

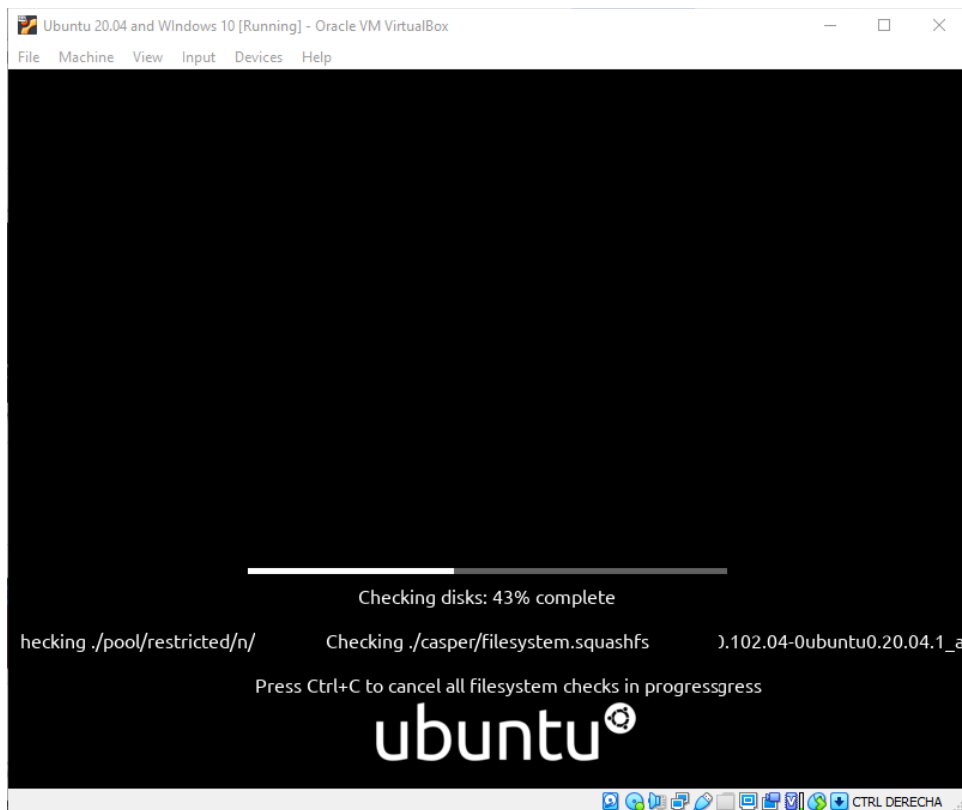
# ASSIGNMENT UNIT2

## **\*Important\***

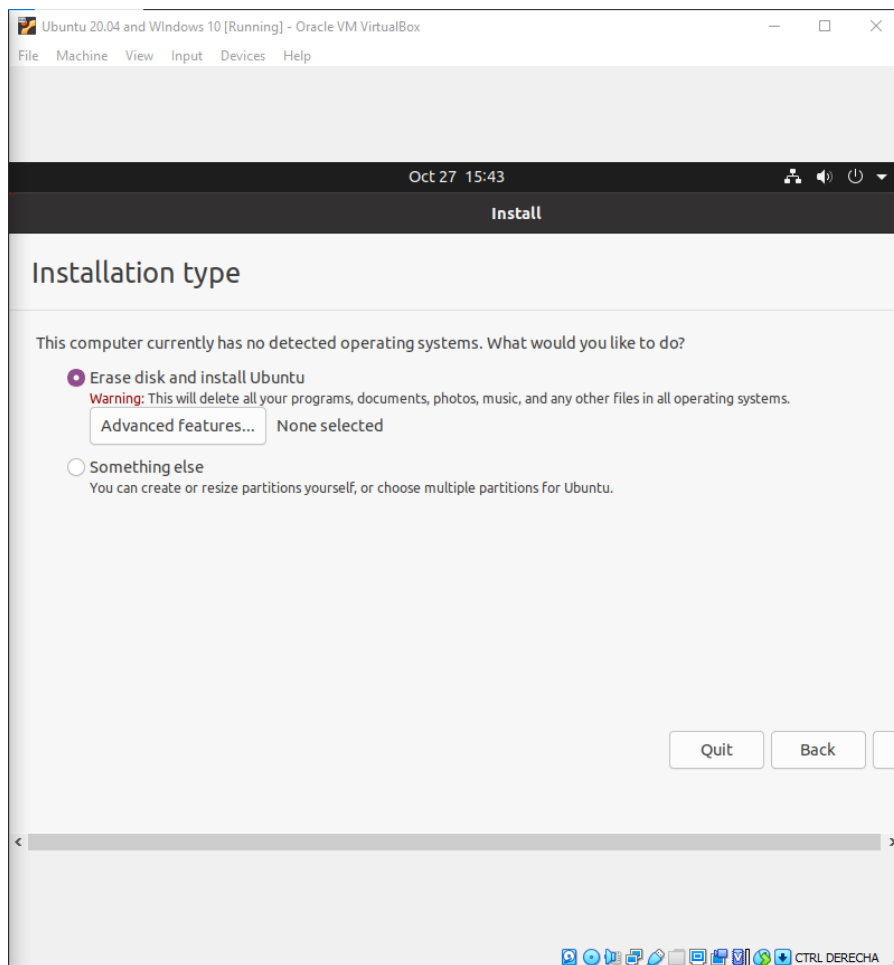
- I changed the name cause reading the assignment I see that my name should be there
- I dont put my name to the Ubuntu System but to the Windows yes
- I'm gonna save the VM if you need to confirm anything

## **Ubuntu installation**

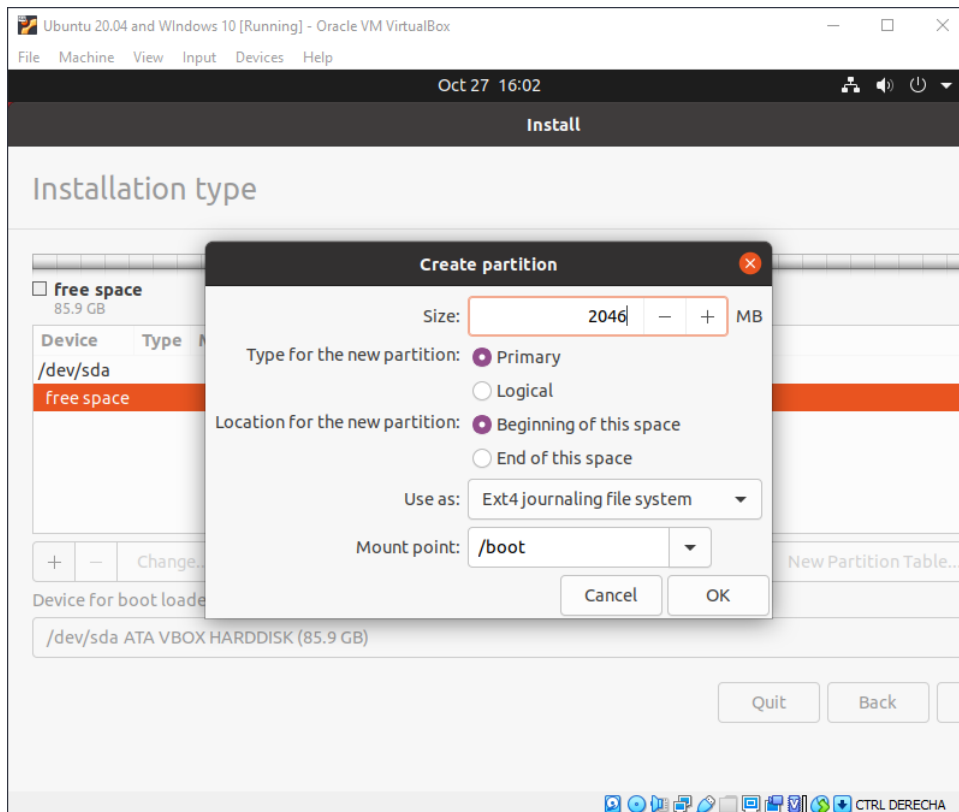
1. I install Ubuntu first and then Windows 10



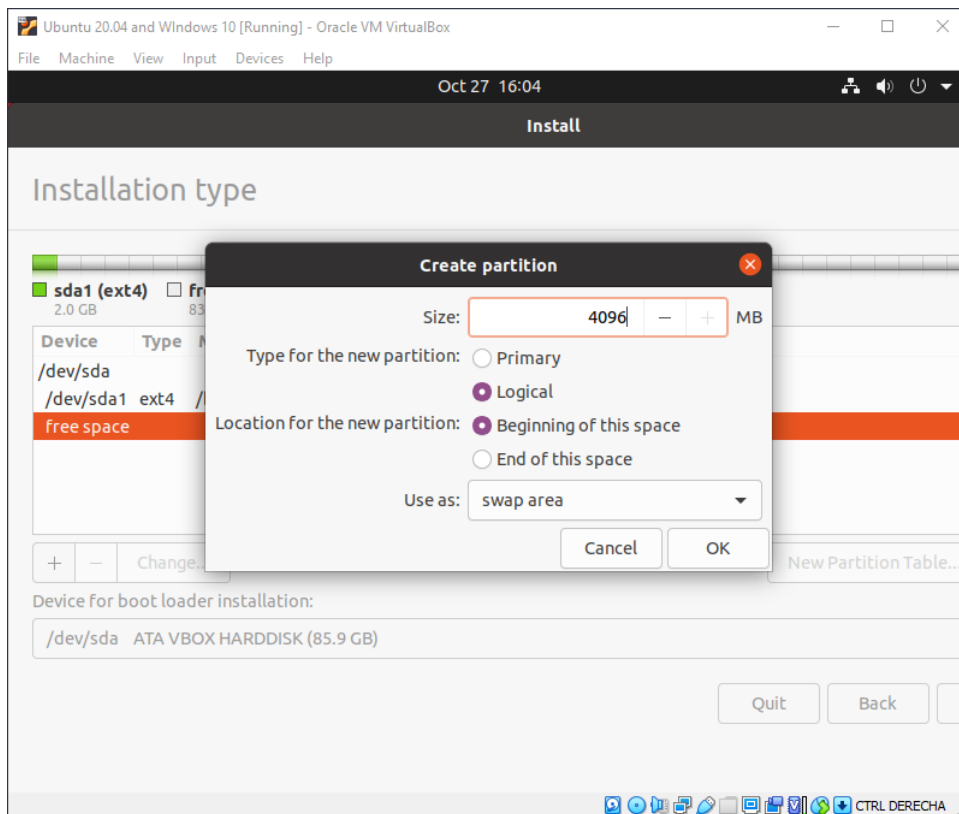
2. At this point of the installation I clicked the "Something else" option



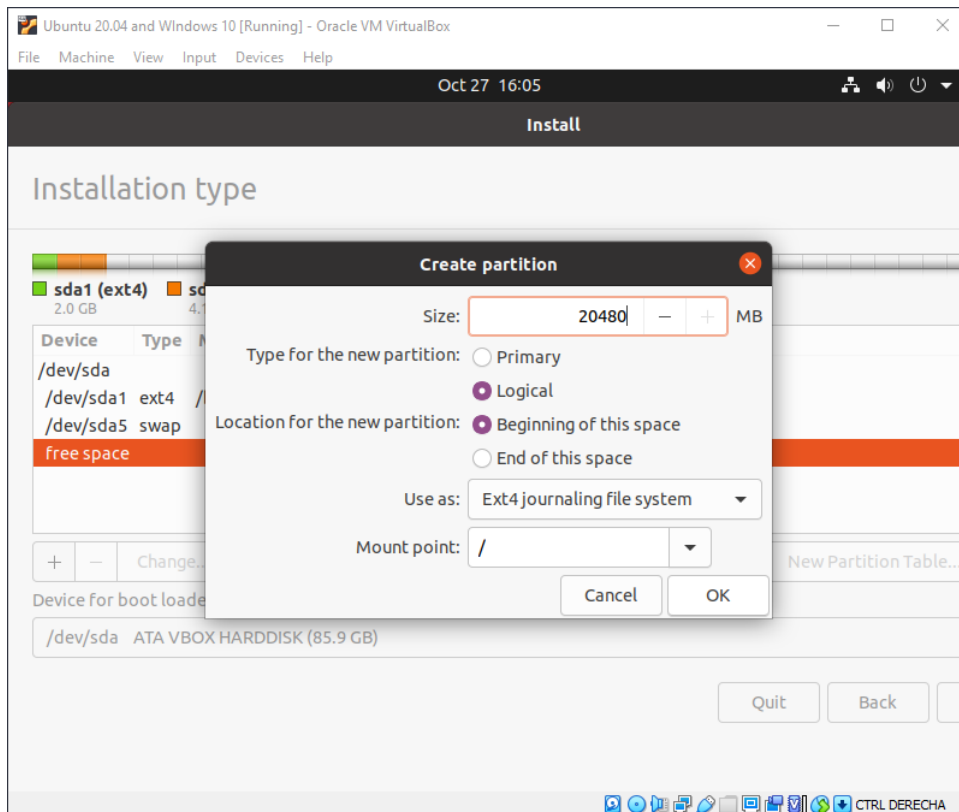
3. This is my first partition on the 80GB disk. Is a primary partition and is for the boot and the format is EXT4



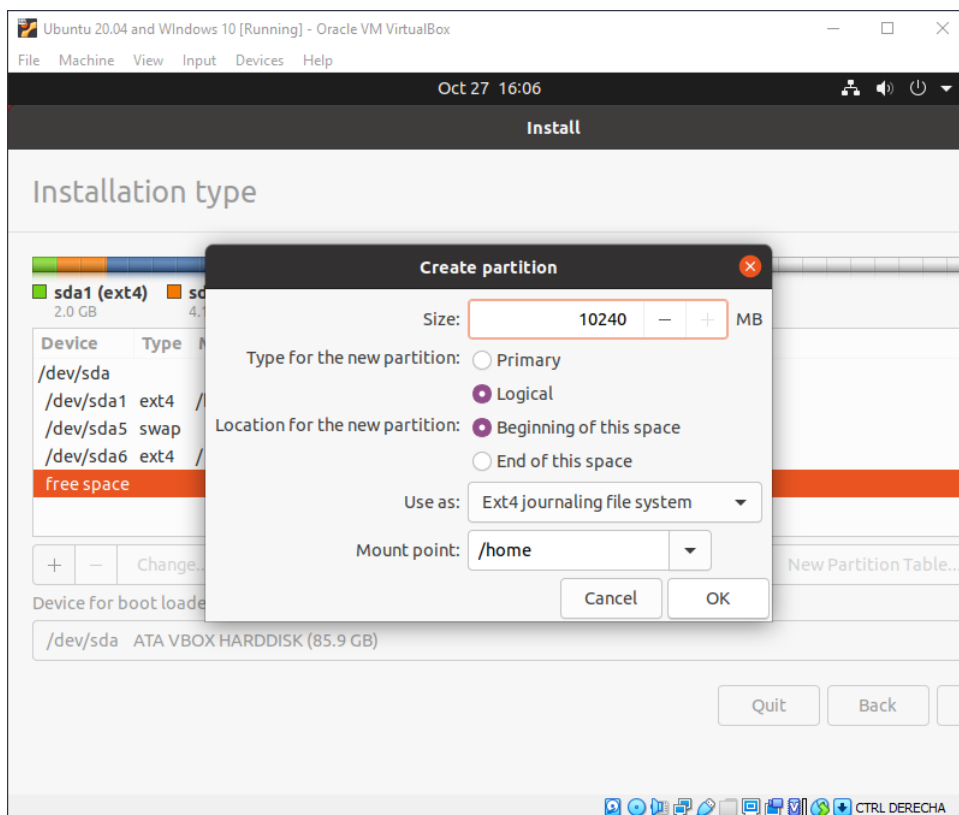
4. On the second partition I create a logical partition of 4GB dedicated to swap area



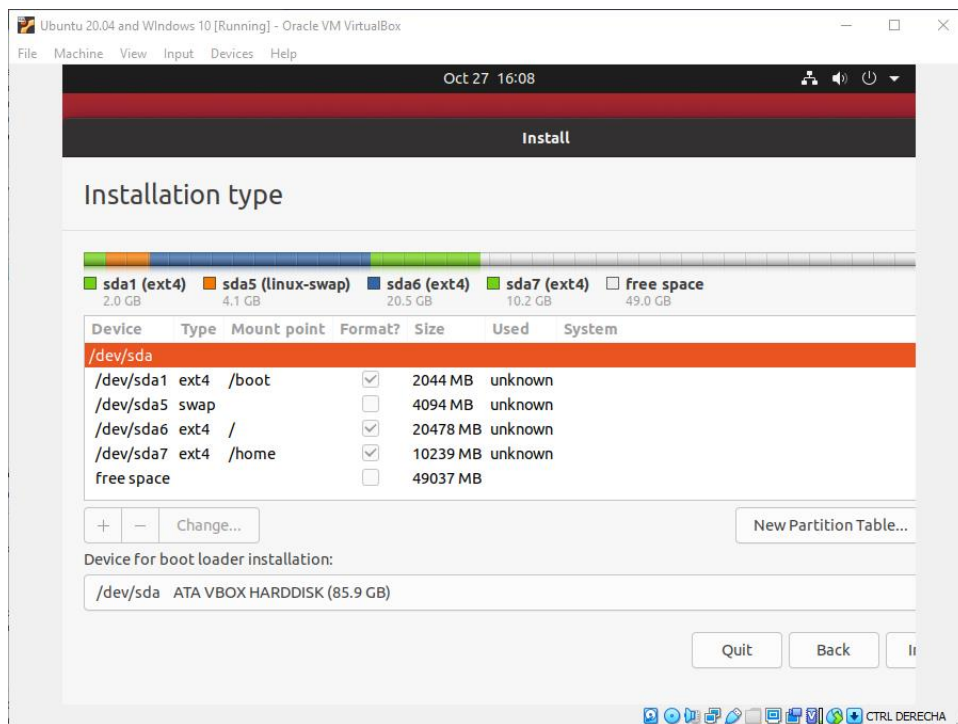
5. On this partition I give 20GB for the root and the type is logical



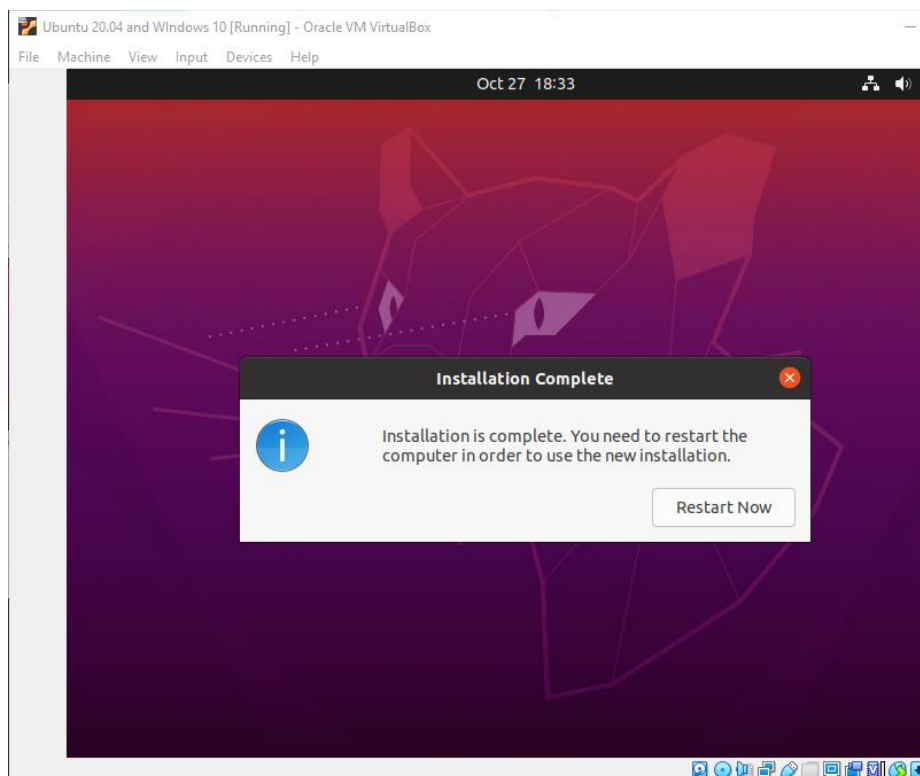
6. On the last partition I use 10GB for home, and the type is logical partition

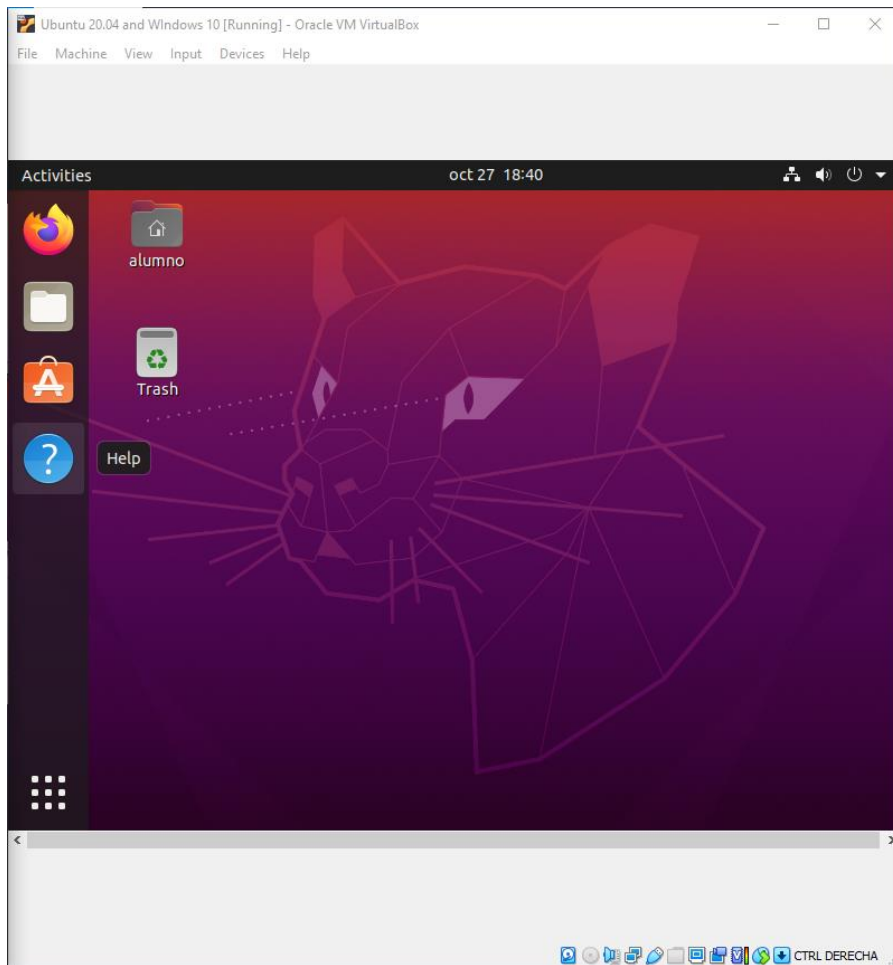


7. This is how I distribute the partition table of the installation of Ubuntu. As you can see I left 50GB for the installation of Windows, that is called free space



8. And Ubuntu is installed

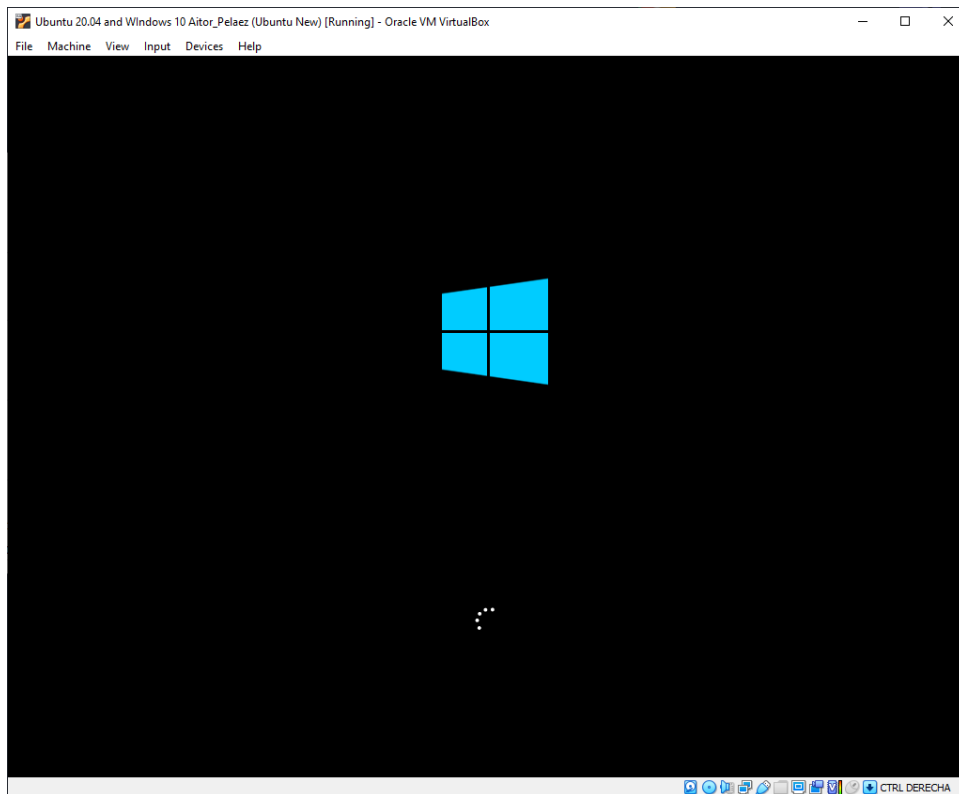




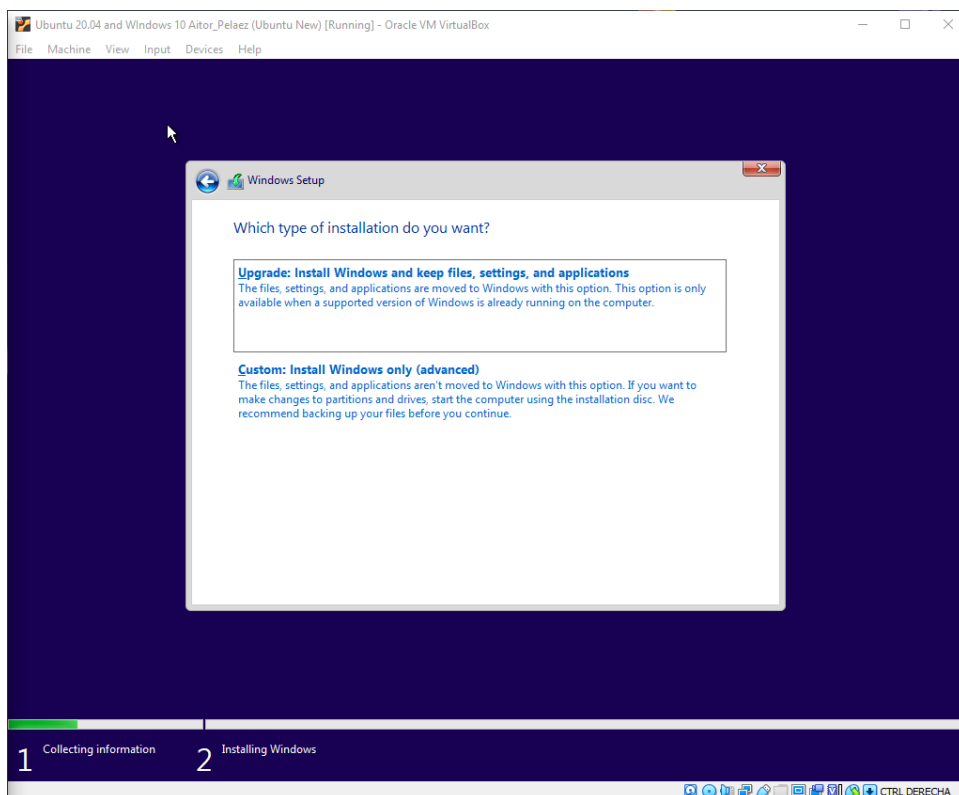
And I create a Snapshot

## Windows Installation

1. We can see how im installing Windows in the same hardrive

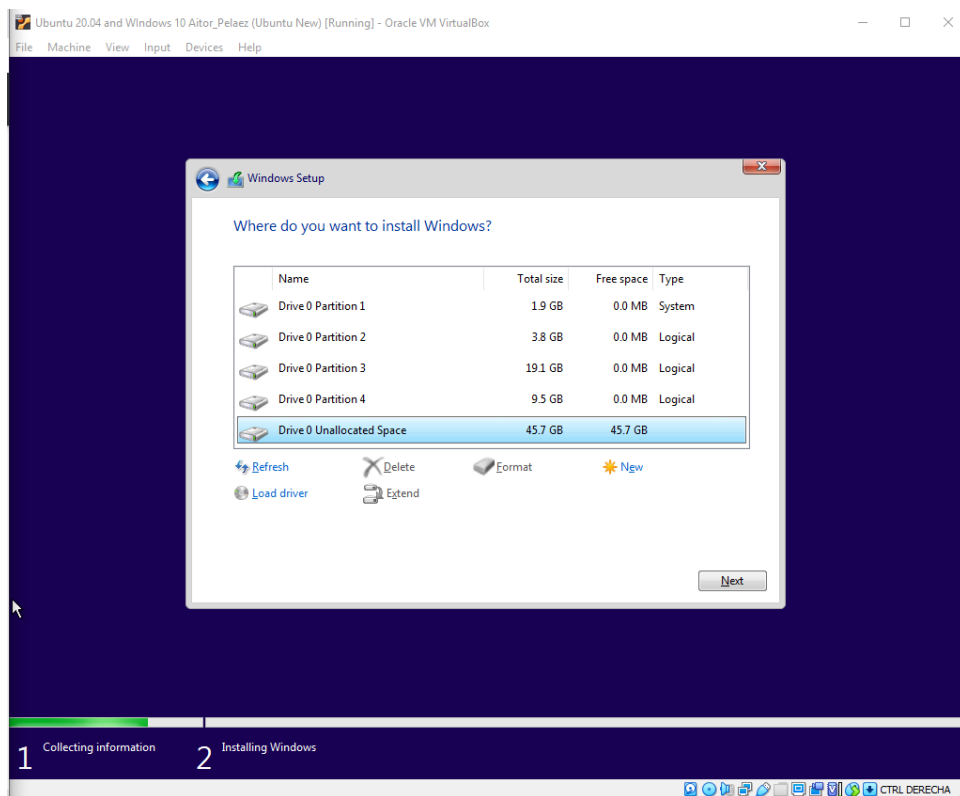


2. When I reach this option i clicked to the second option that is “custom”

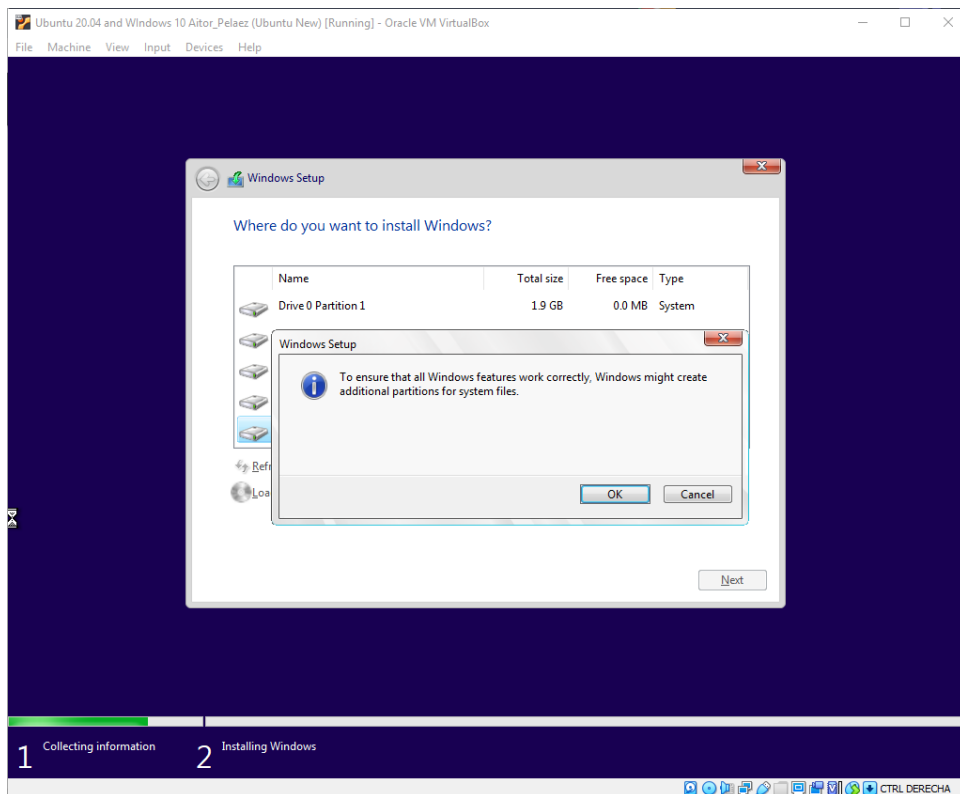


3. Here I selected the free space I left on the first installation of Ubuntu.

Here im going to install Windows



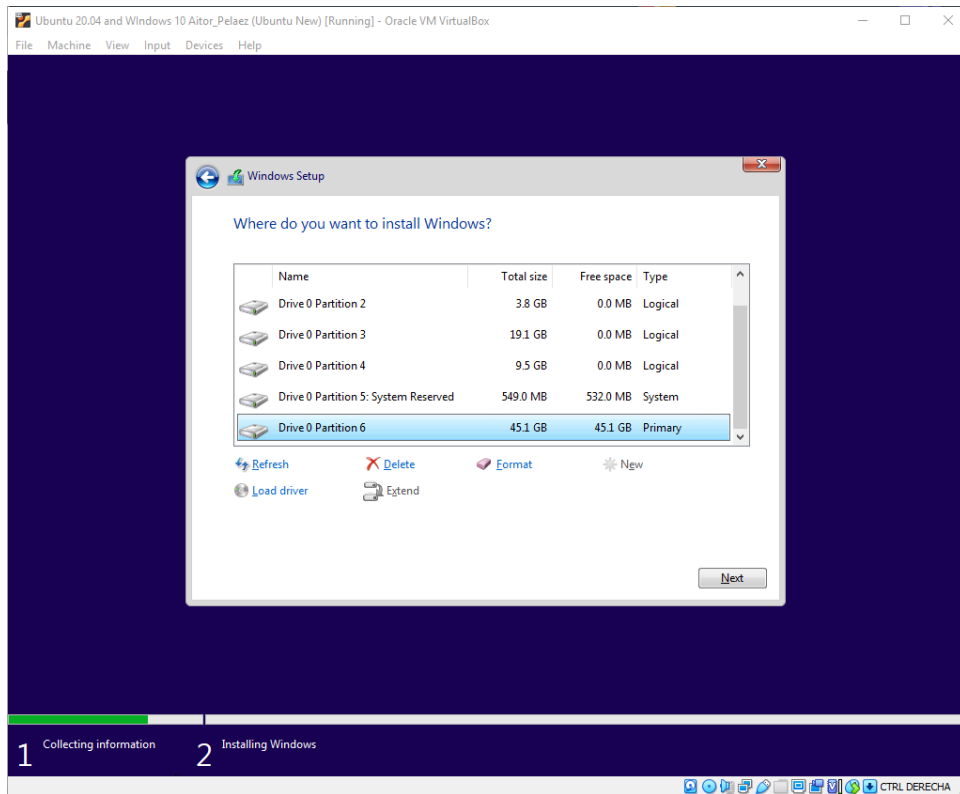
4. You clicked new and this window appear, you click ok



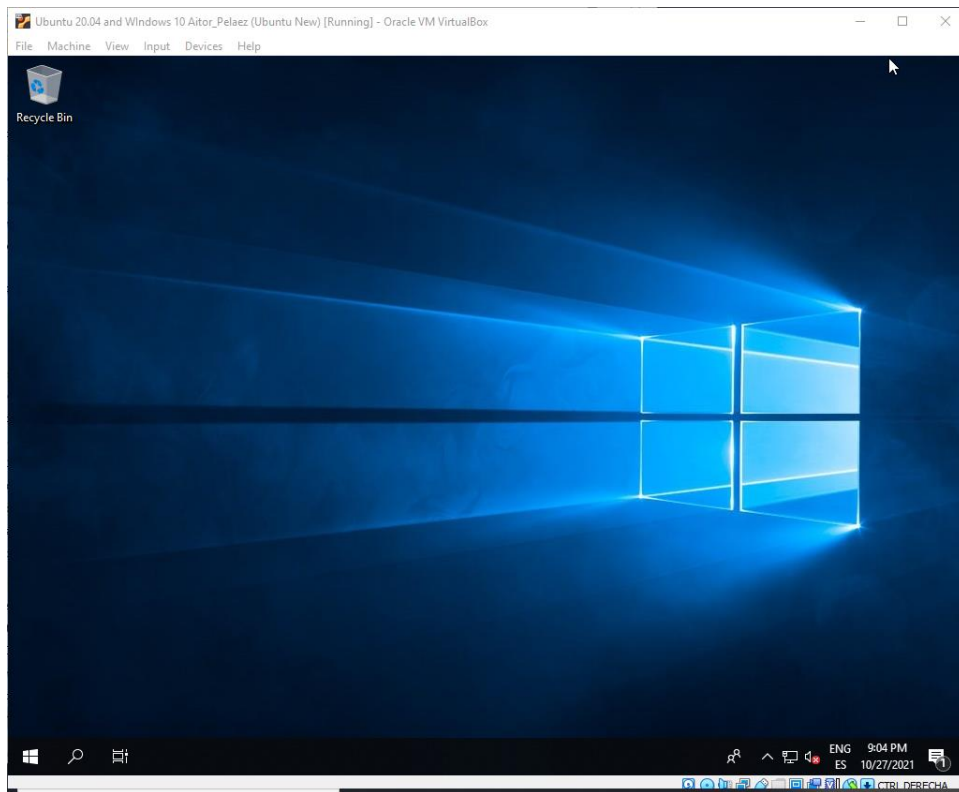
5. And this is how the partition table is organized



Windows partitions are made automatically so i cant explain really in detail those partitions. The only thing I can say is that Windows reserves 500 MB for the system

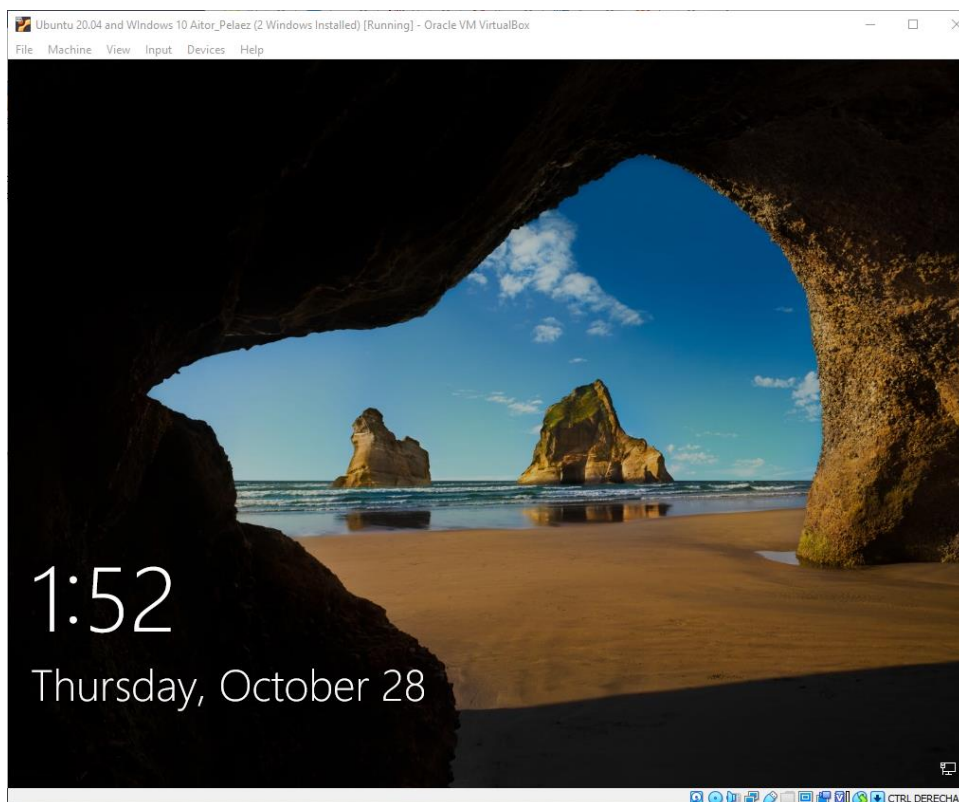


6. Here is Windows installed

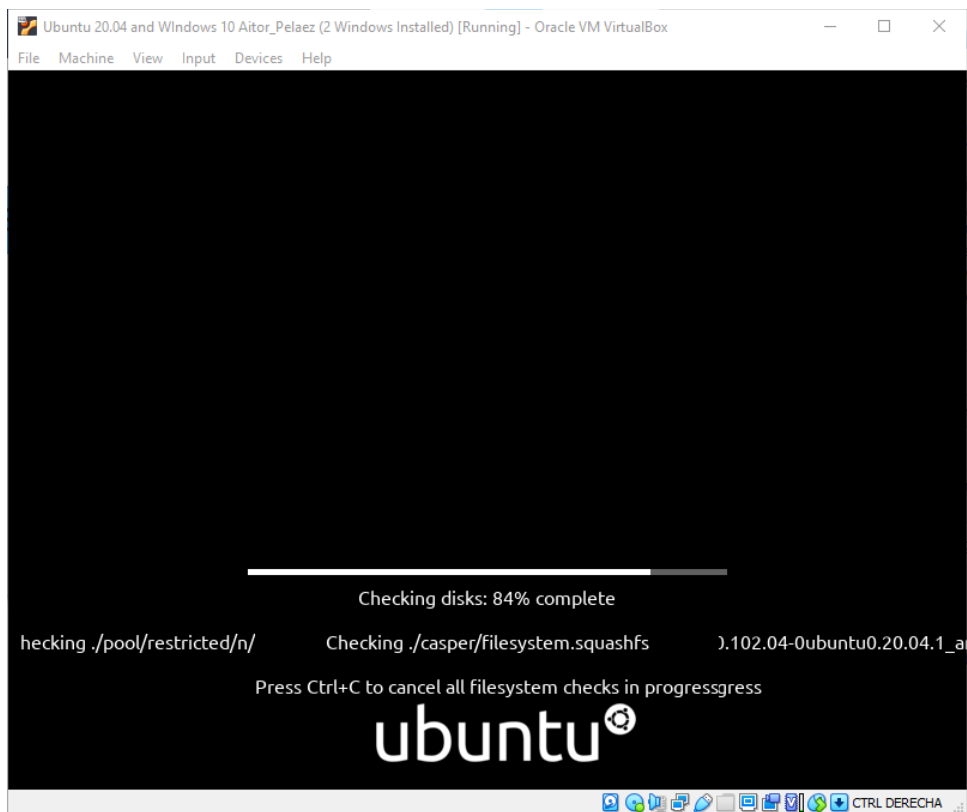


## Recuperation of grub

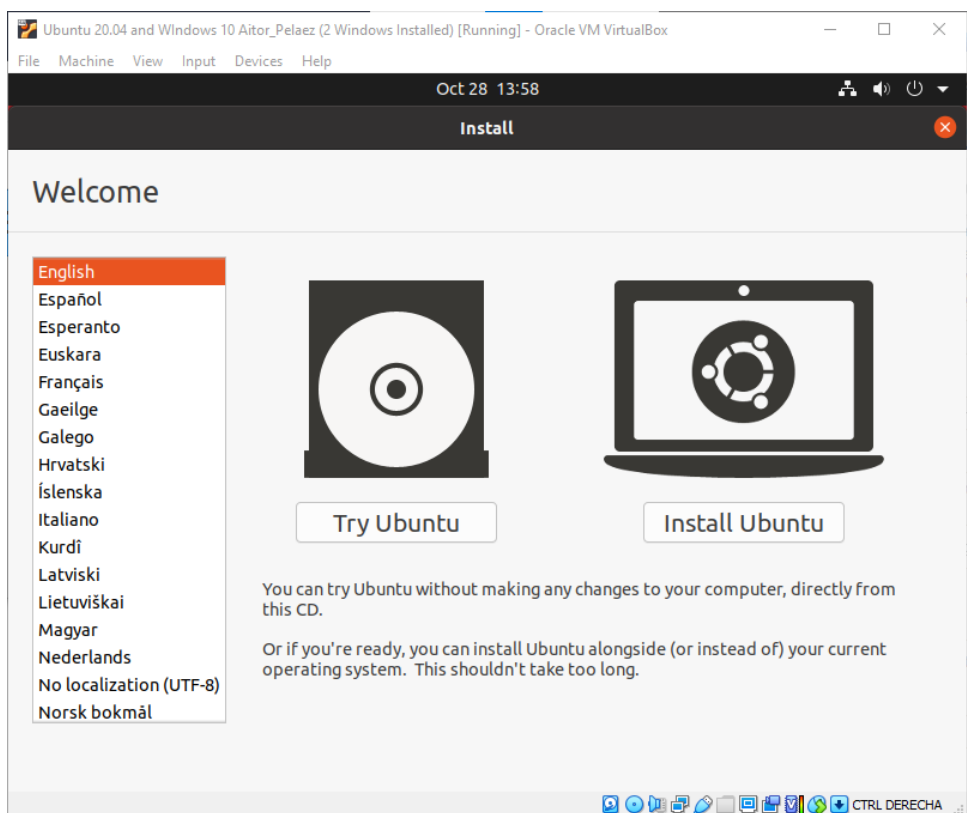
1. The grub doesnt appear and Windows was automatically initiated



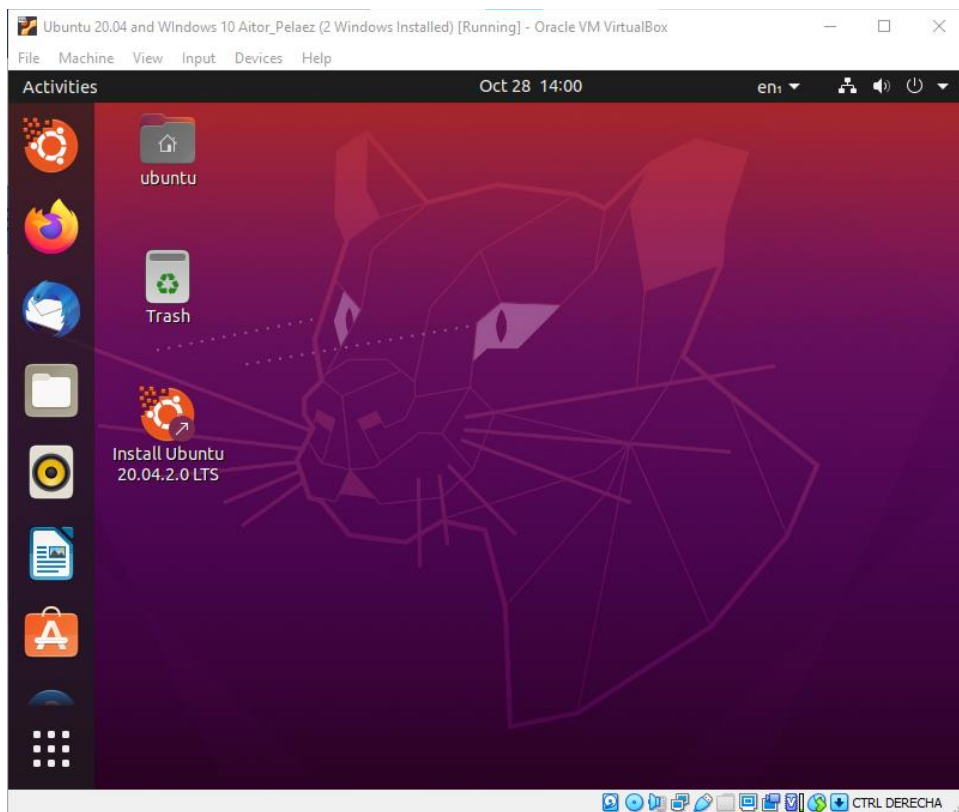
2. Now I insert the Ubuntu 20.04 iso and run it



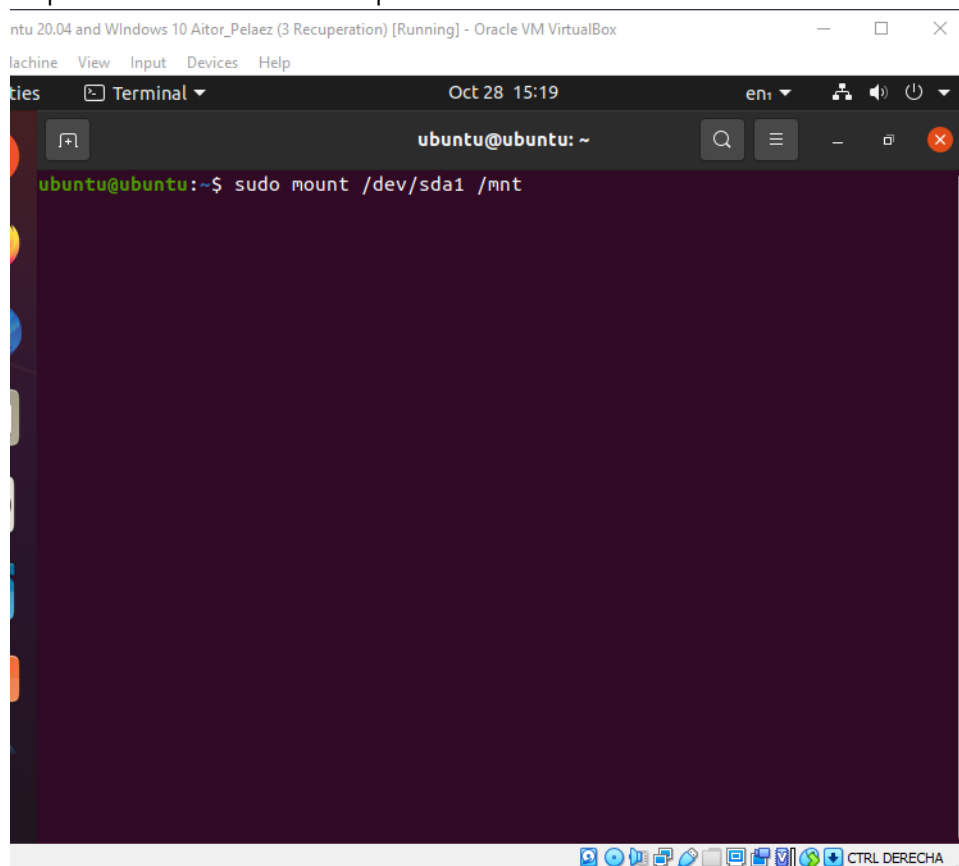
3. Now I choose try Ubuntu

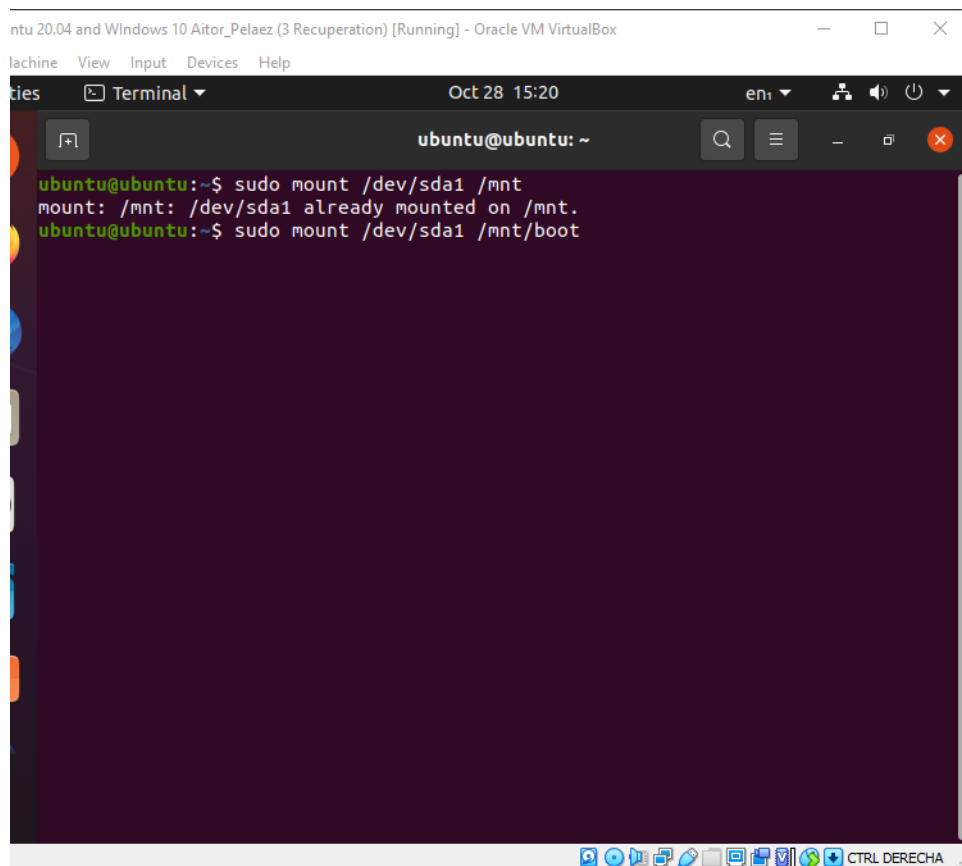


4. Here is try Ubuntu running



5. I open the terminal and I put this command

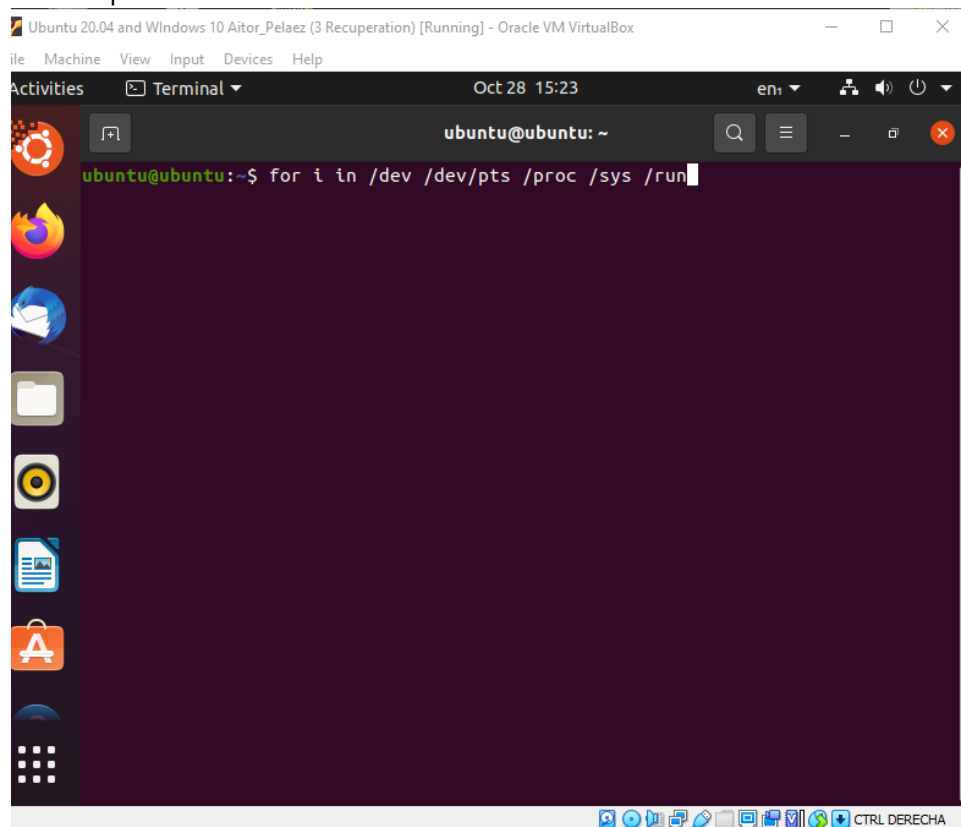




The screenshot shows a terminal window titled "ubuntu 20.04 and Windows 10 Aitor\_Pelaez (3 Recuperation) [Running] - Oracle VM VirtualBox". The terminal prompt is "ubuntu@ubuntu: ~". The user has entered the command "sudo mount /dev/sda1 /mnt", which resulted in the output "mount: /mnt: /dev/sda1 already mounted on /mnt.". The user then entered "sudo mount /dev/sda1 /mnt/boot". The terminal window has a dark purple background and a light gray title bar. The top of the window shows the menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". The bottom of the window shows a taskbar with various icons and the text "CTRL DERECHA".

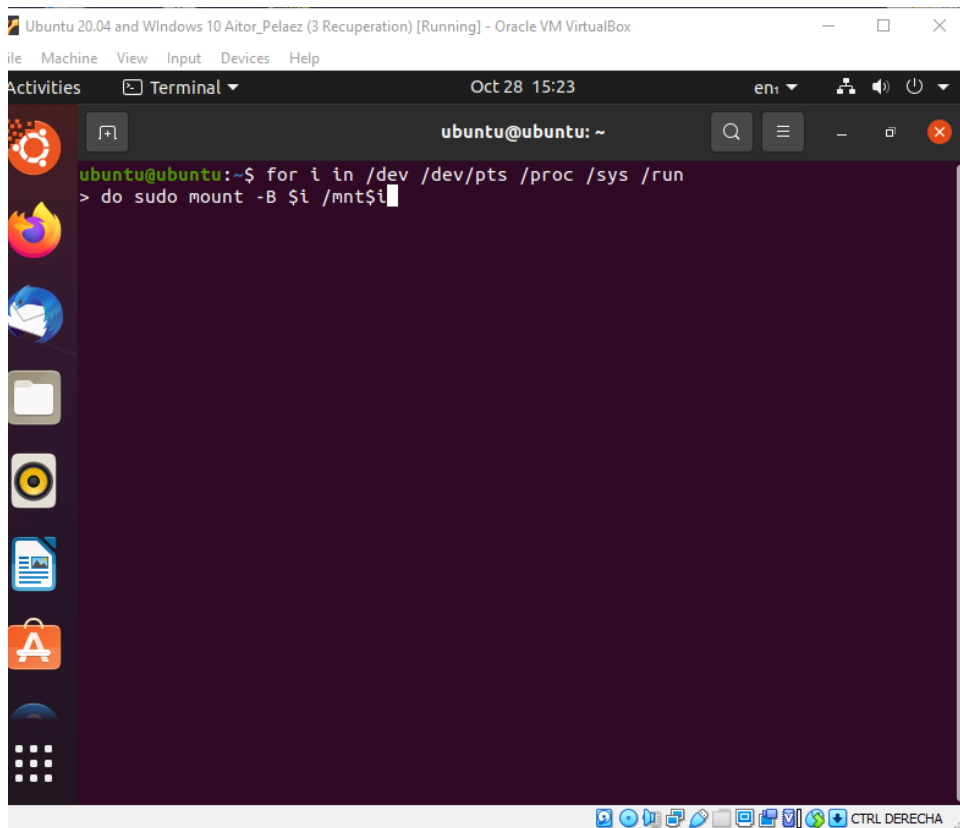
```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ sudo mount /dev/sda1 /mnt  
mount: /mnt: /dev/sda1 already mounted on /mnt.  
ubuntu@ubuntu:~$ sudo mount /dev/sda1 /mnt/boot
```

6. Then I put this other commands



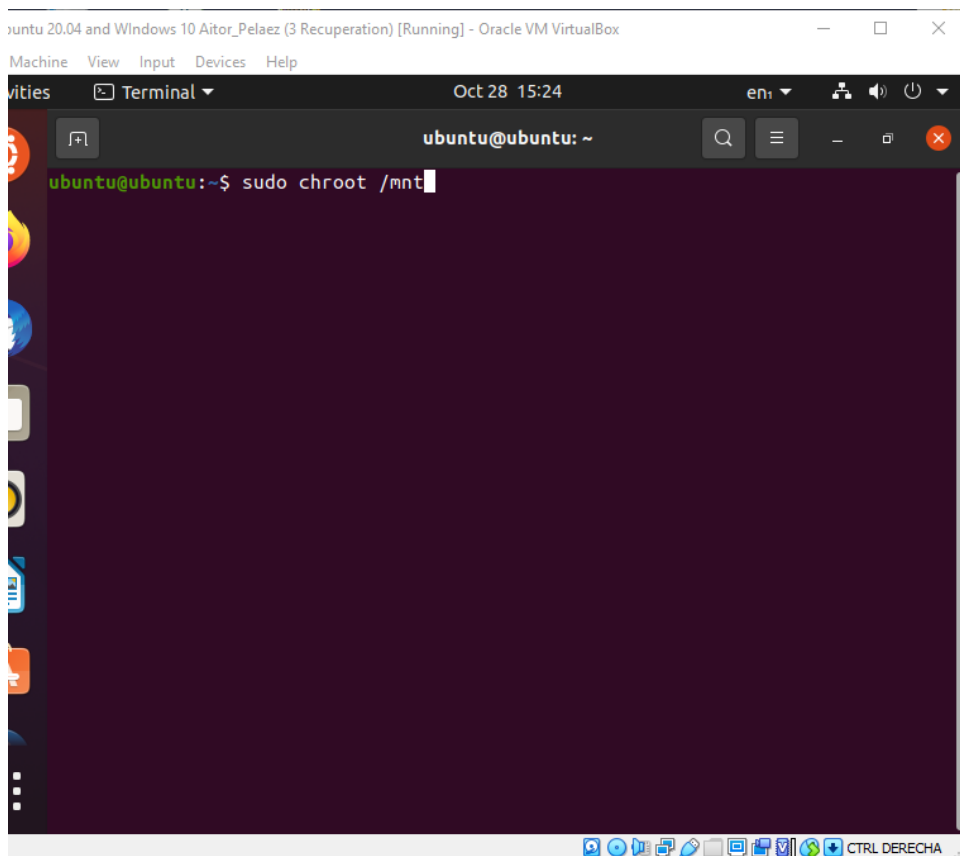
The screenshot shows a terminal window titled "Ubuntu 20.04 and Windows 10 Aitor\_Pelaez (3 Recuperation) [Running] - Oracle VM VirtualBox". The terminal prompt is "ubuntu@ubuntu: ~". The user has entered the command "for i in /dev /dev/pts /proc /sys /run". The terminal window has a dark purple background and a light gray title bar. The top of the window shows the menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". The bottom of the window shows a taskbar with various icons and the text "CTRL DERECHA".

```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ for i in /dev /dev/pts /proc /sys /run
```

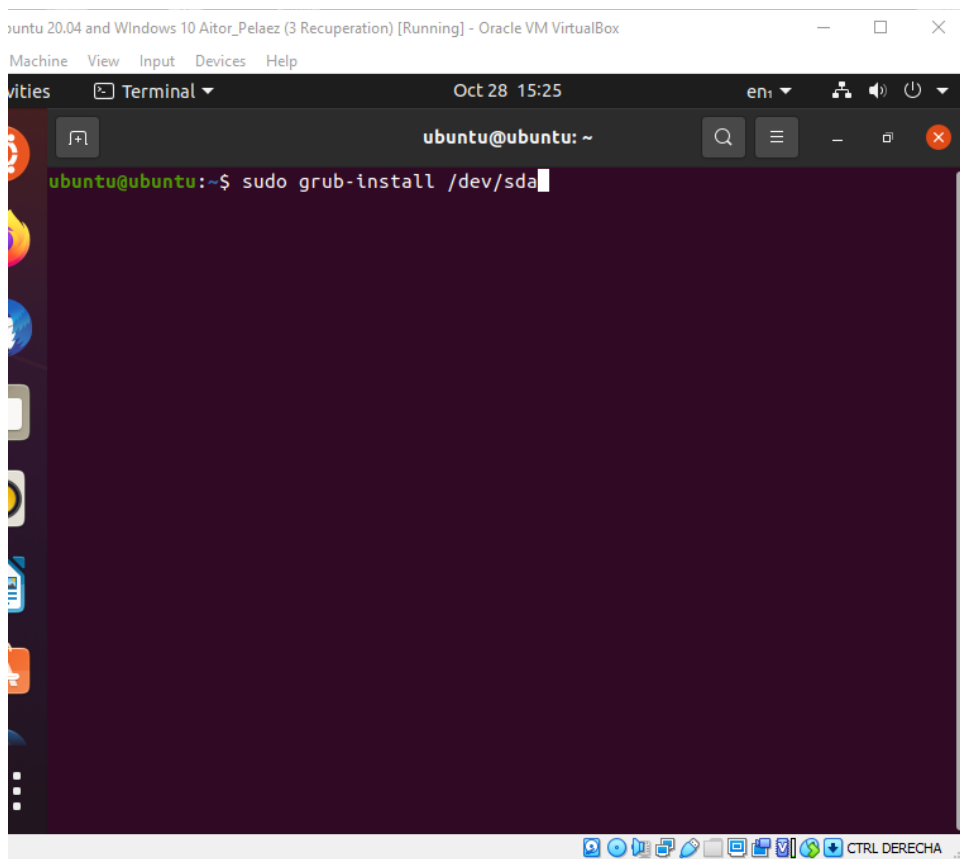


After this command put done, I forgot to do the screenshot

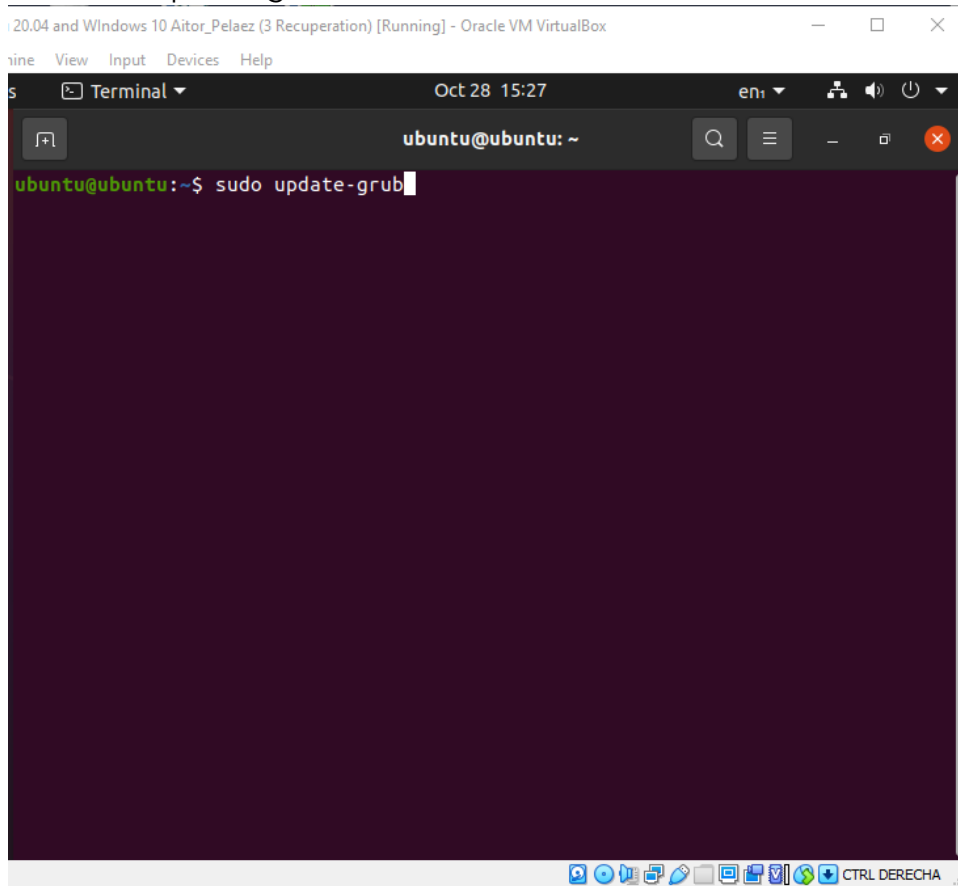
7. Then I put this command



8. Then this other command

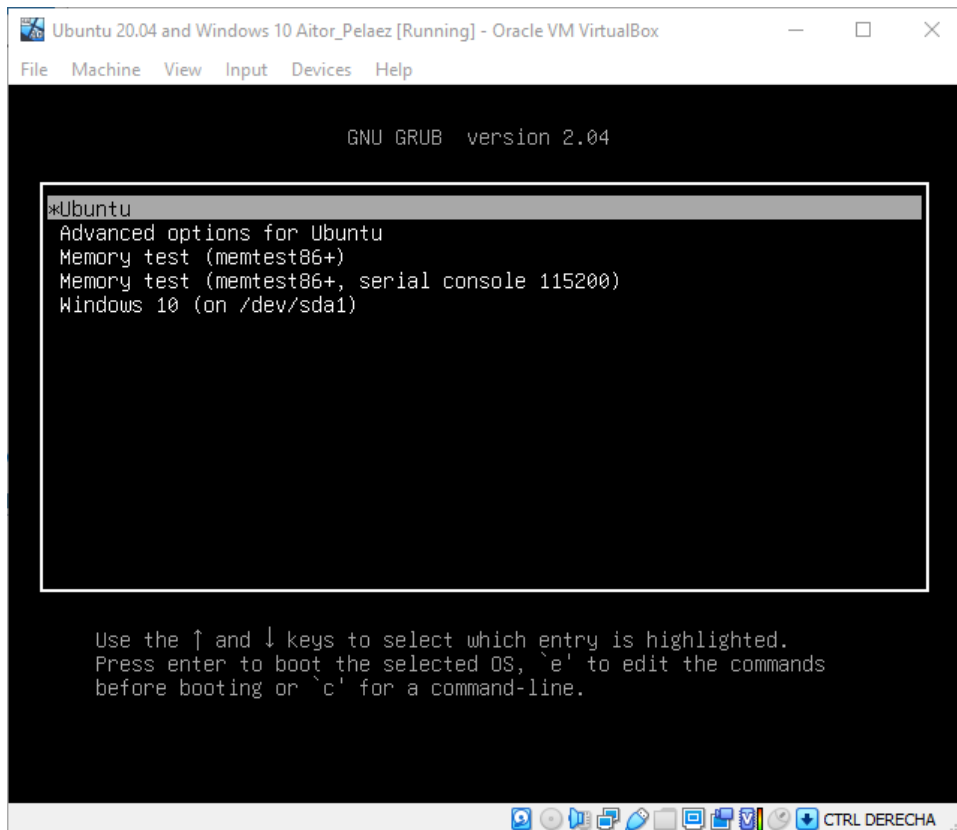


## 9. And then update-grub



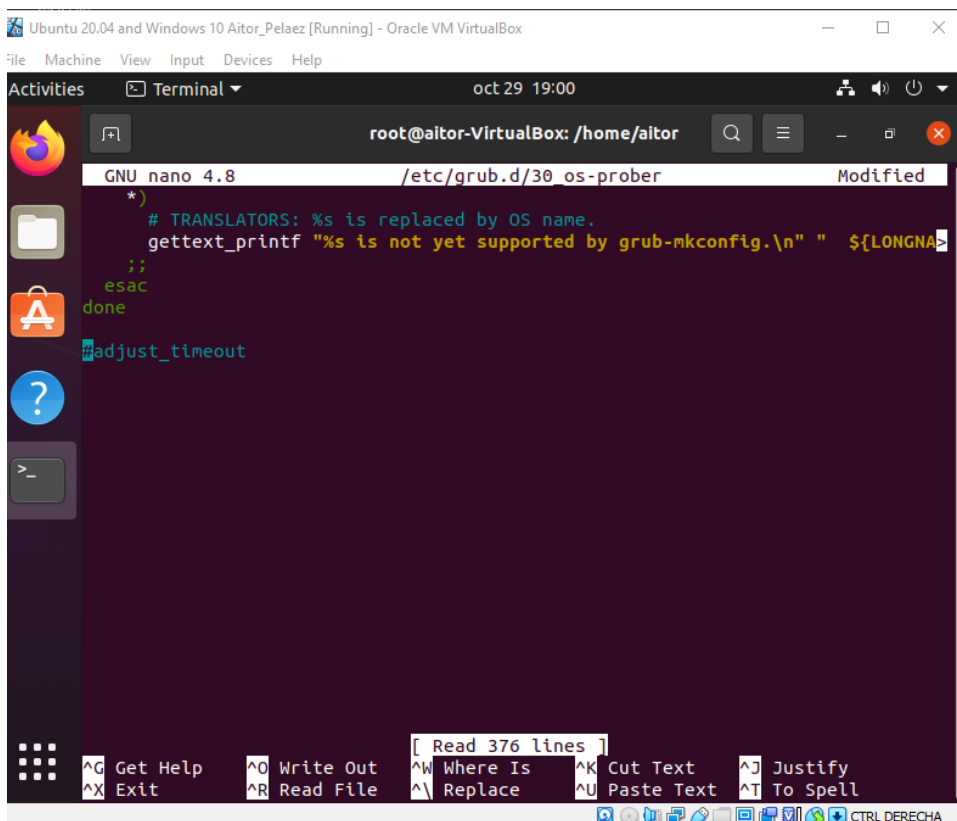
Now restart the VM. IS seen like this becuse I do a snapshot and none of the links worked.



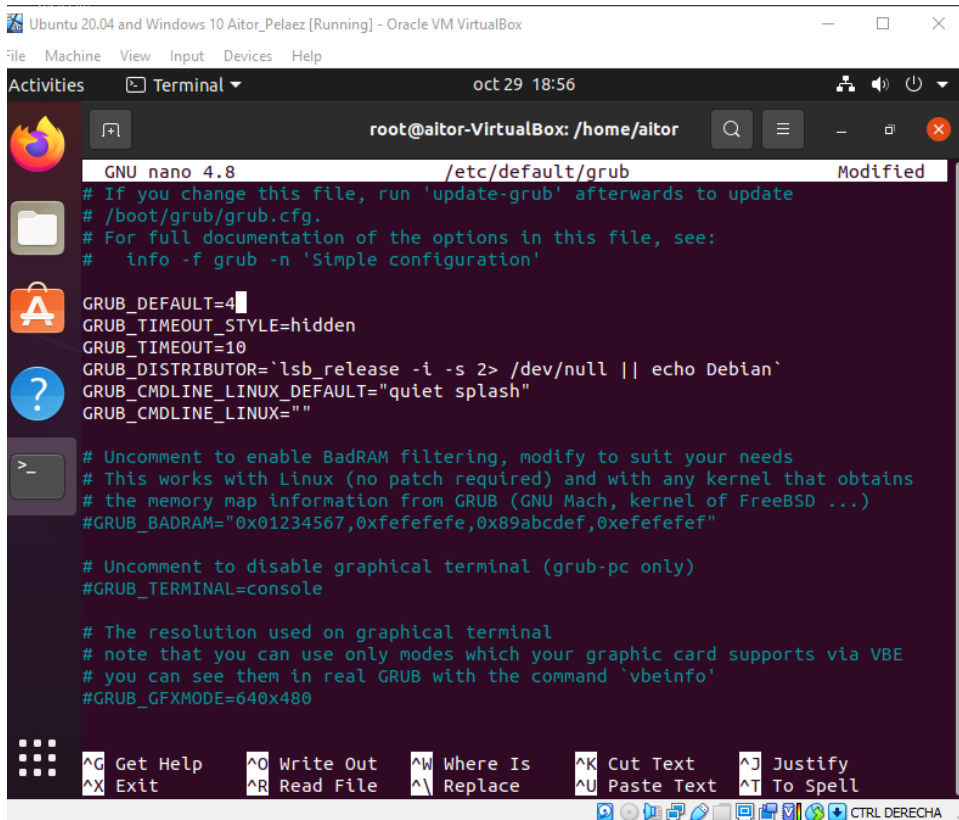


## Exercises

- **No menu displayed by default.**



- **Windows as default OS.**



```
GNU nano 4.8 /etc/default/grub Modified
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
# info -f grub -n 'Simple configuration'

GRUB_DEFAULT=4
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

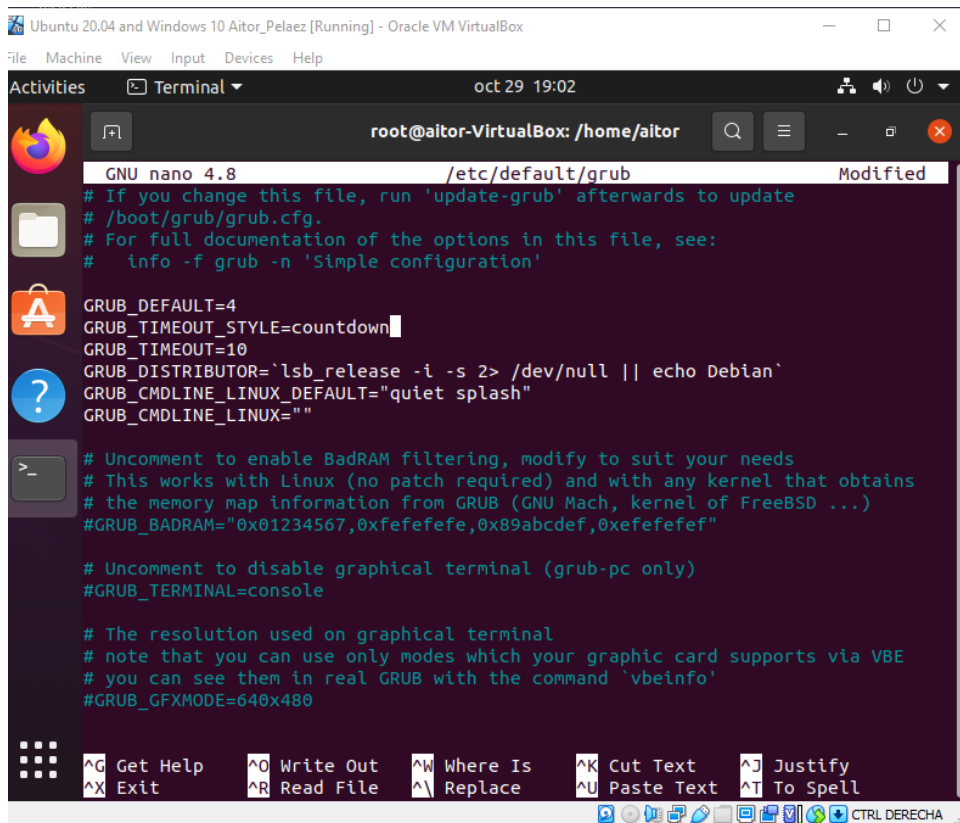
# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command 'vbeinfo'
#GRUB_GFXMODE=640x480

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit          ^R Read File    ^\ Replace      ^U Paste Text   ^T To Spell
```

- **Windows boots automatically after 10 seconds. You must display a countdown for that purpose.**



The screenshot shows a terminal window titled "Ubuntu 20.04 and Windows 10 Aitor\_Pelaez [Running] - Oracle VM VirtualBox". The terminal is running the nano text editor, editing the file `/etc/default/grub`. The user is `root` at `aitor-VirtualBox` in the directory `/home/aitor`. The date and time are `oct 29 19:02`. The terminal content is as follows:

```
GNU nano 4.8 /etc/default/grub Modified
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=4
GRUB_TIMEOUT_STYLE=countdown
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

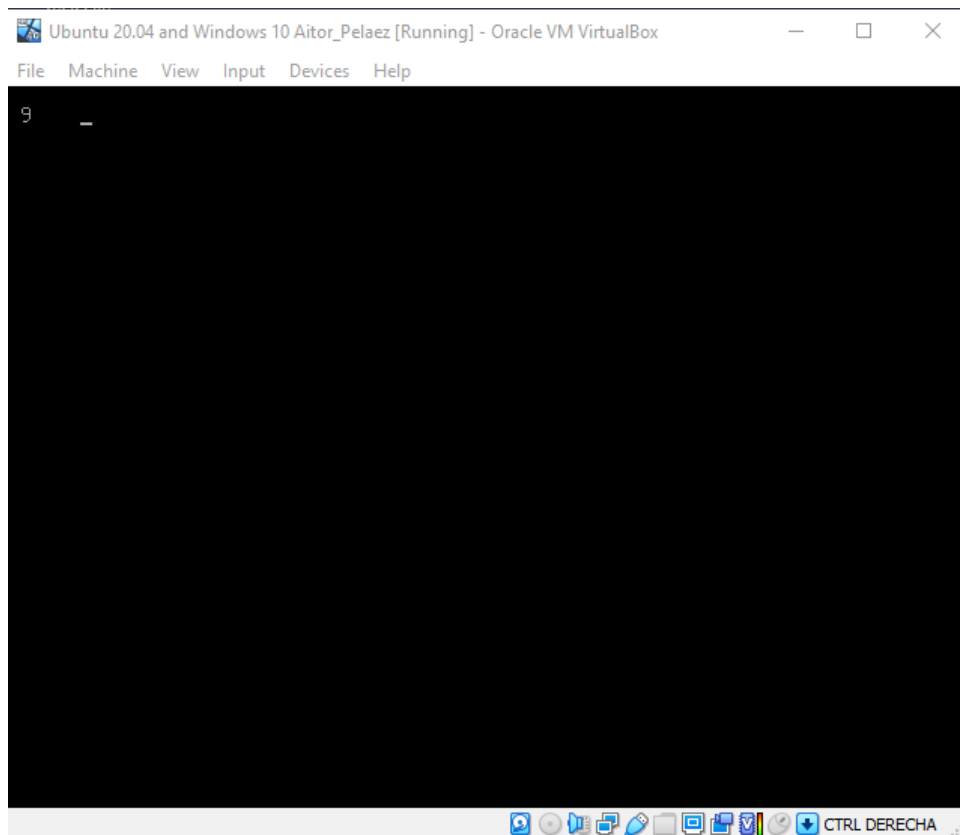
# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text      ^J Justify
^X Exit          ^R Read File    ^_ Replace      ^U Paste Text    ^T To Spell
```

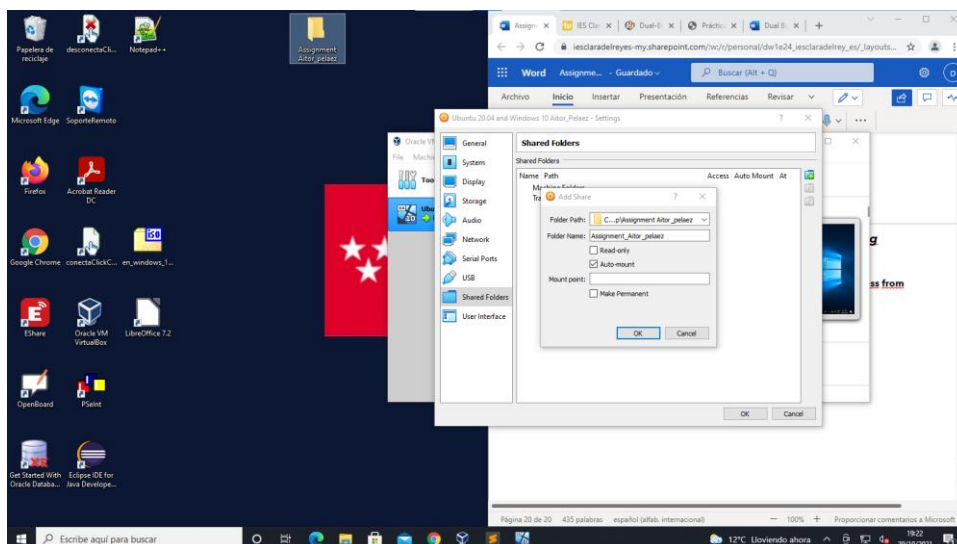
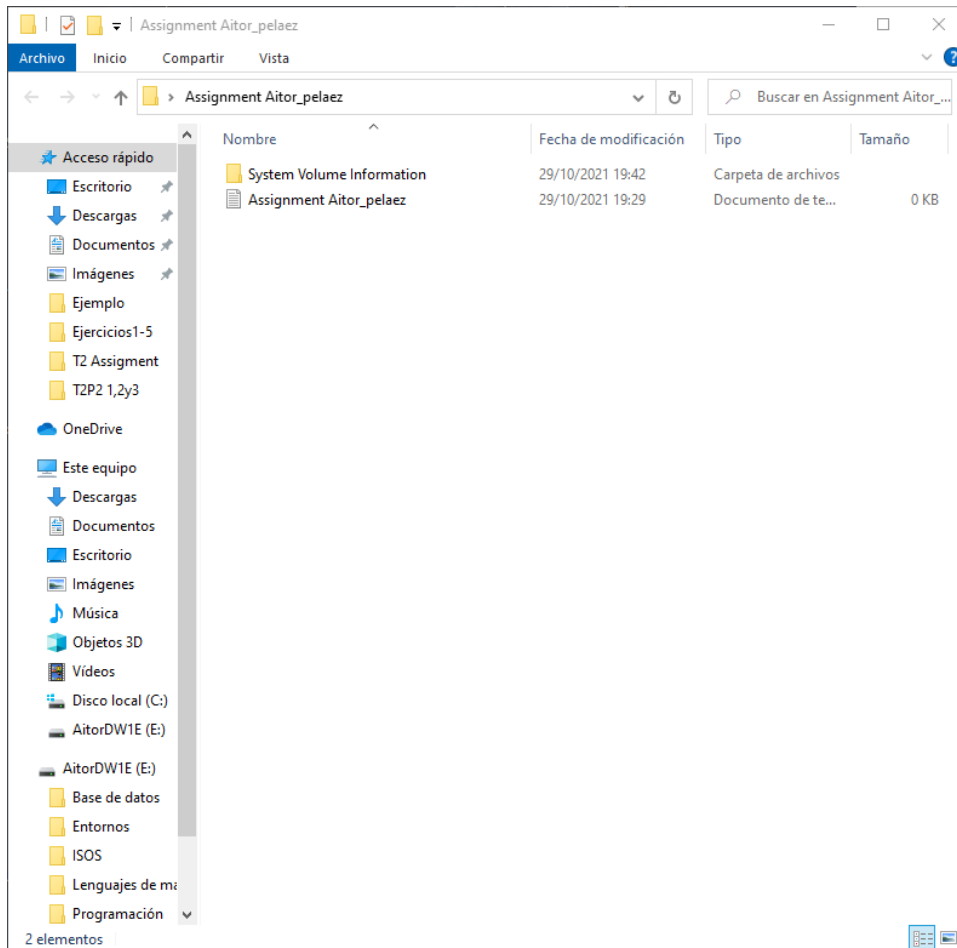
## DEMONSTRATION

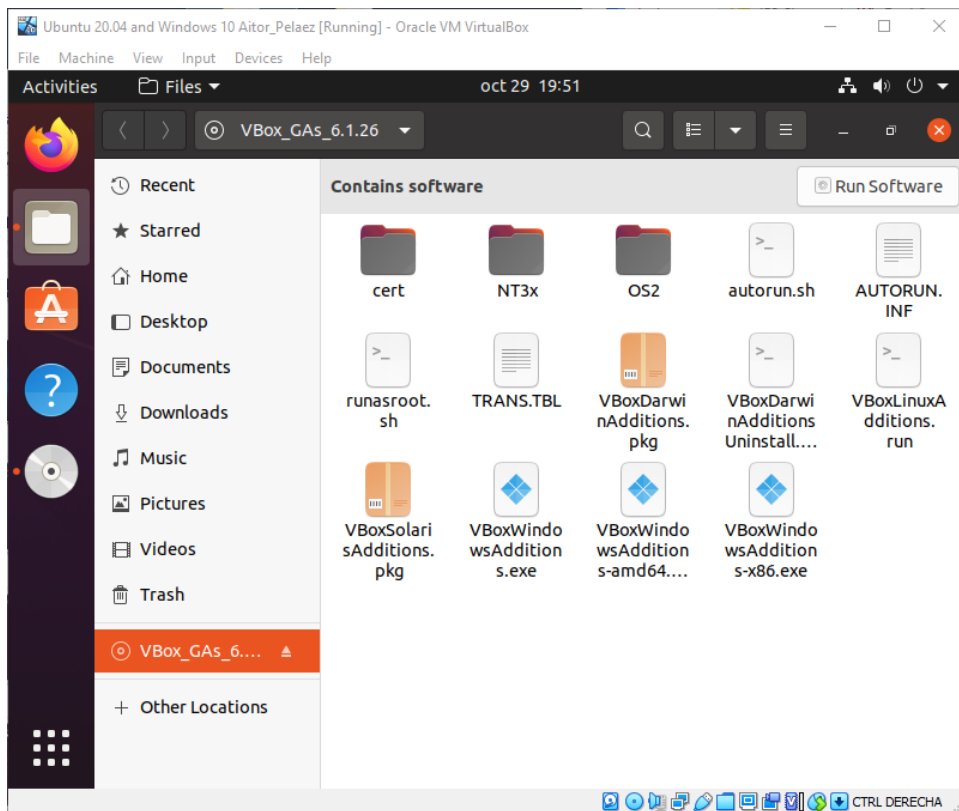
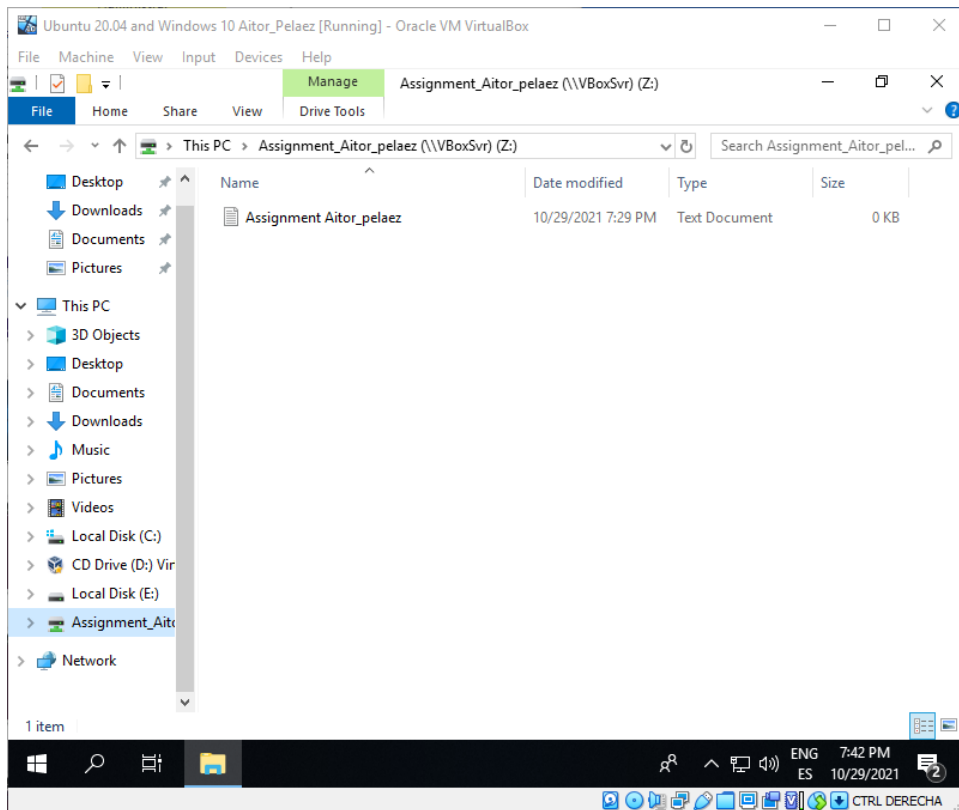


## ***Finally, the virtual machine should meet the following requirements:***

- **A shared folder to an external storage device that you can access from both operating systems.**

We need to install guest additions for doing this





- **Internet connection**

