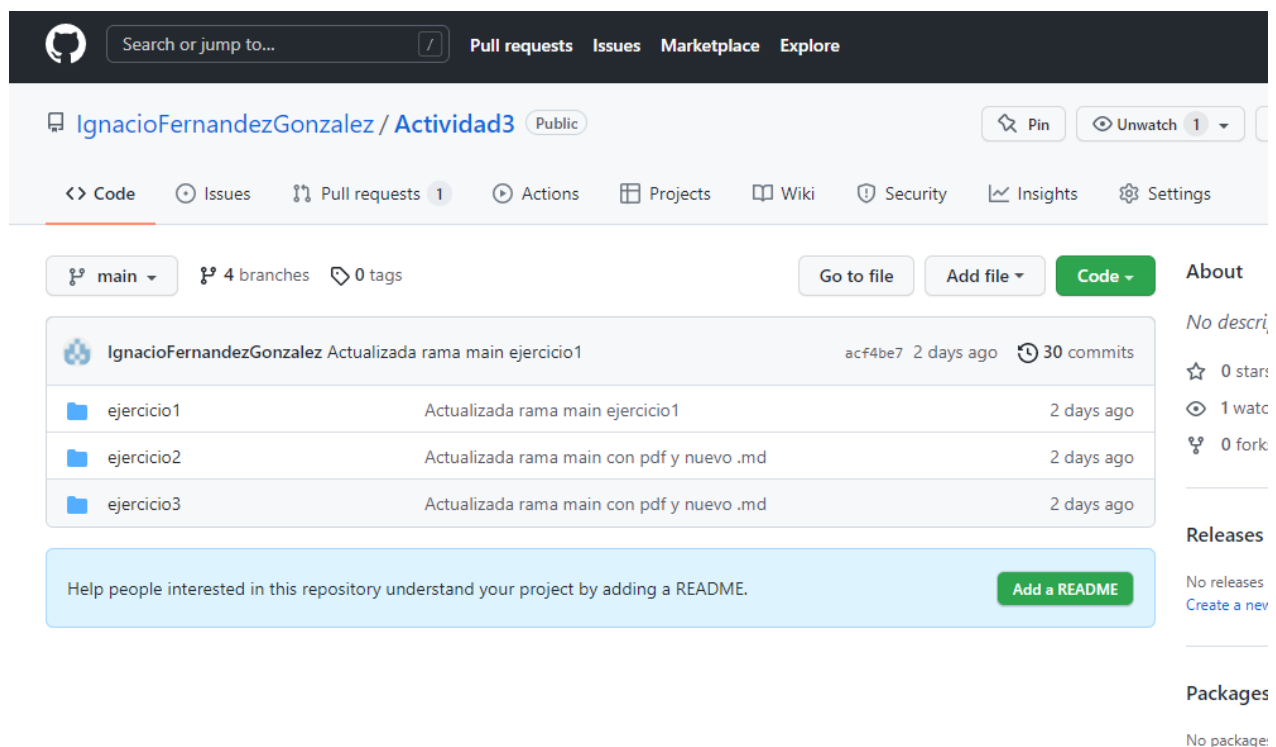
ACTIVIDAD EVALUABLE 3 - GIT Y DOCKER

INTRODUCCION

Para empezar a organizar la entrega de la tarea se va a crear un repositorio GitHub (

[<https://github.com/IgnacioFernandezGonzalez/Actividad3>]:

) para alojar las distintas soluciones de los ejercicios que componen la tarea.



A continuación he generado un directorio en mi disco local para alojar mis ficheros y poder trasladar los cambios a GitHub

Ejercicio 4 - imagen con Dockerfile

En este ejercicio vamos a crear una imagen personalizada basada en nginx, para poder desplegar una pagina web en nuestro propio servidor.

Lo primero que tenemos que hacer es crear nos un directorio en local donde vamos a almacenar los ficheros necesarios para que nuestra pagina web funcione correctamente, además de un fichero Dockerfile, que será el que usemos para montar la imagen deseada.



Ahora tendremos que configurar el fichero Dockerfile, para que tenga las instrucciones necesarias para montar nuestra imagen, como nosotros queramos, el código de este fichero es el siguiente

```
Actividades Editor de textos 1 de abr 19:57
Abrir Dockerfile ~/web
1 FROM nginx:latest
2 COPY . /usr/share/nginx/html
3 COPY index.html /usr/share/nginx/html
4 COPY imagenes.css /usr/share/nginx/html
5
6
```

Una vez que hemos guardado estas modificaciones en el fichero Dockerfile, abrimos una terminal en este mismo directorio y creamos la nueva imagen, en mi caso le he puesto de nombre imagen_nginx_ignacio

Lo haremos con el comando

```
docker build -i imagen_nginx_ignacio .

ignaciofernandez@clienteLinux:~/web$ docker build -t imagen_nginx_ignacio .
Sending build context to Docker daemon 1.465MB
Step 1/4 : FROM nginx:latest
----> 12766a6745ee
Step 2/4 : COPY . /usr/share/nginx/html
----> 2523671cd4bb
Step 3/4 : COPY index.html /usr/share/nginx/html
----> ea69325b7fcb
Step 4/4 : COPY imagenes.css /usr/share/nginx/html
----> f34b40000046
Successfully built f34b40000046
Successfully tagged imagen_nginx_ignacio:latest
```

Ya tenemos creada nuestra propia imagen, ahora nos quedará crear una contenedor con esta imagen, en mi caso lo he llamado mynginx.

Lo haremos con el comando

```
docker run --name mynginx -p 80:80 -d imagen_nginx_ignacio
```

```
ignaciofernandez@clienteLinux:~/web$ docker run --name mynginx -p 80:80 -d image_n_nginx_ignacio_e52172f393c59409d22e08cf0c30ef3160c9a45afd5bd7ecfa55f21356e40128
```

Ahora ya podremos acceder a nuestra web desde nuestro navegador



Por último vamos a subir nuestra imagen de nginx modificada a nuestro repositorio de Docker Hub, para ello tendremos que como primer paso crear una espejo de nuestra imagen, lo haremos con el comando

```
docker commit e52172f393c5 nginx_ignacio:1.0
```

```
ignaciofernandez@clienteLinux:~/web$ docker commit e52172f393c5 nginx_ignacio:1.0  
sha256:05a8f9e84d367121df72a33f06adcb601b3edbffa4a234fcc6909797346bd3c5
```

A continuación, nos exportaremos a un fichero .jar este espejo. lo haremos con el comando

```
docker save -o nginx_ignacio.jar nginx_ignacio:1.0
```

```
ignaciofernandez@clienteLinux:~/web$ docker save -o nginx_ignacio.jar nginx_ignacio:1.0  
ignaciofernandez@clienteLinux:~/web$ ls  
Dockerfile imagenes.css index.html nginx_ignacio.jar recursos
```

Con la imagen ya exportada a nuestro fichero `nginx_ignacio.jar`, lo que nos queda es subir nuestra imagen a nuestra cuenta de Docker Hub para que cualquier persona pueda usarla, incluso nosotros mismos tengamos acceso desde cualquier lado.

Para ello, lo primero que vamos a hacer es modificar el tag de la imagen en cuestión, acorde a nuestra cuenta de Docker Hub (ignaciofernandezgonzalez), lo haremos con el comando

```
docker tag nginx_ignacio:1.0 docker.io/ignaciofernandezgonzalez/nginx_ignacio
```

```
ignaciofernandez@clienteLinux:~/web$ docker tag nginx_ignacio:1.0 docker.io/ignaciofernandez/nginx_ignacio
```

Una vez que lo hemos hecho sólo nos quedará subir nuestra imagen a nuestra cuenta de Docker Hub, para lo que nos tendremos que logear con el comando

```
docker login -i "ignaciofernandezgonzalez" -p "contraseña" docker.io
```

```
ignaciofernandez@clienteLinux:~/web$ docker login -u "ignaciofernandezgonzalez" -p "contraseña" docker.io  
WARNING! Using --password via the CLI is insecure. Use --password-stdin.  
WARNING! Your password will be stored unencrypted in /home/ignaciofernandez/.docker/config.json.  
Configure a credential helper to remove this warning. See  
https://docs.docker.com/engine/reference/commandline/login/#credentials-store  
Login Succeeded
```

Y a continuación, subir la imagen con el comando

```
docker push ignaciofernandezgonzalez/nginx_ignacio
```

```
ignaciofernandez@clienteLinux:~/web$ docker push ignaciofernandezgonzalez/nginx_ignacio
Using default tag: latest
The push refers to repository [docker.io/ignaciofernandezgonzalez/nginx_ignacio]
8cfc84852d25: Pushed
f75460560ef3: Pushed
c626547b7a9e: Pushed
453bf3ffa025: Pushed
ea4bc0cd4a93: Pushed
fac199a5a1a5: Pushed
5c77d760e1f4: Pushed
33cf1b723f65: Pushed
ea207a4854e7: Pushed
608f3a074261: Pushed
latest: digest: sha256:50ec0cf19ee4b37dd0667f23781345db50ed1820eed145c64e4e90b5a32cacfe size: 2402
```

Como se puede observar en la imagen siguiente, se ha subido nuestra imagen "nginx_ignacio" a nuestro repositorio en Docker Hub.

