ACTIVIDAD 5: Modificación y publicación en Docker Hub

5.1. Modificar la imagen

- Empezamos abriendo el nano donde se encuentra nuestro Dockerfile.

```
jesus@vbox:~/docker-dev$ nano Dockerfile
```

- Instalamos las aplicaciones extras que son las ofimáticas, como lo son
  - ✓ sc para tablas Excel
  - ✓ poppler-utils para trabajar archivos pdf.
  - ✓ mupdf es para visualizar archivos pdfs
  - ✓ w3m sirve para abrir páginas web o archivos html.

```
# Instalacion de aplicaciones de ofimatica
RUN apt install -y sc poppler-utils mupdf w3m
```

- Actualizamos los datos trabajados con Docker build.

```
jesus@vbox:~/docker-dev$ docker build -t jesus-ome .
[+] Building 0.5s (10/10) FINISHED                                                                              docker:default
 => [internal] load build definition from Dockerfile                                                                      0.0s
 => => transferring dockerfile: 582B                                                                                      0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest                                                          0.3s
 => [internal] load .dockerignore                                                                                         0.0s
 => => transferring context: 2B                                                                                           0.0s
 => [1/6] FROM docker.io/library/ubuntu:latest@sha256:c35e29c9450151419d9448b0fd75374fec4fff364a27f176fb458d472dfc9e54    0.0s
 => => resolve docker.io/library/ubuntu:latest@sha256:c35e29c9450151419d9448b0fd75374fec4fff364a27f176fb458d472dfc9e54    0.0s
 => CACHED [2/6] RUN apt update && apt upgrade -y                                                                         0.0s
 => CACHED [3/6] RUN apt install -y vim nano emacs git curl wget gcc g++ make                                             0.0s
 => CACHED [4/6] RUN apt install -y sc poppler-utils mupdf w3m                                                            0.0s
 => CACHED [5/6] RUN useradd -ms /bin/bash jesus                                                                          0.0s
 => CACHED [6/6] WORKDIR /home/jesus                                                                                      0.0s
 => exporting to image                                                                                                    0.1s
 => => exporting layers                                                                                                   0.0s
 => => exporting manifest sha256:d8e5a0d07275d79f84f73a0e1b7416cd50560b415db82a8316ab39f1e27af877                         0.0s
 => => exporting config sha256:8fe496620da9a4aa2aded39ec74a66b1e01656abfd205e897d81a281df129a69                           0.0s
 => => exporting attestation manifest sha256:9af951d59b67b9358425a7cf6b5df48e4b2423bfcaf0e539bcebb63796369732             0.0s
 => => exporting manifest list sha256:06d8f396a71eecc76e7572c10c18e886cb205bf096434f2ea8b06acfbc67bf5c                    0.0s
 => => naming to docker.io/library/jesus-ome:latest                                                                       0.0s
 => => unpacking to docker.io/library/jesus-ome:latest                                                                    0.0s
```

- Iniciamos la imagen.

```
jesus@vbox:~/docker-dev$ docker run -it jesus-ome bash
jesus@c54f9b07a63a:~$ sc
```

- Hacemos la prueba de los programas de ofimática.
  Ejemplo de uso:

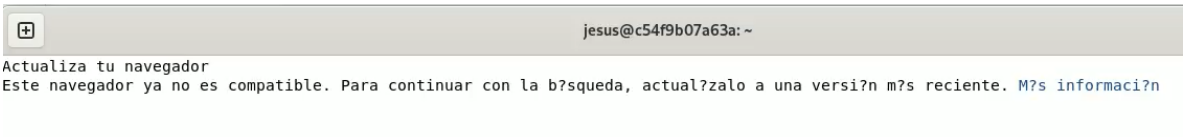
```
A2 (10 2 0) [A0+A2]

     A       B       C       D       E       F       G       H
 0    10.00
 1    20.00
 2   200.00
 3    30.00
 4
 5
 6
 7
 8
 9
10
11
12
```

- Prueba w3m

```
jesus@vbox:~/docker-dev$ docker run -it jesus-ome bash
jesus@c54f9b07a63a:~$ sc
jesus@c54f9b07a63a:~$ w3m google.com
```

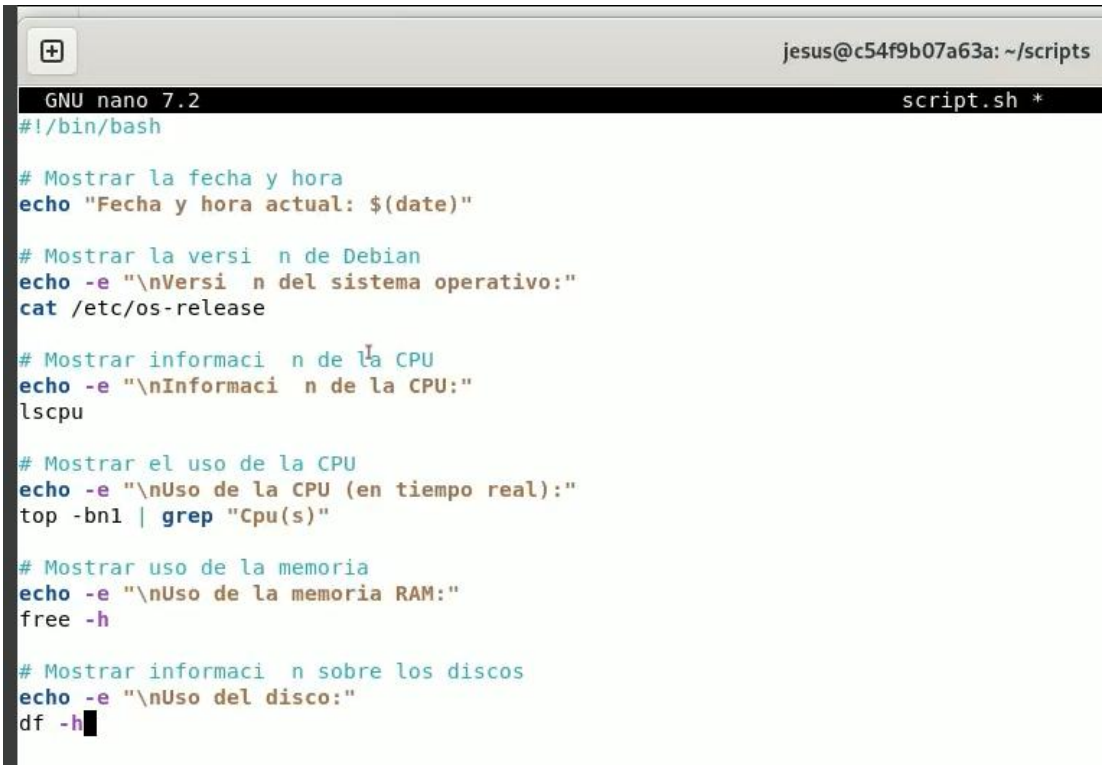
- Sirve para navegación y abrir archivos HTML pero se muestra que Google no es compatible con el.



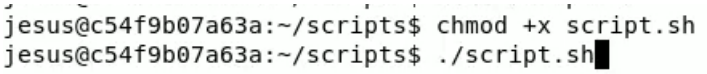
- Para la creación del script, creamos la carpeta donde lo vamos a guardar, ingresamos a la carpeta y creamos en nano el script.sh indicando que es un script.

```
jesus@c54f9b07a63a:~$ mkdir scripts
jesus@c54f9b07a63a:~$ ls
SC.SAVE  scripts
jesus@c54f9b07a63a:~$ cd ~/scripts
jesus@c54f9b07a63a:~/scripts$ nano script.sh
```

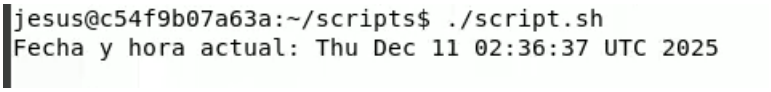
- Después pegamos el script, con #/!bin/bash le indicamos al sistema que es un bash.



- Damos permiso para poderlo hacer funcionar, y como vemos nos muestra toda la información de sistema, y uso de la cpu y otros datos importantes de la máquina virtual.



- Con echo imprimimos en la terminal la información.



- Con cat hacemos que se vea la información del sistema.

```
Versión del sistema operativo:
PRETTY_NAME="Ubuntu 24.04.3 LTS"
NAME="Ubuntu"
VERSION_ID="24.04"
VERSION="24.04.3 LTS (Noble Numbat)"
VERSION_CODENAME=noble
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=noble
LOGO=ubuntu-logo
```

- Con lscpu hacemos que sea vea la informacion de la cpu.

```
Información de la CPU:
Architecture:                x86_64
  CPU op-mode(s):            32-bit, 64-bit
  Address sizes:             48 bits physical, 48 bits virtual
  Byte Order:                Little Endian
CPU(s):                      4
  On-line CPU(s) list:       0-3
Vendor ID:                   AuthenticAMD
  Model name:                AMD Ryzen 5 2600 Six-Core Processor
    CPU family:              23
    Model:                   8
    Thread(s) per core:      1
    Core(s) per socket:      4
    Socket(s):               1
    Stepping:                2
    BogoMIPS:                6799.98
    Flags:                   fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx
                              mmxext fxsr_opt rdtscp lm constant_tsc rep_good nopl nonstop_tsc cpuid extd_apicid tsc_known_freq pni pclmulqd
                             q ssse3 fma cx16 sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx f16c rdrand hypervisor lahf_lm cmp_legacy cr8_
                             legacy abm sse4a misalignsse 3dnowprefetch ssbd vmmcall fsgsbase bmi1 avx2 bmi2 rdseed adx clflushopt sha_ni ar
                             at
```

- Con top -bn1 | grep "Cpu(s)" se muestra el uso actual de la CPU.

```
Uso de la CPU (en tiempo real):
%Cpu(s):  2.4 us,  2.4 sy,  0.0 ni, 95.2 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
```

- Con free -h mostramos la info de la memoria ram libre y utilizada.

```
Uso de la memoria RAM:
               total        used        free      shared  buff/cache   available
Mem:           7.8Gi       2.8Gi       297Mi        35Mi       5.0Gi       4.9Gi
Swap:          1.1Gi        12Ki       1.1Gi
```

- Con df -h cuando espacio hay en el disco duro.

```
Uso del disco:
Filesystem      Size  Used Avail Use% Mounted on
overlay          19G   13G  4.9G  73% /
tmpfs            64M     0   64M   0% /dev
shm              64M     0   64M   0% /dev/shm
/dev/sda1        19G   13G  4.9G  73% /etc/hosts
tmpfs           3.9G     0  3.9G   0% /proc/acpi
tmpfs           3.9G     0  3.9G   0% /proc/asound
tmpfs           3.9G     0  3.9G   0% /sys/devices/virtual/powercap
tmpfs           3.9G     0  3.9G   0% /sys/firmware
```

5.2. Publicación en Docker Hub - explicaremos como subir al repositorio de Docker Hub nuestra imagen.

- primer paso le agregamos una etiqueta a la imagen que vamos a subir, antes de hacer todo este proceso debimos haber iniciado sesión del DockerHub en la terminal debian.

```
jesus@vbox: ~/docker-dev
jesus@vbox:~/docker-dev$ docker tag jesus-ome jesusmartinezomeeee/jesus-ome:latest
```

- Cómo podemos ver en docker images, esta nuestra etiqueta de la imagen creada.

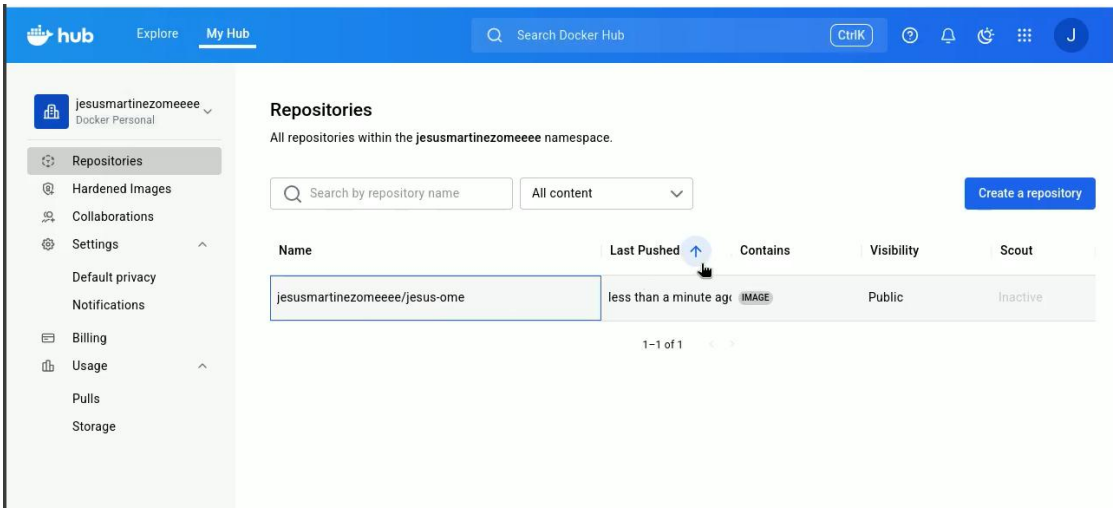
```
jesus@vbox: ~/docker-dev
jesus@vbox:~/docker-dev$ docker tag jesus-ome jesusmartinezomeeee/jesus-ome:latest
jesus@vbox:~/docker-dev$ docker images

IMAGE                                ID             DISK USAGE   CONTENT SIZE   EXTRA
hello-world:latest                   f7931603f70e       25.9kB         9.52kB
jesus-ome:latest                     3723ad30e6da       1.99GB          524MB
jesusmartinezomeeee/jesus-ome:latest 3723ad30e6da       1.99GB          524MB
```

- Con docker push lo subimos a nuestro repositorio

```
jesus@vbox:~/docker-dev$ docker push jesusmartinezomeeee/jesus-ome:latest
The push refers to repository [docker.io/jesusmartinezomeeee/jesus-ome]
eee43be0ef26: Pushed
4f4fb700ef54: Pushed
20043066d3d5: Pushed
5790dfc9467f: Pushed
98b2e740fdff: Pushed
latest: digest: sha256:3723ad30e6dae3963640e9d1b039e851aa6e1a1dce4dd6103889717da95108e4 size: 856
jesus@vbox:~/docker-dev$
```

- Como apreciamos, se encuentra en el repositorio Docker.



Link: https://hub.docker.com/r/jesusmartinezomeeee/jesus-ome