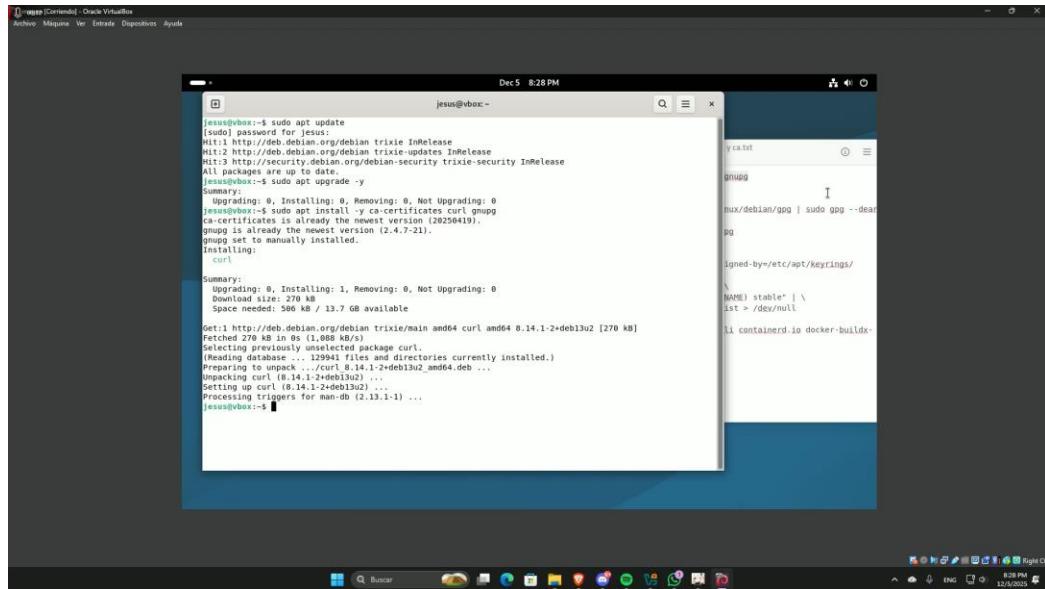


DOCUMENTO ACTIVIDAD N.4

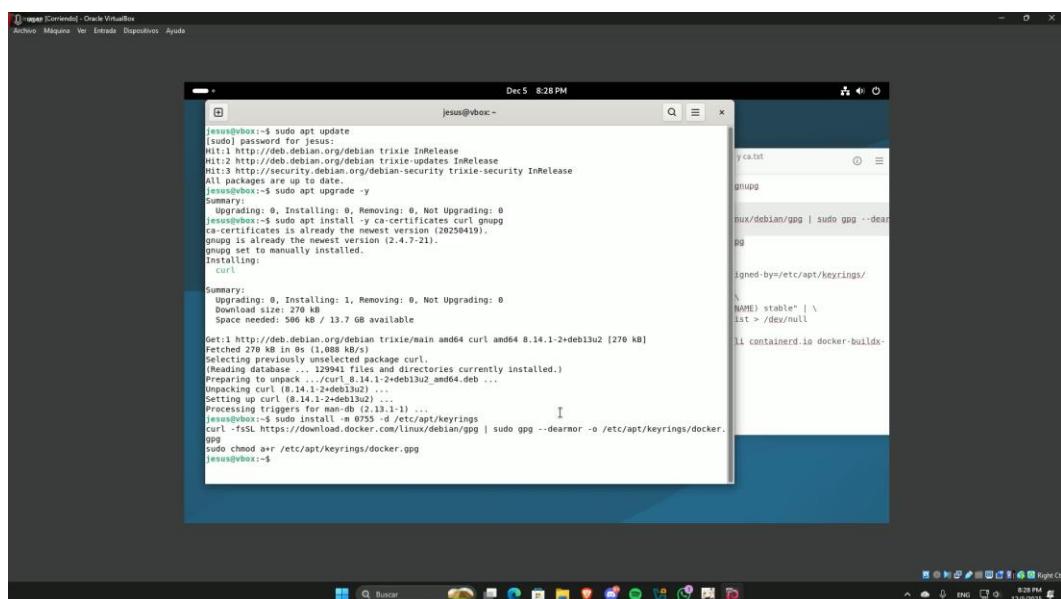
A continuación, se anexan las capturas con sus respectivas explicaciones

4.1 INSTALACION DOCKER ENGINE EN DEBIAN:

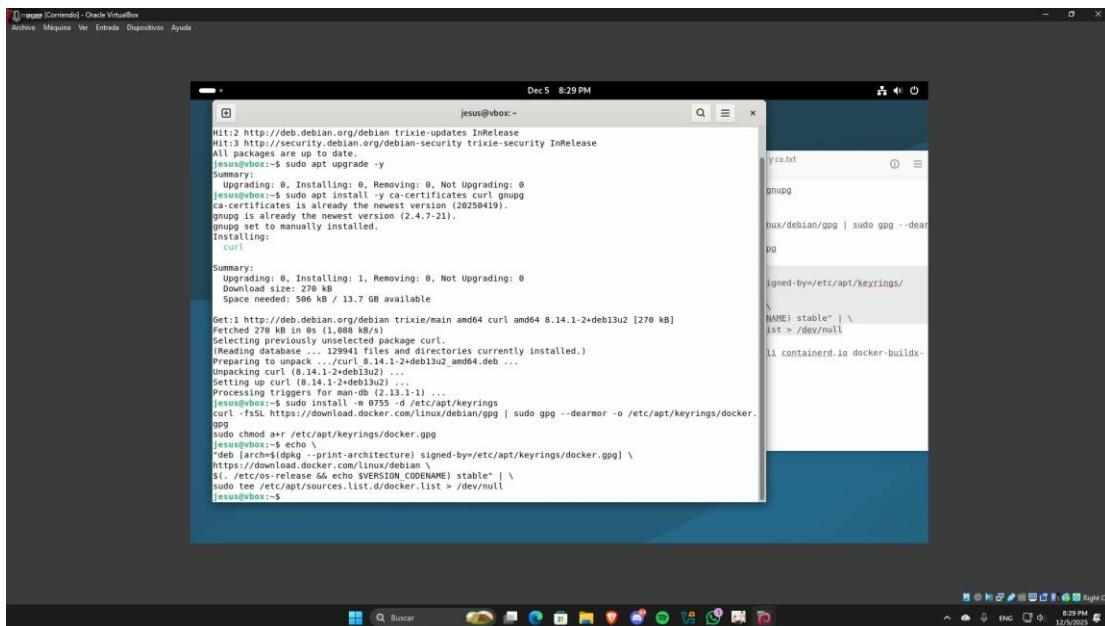
1.actualizamos paquetes y versiones, también instalamos las ultimas actualizaciones disponibles



2. verificamos certificados, el curl es la herramienta que permite descargar este contenido, el gnupg verifican la autenticidad del repositorio docker



3. Este bloque de comandos crea la carpeta donde se guardarán las llaves de seguridad de los repositorios (/etc/apt/keyrings), descarga la llave GPG oficial de Docker y la convierte al formato correcto para que el sistema pueda usarla, y finalmente le da permisos de lectura para que apt pueda verificar que los paquetes instalados desde el repositorio de Docker son auténticos y seguros, el echo construye la línea para conectar al Docker para el apt pueda saber dónde buscar los paquetes para actualizar.

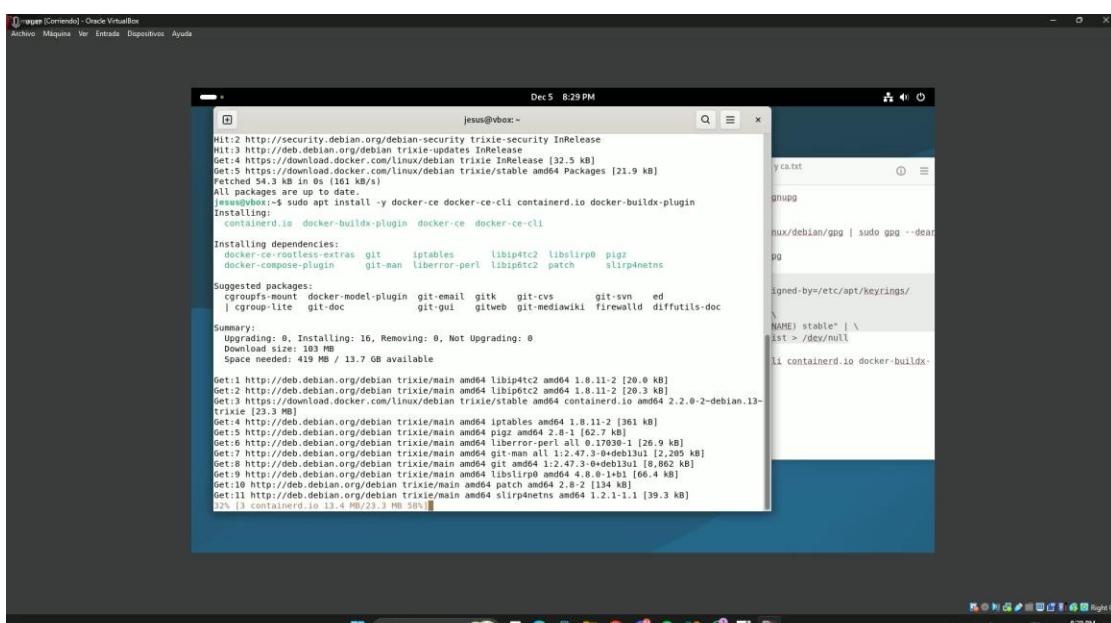


```

Dec 5 8:29PM
jesus@vbox:~$ sudo apt upgrade -y
Hit:2 http://deb.debian.org/debian trixie-updates InRelease
Hit:3 http://security.debian.org/debian-security trixie-security InRelease
All packages are up to date.
jesus@vbox:~$ sudo apt install -y ca-certificates curl gnupg
ca-certificates is already the newest version (20250419).
gnupg is already the newest version (2.4.7-21).
gnupg was manually installed.
Installing:
  curl
Summary:
  Upgrading: 0. Installing: 1. Removing: 0. Not Upgrading: 0
  Download size: 278 kB
  Space needed: 506 kB / 13.7 GB available
Get:1 http://deb.debian.org/debian/main amd64 curl amd64 8.14.1-2+deb13u2 [278 kB]
Fetched 278 kB in 0s (1,000 kB/s)
Selecting previously unselected package curl.
(Reading database ... 1299 files and directories currently installed.)
Preparing to unpack .../curl_8.14.1-2+deb13u2_amd64.deb ...
Unpacking curl (8.14.1-2+deb13u2) ...
Setting up curl (8.14.1-2+deb13u2) ...
Processing triggers for man-db (2.13.1-1)
jesus@vbox:~$ sudo install -m 0755 -o /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
jesus@vbox:~$ echo \
"deb [arch=amd64] https://download.docker.com/linux/debian \
$VERSION_CODENAME stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
jesus@vbox:~$"

```

4. instalamos el docker engine y sus paquetes relacionados



```

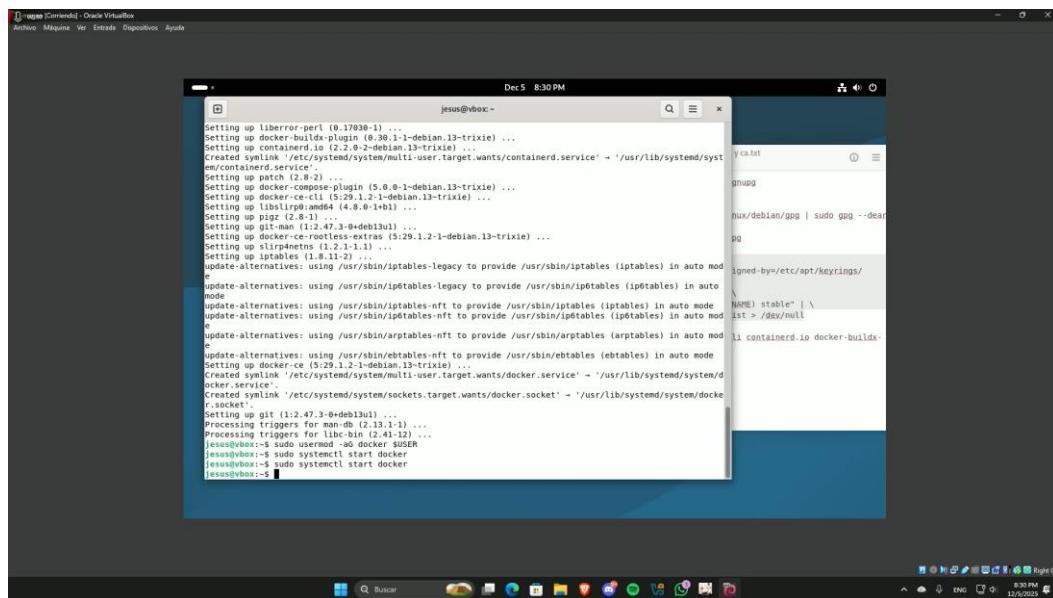
Dec 5 8:29PM
jesus@vbox:~$ sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin
Reading package lists...
Building dependency tree...
Suggested packages:
  docker-ce-rootless-extras git iptables libipipctc libslirp0 pigz
  docker-compose-plugin git-man liberror-perl libipipctc patch slirp4netns
Installing dependencies:
  docker-ce-rootless-extras git iptables libipipctc libslirp0 pigz
  docker-compose-plugin git-man liberror-perl libipipctc patch slirp4netns

Suggested packages:
  cgroups-mount docker-model-plugin git-email gitk git-cvs git-svn ed
  | cgroup-lite git-doc git-email git-gui gitweb git-mediawiki firewalld diffutils-doc

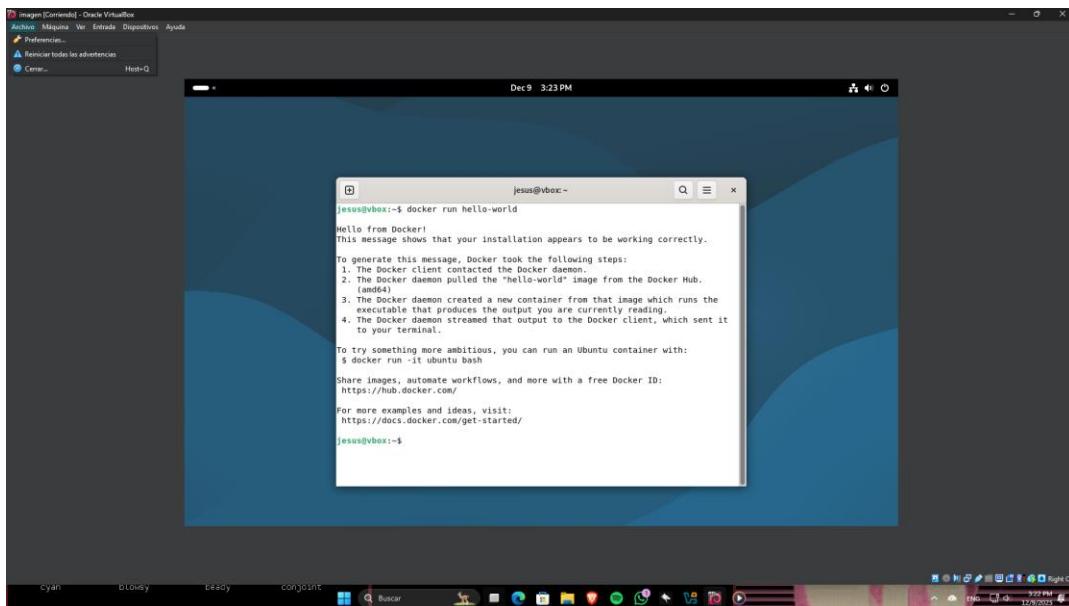
Summary:
  Upgrading: 0. Installing: 16. Removing: 0. Not Upgrading: 0
  Download size: 413 MB
  Space needed: 413 MB / 13.7 GB available
Get:1 http://deb.debian.org/debian/main amd64 libipipctc amd64 1.8.11-2 [20.0 kB]
Get:2 http://deb.debian.org/debian/trixie/main amd64 libipipctc2 amd64 1.8.11-2 [20.3 kB]
Get:3 https://download.docker.com/linux/debian trixie/stable amd64 containerd.io amd64 2.2.0-2~debian.13-trixie [23.3 MB]
Get:4 http://deb.debian.org/debian/main amd64 ipables amd64 1.8.11-2 [361 kB]
Get:5 http://deb.debian.org/debian/trixie/main amd64 ipables amd64 2.8-1 [62.7 kB]
Get:6 http://deb.debian.org/debian/trixie/main amd64 ipaz amd64 2.8-1 [62.7 kB]
Get:7 http://deb.debian.org/debian/trixie/main amd64 git-man all 1:2.47.3-0+deb13u1 [2,205 kB]
Get:8 http://deb.debian.org/debian/trixie/main amd64 git amdgpu 1:2.47.3-0+deb13u1 [8,862 kB]
Get:9 http://deb.debian.org/debian/trixie/main amd64 gitk amdgpu 1:2.47.3-0+deb13u1 [66.4 kB]
Get:10 http://deb.debian.org/debian/trixie/main amd64 patch amdgpu 2.8-2 [134 kB]
Get:11 http://deb.debian.org/debian/trixie/main amd64 slirp4netns amd64 1.2.1-1.1 [39.3 kB]
12% [3 containerd.io 13.4 MB/23.3 MB 505]

```

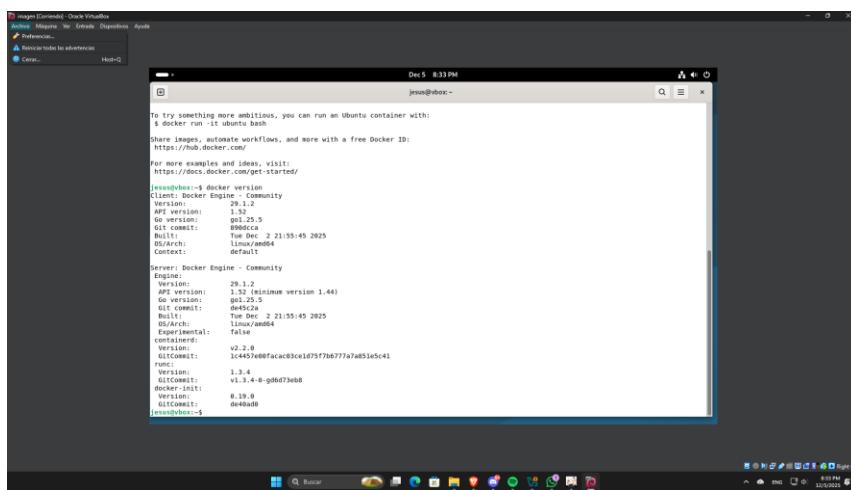
5. agregamos al usuario al grupo docker, para no poner sudo cada vez, después habilitamos el servidor docker y iniciamos el Docker

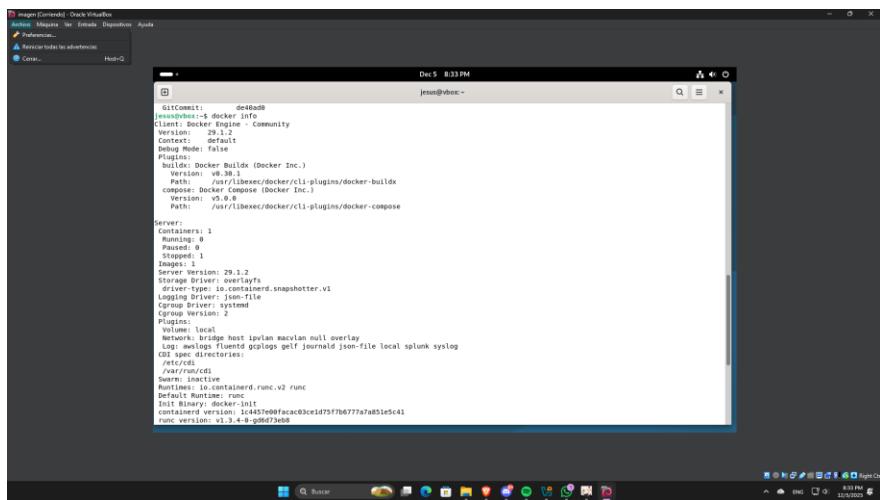


6. descargamos una imagen de prueba



7. Docker info y Docker versión

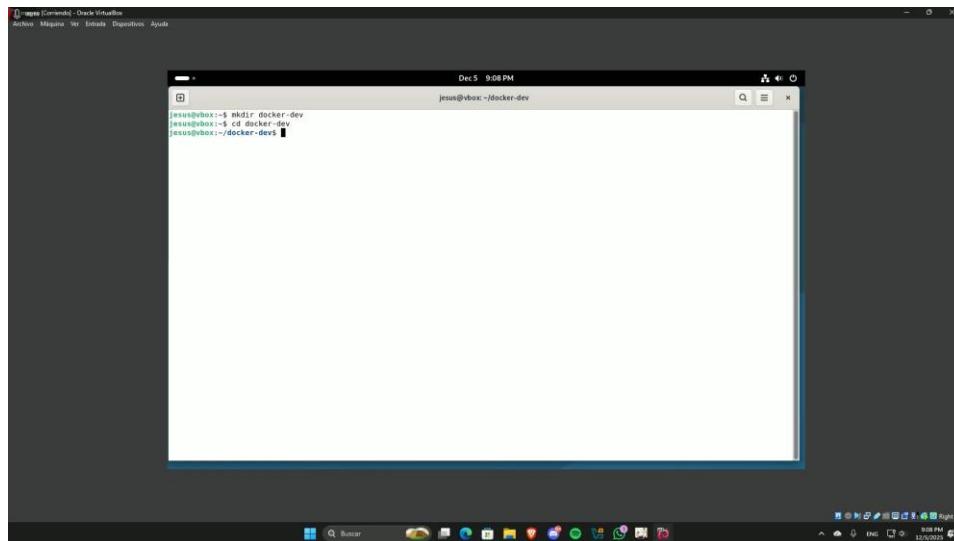




```
[root@vbox ~]# docker info
Client: Docker Engine - Community
 Version: 20.10.12
 API version: 1.41
 Go version: go1.13.15
 Git commit: d96a85a
 Built: 05/05/2021 15:45:41
 OS/Arch: linux/amd64
 Experimental: false
 Plugins:
  buildx (Docker Inc.)
  v2 (Docker Inc.)
  Path: /usr/libexec/docker/cli-plugins/docker-buildx
  compose (Docker Inc.)
  Path: /usr/libexec/docker/cli-plugins/docker-compose
  v5.0.8
  Path: /usr/libexec/docker/cli-plugins/docker-compose
  server (Docker Inc.)
  containers: 1
  Running: 0
  Paused: 0
  Stopped: 1
  Images: 1
  Server Version: 29.1.2
  Storage Driver: overlayfs
  graphDriver: zfs
  Log Driver: json-file
  Cgroup Driver: systemd
  Cgroup Version: 2
  Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gelf journald json-file local splunk syslog
  Cache Directories: /etc/crd
  Exec CD: inactive
  Swarm: inactive
  Runtime: runc
  Default Runtime: runc
  Init: busybox
  containerd: 1e449e0bfaac03ce1d75f7bd777a7a851e5c41
  runc: version: v1.3.4-0-gd8d73eb
```

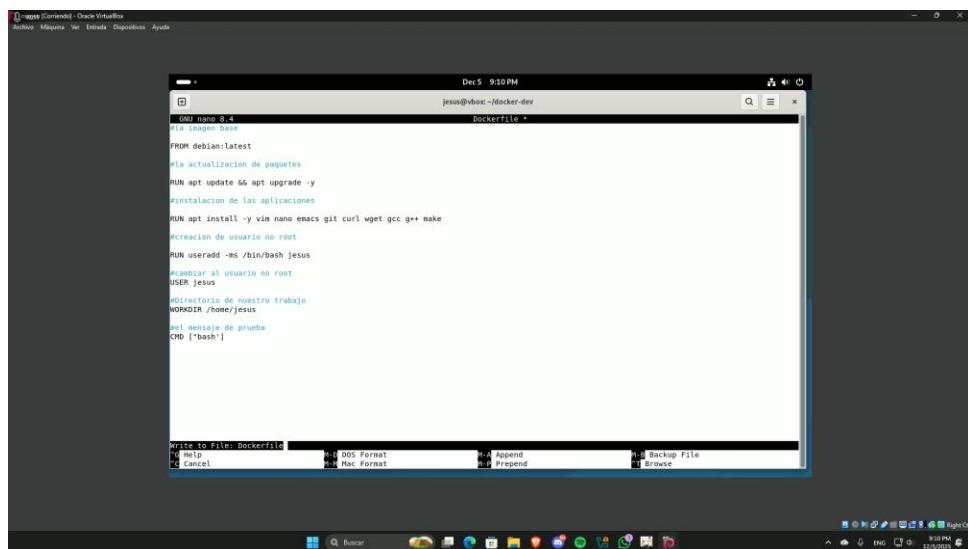
4.2 CREACION DE IMAGEN DOCKER

1. Creación de la carpeta donde se almacenará la imagen creada



```
[root@vbox ~]# mkdir docker-dev
[root@vbox ~]# cd docker-dev
[root@vbox ~]#
```

2. Configuración del dockerfile, en la primera linea de FROM debian, nos indica la imagen base donde trabajaremos el dockerfile, la segunda RUN apt update && apt upgrade -y nos ayuda a que los paquetes esten actualizados en cada momento que se abra el dockerfile, RUN apt install nos descarga las aplicaciones de desarrollo, RUN useradd crea el usuario no root, con el shell /bin/bash y crea su carpeta home -ms, el USER jesus nos ayuda a que las instrucciones se ejecuten como usuario jesus, para mayor seguridad, el WORKDIR establece el directorio donde se trabajara y el CMD es el comando por defecto que se ejecuta al abrir el contenedor.



The screenshot shows a terminal window titled "Dockerfile" running on a Linux host. The terminal content is a Dockerfile script:

```
GNU nano 2.4
#! /usr/bin/python
FROM debian:latest
#actualización de paquetes
RUN apt update && apt upgrade -y
#instalación de las aplicaciones
RUN apt install -y vim nano emacs git curl wget gcc g++ make
#creación de usuario no root
RUN useradd -ms /bin/bash jesus
#cambiar al usuario no root
USER jesus
#directorío de nuestro trabajo
WORKDIR /home/jesus
#el mensaje de prueba
CMD ["bash"]
```

At the bottom of the terminal, there is a "Write to File: Dockerfile" dialog box with options: Help, Cancel, DOS Format, Mac Format, Append, Prepend, Backup File, and Browse. The "Append" option is selected.