



Recuperación Iso

Cristian Oliva Llanos

Implantación de sistemas operativos

ASIR

10/11/2023

ÍNDICE

1. PROGRAMACIÓN PYTHON.....	3
1.1 Código 1.....	3
1.2 Código 2.....	5
1.3 Código 3.....	7
1.4 Enunciado 1.....	9
1.5 Enunciado 2.....	9
1.6 Enunciado 3.....	9
1.7 Enunciado 4.....	9
1.8 Enunciado 5.....	9
2. LINUX.....	10
2.1 Supuesto práctico.....	10
2.2 Comandos LINUX.....	10
2.2.1 Ejercicio 1.....	10
2.2.2 Ejercicio 2.....	10
2.2.3 Ejercicio 3.....	10
3. WINDOWS.....	11
3.1 Supuesto práctico.....	11
4. GESTIÓN DE RECURSOS Y OTROS.....	12
4.1 Comandos para la gestión de procesos y memoria.....	12
4.2 Licencias 1.....	14
4.3 Licencias 2.....	15
5. VIRTUALIZACIÓN.....	16
5.1 Virtualización dual.....	16
5.2 Red NAT.....	16

1. PROGRAMACIÓN PYTHON

1.1 Código 1

Realiza un programa en el que pida al usuario tres números distintos y de los tres que muestre el mayor, para finalizar el programa presione la tecla 0.

Implantación y solución en python

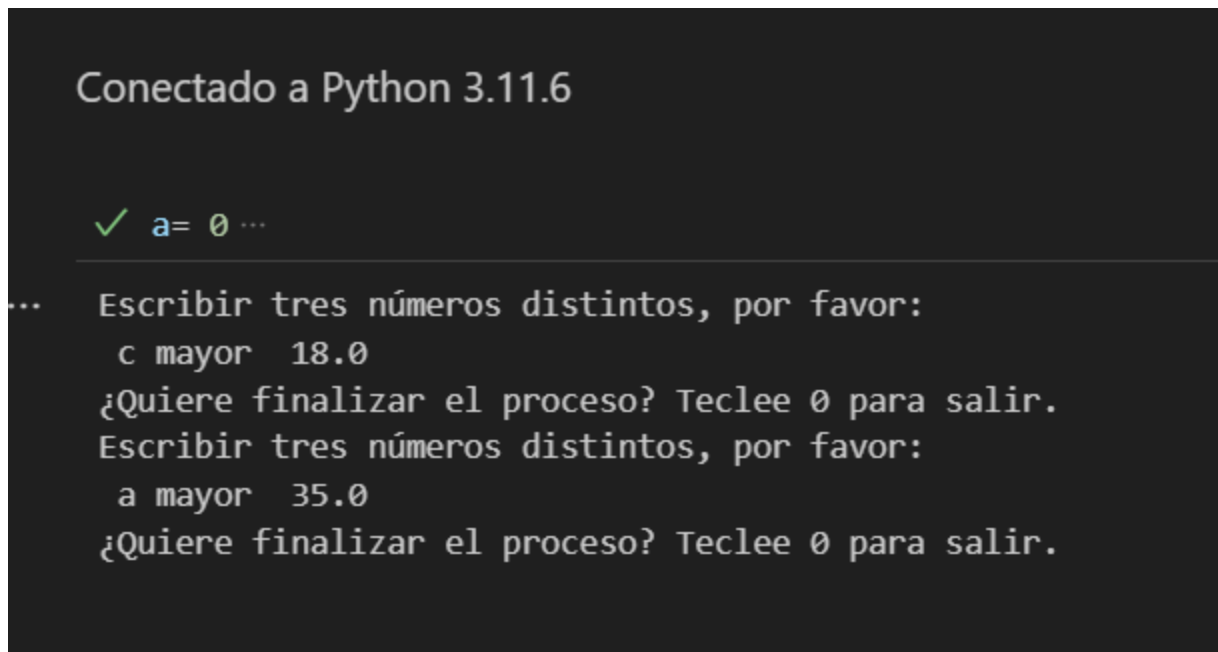
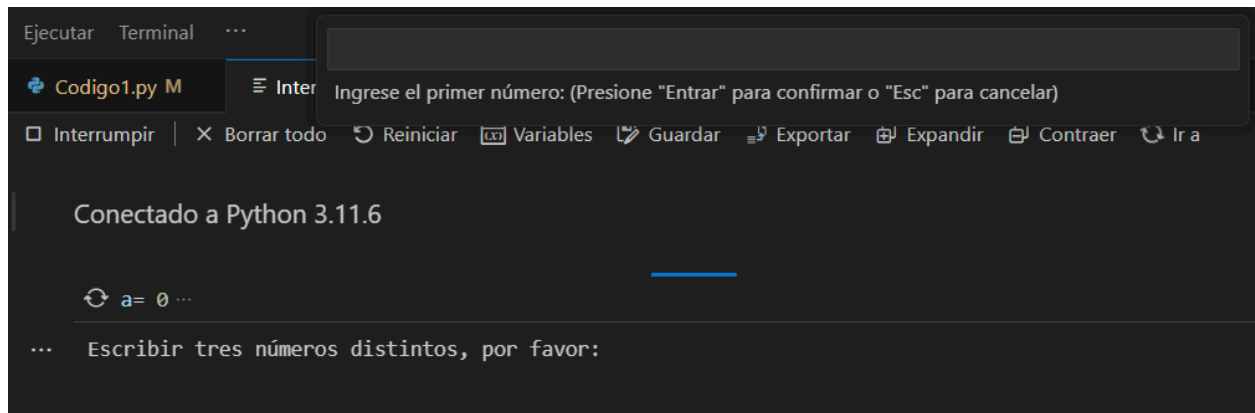
Código dado para realizar el ejercicio.

```
a= 0
b= 0
c= 0
final= 1

while final != 0:
    print("Escribir tres números distintos, por favor: ")
    a= float(input("Ingrese el primer número: "))
    b= float(input("Ingrese el segundo número: "))
    c= float(input("Ingrese el tercer número: "))

    if a>b:
        if a>c:
            print(" a mayor ", a)
        else:
            print(" c mayor ", c)
    else:
        if b>c:
            print(" b mayor ", b)
        else:
            print(" c mayor ", c)

    print( "¿Quiere finalizar el proceso? Teclee 0 para salir.")
    final= int(input())
```



1.2 Código 2

Realice un programa en el que se le pregunte al usuario si quiere realizar o no el programa si lo quiere realizar realice un bucle en el que se le vaya pidiendo al usuario números y al llegar a 10 números salga del bucle.

Implantación y solución en python

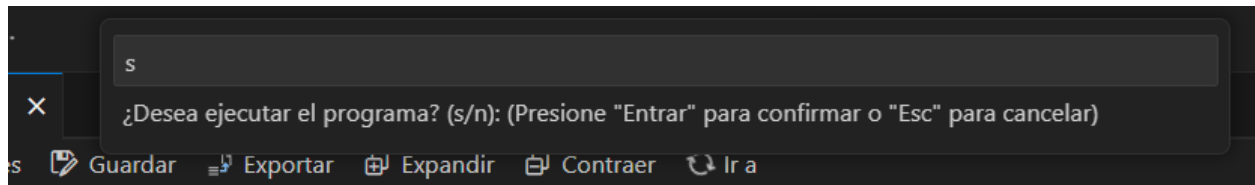
Código dado para realizar el ejercicio.

```
def un_proceso():
    a= 0
    r= 0
    c= 0
    pr= input("¿Desea ejecutar el programa? (s/n): ")

    if pr == "s":
        c = 0
        while c < 10:
            a = int(input("Ingrese un número: "))
            c += 1
            r = a % 2
            if r == 0:
                print(str(a) + 'seleccionado')
            else:
                print(str(a) + 'no seleccionado')
            print(r)
        print("ya terminamos")

un_proceso()
```

Si el usuario quiere realizar el programa.



```
1 no seleccionado
1
2 seleccionado
0
3 no seleccionado
1
4 seleccionado
0
5 no seleccionado
1
6 seleccionado
0
7 no seleccionado
1
8 seleccionado
0
9 no seleccionado
1
10 seleccionado
0
ya terminamos
```

Si el usuario no quiere realizar el programa.

```
n|
¿Desea ejecutar el programa? (s/n): (Presione "Entrar" para confirmar o "Esc" para cancelar)
```

```
Conectado a Python 3.11.6

✓ def un_proceso(): ...
... ya terminamos
```

1.3 Código 3

Realiza un proceso en el que te muestre la tabla de multiplicar y para finalizar pulse la tecla 0.

Implantación y solución en python

Código dado para realizar el ejercicio.

```
def tabla_multiplicar():
    i = 1
    j = 1
    while i != 0:
        i = int(input("Escribe un número. Pulsa 0 para finalizar: "))
        if i != 0:
            j = 1
            while j < 11:
                r = i * j
                print(f"{i} x {j} = {r}")
                j += 1
    tabla_multiplicar()
```

8

Escribe un número. Pulsa 0 para finalizar: (Presione "Entrar" para confirmar o "Esc" para cancelar)

5) Reiniciar Variables Guardar Exportar Expandir Centrar (X) Ir a

0

Escribe un número. Pulsa 0 para finalizar: (Presione "Entrar" para confirmar o "Esc" para cancelar)

Conectado a Python 3.11.6

✓ `def tabla_multiplicar():`

```
.. 8 x 1 = 8
    8 x 2 = 16
    8 x 3 = 24
    8 x 4 = 32
    8 x 5 = 40
    8 x 6 = 48
    8 x 7 = 56
    8 x 8 = 64
    8 x 9 = 72
    8 x 10 = 80
```


1.4 Enunciado 1

https://github.com/colival03/Oliva_Cristian_ReclSO/blob/ea1ce8d8e16f9474b7fe5ccabd589984b4296665/Python/Enunciado1.py

1.5 Enunciado 2

https://github.com/colival03/Oliva_Cristian_ReclSO/blob/ea1ce8d8e16f9474b7fe5ccabd589984b4296665/Python/Enunciado2.py

1.6 Enunciado 3

https://github.com/colival03/Oliva_Cristian_ReclSO/blob/ea1ce8d8e16f9474b7fe5ccabd589984b4296665/Python/Enunciado3.py

1.7 Enunciado 4

https://github.com/colival03/Oliva_Cristian_ReclSO/blob/ea1ce8d8e16f9474b7fe5ccabd589984b4296665/Python/Enunciado4.py

1.8 Enunciado 5

https://github.com/colival03/Oliva_Cristian_ReclSO/blob/ea1ce8d8e16f9474b7fe5ccabd589984b4296665/Python/Enunciado5.py

2. LINUX

2.1 Supuesto práctico

https://github.com/colival03/Oliva_Cristian_ReclSO/blob/main/Supuesto%20práctico%20linux/Supuesto%20práctico%20de%20linux.mp4

2.2 Comandos LINUX

2.2.1 Ejercicio 1

https://github.com/colival03/Oliva_Cristian_ReclSO/blob/main/Linux/Ejercicio1

2.2.2 Ejercicio 2

https://github.com/colival03/Oliva_Cristian_ReclSO/blob/main/Linux/Ejercicio2

2.2.3 Ejercicio 3

https://github.com/colival03/Oliva_Cristian_ReclSO/blob/main/Linux/Ejercicio3

He creado el directorio "prueba" al quitarle los permisos de ejecución (x) e intentar entrar en él, con el comando "cd" no permite el acceso.

```

cristianrequiso@cristianrequiso-VirtualBox:~$ mkdir prueba
cristianrequiso@cristianrequiso-VirtualBox:~$ ls
Descargas Documentos Escritorio Imágenes Música Plantillas prueba Público
cristianrequiso@cristianrequiso-VirtualBox:~$ ls -l
total 40
drwxr-xr-x 2 cristianrequiso cristianrequiso 4096 nov 18 18:47 Descargas
drwxr-xr-x 3 cristianrequiso cristianrequiso 4096 dic 19 11:56 Documentos
drwxr-xr-x 2 cristianrequiso cristianrequiso 4096 nov 18 18:47 Escritorio
drwxr-xr-x 2 cristianrequiso cristianrequiso 4096 nov 18 18:47 Imágenes
drwxr-xr-x 2 cristianrequiso cristianrequiso 4096 nov 18 18:47 Música
drwxr-xr-x 2 cristianrequiso cristianrequiso 4096 nov 18 18:47 Plantillas
drwxrwxr-x 2 cristianrequiso cristianrequiso 4096 dic 19 21:04 prueba
drwxr-xr-x 2 cristianrequiso cristianrequiso 4096 nov 18 18:47 Público
drwx----- 3 cristianrequiso cristianrequiso 4096 nov 18 18:47 snap
drwxr-xr-x 2 cristianrequiso cristianrequiso 4096 nov 18 18:47 Videos
cristianrequiso@cristianrequiso-VirtualBox:~$ chmod 666 prueba/
cristianrequiso@cristianrequiso-VirtualBox:~$ ls -l
total 40
drwxr-xr-x 2 cristianrequiso cristianrequiso 4096 nov 18 18:47 Descargas
drwxr-xr-x 3 cristianrequiso cristianrequiso 4096 dic 19 11:56 Documentos
drwxr-xr-x 2 cristianrequiso cristianrequiso 4096 nov 18 18:47 Escritorio
drwxr-xr-x 2 cristianrequiso cristianrequiso 4096 nov 18 18:47 Imágenes
drwxr-xr-x 2 cristianrequiso cristianrequiso 4096 nov 18 18:47 Música
drwxr-xr-x 2 cristianrequiso cristianrequiso 4096 nov 18 18:47 Plantillas
drw-rw-rw- 2 cristianrequiso cristianrequiso 4096 dic 19 21:04 prueba
drwxr-xr-x 2 cristianrequiso cristianrequiso 4096 nov 18 18:47 Público
drwx----- 3 cristianrequiso cristianrequiso 4096 nov 18 18:47 snap
drwxr-xr-x 2 cristianrequiso cristianrequiso 4096 nov 18 18:47 Videos
cristianrequiso@cristianrequiso-VirtualBox:~$ cd prueba/
bash: cd: prueba/: Permiso denegado
cristianrequiso@cristianrequiso-VirtualBox:~$ █

```

Los permisos rwx para los directorios significan lo siguiente:

- r: da u otorga permisos de lectura al directorio.
- w: da u otorga permisos de escritura en el directorio.
- x: da u otorga permisos de ejecución al directorio.

3. WINDOWS

3.1 Supuesto práctico

https://github.com/colival03/Oliva_Cristian_ReclSO/blob/main/Supuesto%20pr%C3%A1ctico%20windows.zip

4. GESTIÓN DE RECURSOS Y OTROS

4.1 Comandos para la gestión de procesos y memoria

Ps: Este sirve para mostrar los procesos activos.

```
cristianrequiso@cristianrequiso-VirtualBox:~$ ps
  PID TTY          TIME CMD
 3118 pts/0    00:00:00 bash
 3129 pts/0    00:00:00 ps
cristianrequiso@cristianrequiso-VirtualBox:~$ S
```

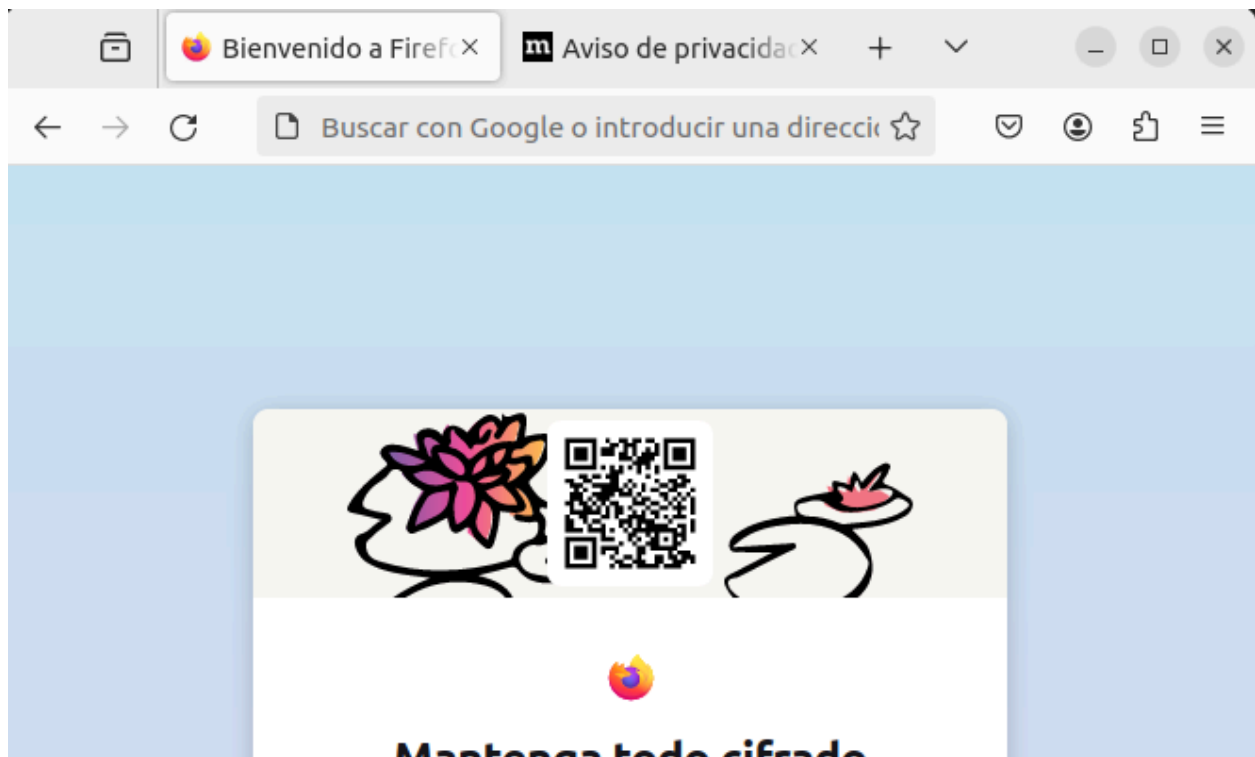
top: Este comando muestra una lista de los procesos en ejecución y su uso de recursos tanto del usuario como el root.

```
top - 20:15:15 up 9 min,  1 user,  load average: 0,09, 0,32, 0,22
Tareas: 186 total,  2 ejecutar, 184 hibernar,  0 detener,  0 zombie
%Cpu(s):  0,0 us,  1,1 sy,  0,0 ni, 98,9 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem : 1959,7 total,  134,5 libre,  798,5 usado,  1026,7 búfer/caché
MiB Intercambio: 2680,0 total,  2638,6 libre,  41,4 usado.  979,3 dispon Mem
```

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
2583	cristia+	20	0	4049328	320232	113700	S	3,0	16,0	0:07.80	gnome-shell
378	systemd+	20	0	14828	6912	6144	S	0,7	0,3	0:00.62	systemd-oomd
3100	cristia+	20	0	566656	52116	39156	S	0,7	2,6	0:00.51	gnome-terminal-
40	root	0	-20	0	0	0	I	0,3	0,0	0:01.17	kworker/0:1H-kblockd
3027	cristia+	20	0	221476	3200	2816	S	0,3	0,2	0:01.77	VBoxClient

kill: Este comando permite acabar con el proceso.

Con la opción killall: puedes acabar con el proceso por su nombre.



```
cristianrequiso@cristianrequiso-VirtualBox:~$ killall firefox
cristianrequiso@cristianrequiso-VirtualBox:~$
```

htop: Este comando realiza algo parecido al “**top**”, pero al mostrar la lista te la muestra de manera más fácil de interpretar.

```
CPU[||||| 4.4%] Tasks: 123, 279 thr; 1 running
Mem[||||| 802M/1.91G] Load average: 0.22 0.17 0.18
Swp[| 105M/2.62G] Uptime: 00:19:00
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
2583	cristianr	20	0	3961M	262M	71868	S	8.8	13.4	0:25.88	/usr/bin/gnome-shell
2627	cristianr	20	0	3961M	262M	71868	S	0.7	13.4	0:00.61	/usr/bin/gnome-shell
5679	cristianr	20	0	13932	4864	3584	R	0.7	0.2	0:00.03	htop
1	root	20	0	163M	12624	7632	S	0.0	0.6	0:01.14	/sbin/init splash
179	root	19	-1	64740	14504	12968	S	0.0	0.7	0:00.32	/lib/systemd/systemd-journald
210	root	20	0	27056	6876	4572	S	0.0	0.3	0:00.11	/lib/systemd/systemd-udev
378	systemd-o	20	0	14828	6528	5760	S	0.0	0.3	0:01.47	/lib/systemd/systemd-oomd
381	systemd-r	20	0	25928	12256	7936	S	0.0	0.6	0:00.25	/lib/systemd/systemd-resolved
476	systemd-t	20	0	89380	7040	6144	S	0.0	0.4	0:00.04	/lib/systemd/systemd-timesyncd
493	systemd-t	20	0	89380	7040	6144	S	0.0	0.4	0:00.00	/lib/systemd/systemd-timesyncd
528	root	20	0	237M	7472	6576	S	0.0	0.4	0:00.12	/usr/libexec/accounts-daemon
529	root	20	0	2812	1920	1792	S	0.0	0.1	0:00.02	/usr/sbin/acpid
532	avahi	20	0	7624	3968	3584	S	0.0	0.2	0:00.05	avahi-daemon: running [cristianr
534	root	20	0	12108	2816	2688	S	0.0	0.1	0:00.00	/usr/sbin/cron -f -P

free: Este comando muestra simplemente la memoria disponible y utilizada en el sistema.

```
cristianrecuiso@cristianrecuiso-VirtualBox:~$ free
              total        usado       libre   compartido    búf/caché    disponible
Mem:          2006696      808028      138284         13716      1060384      1011072
Inter:         2744316      107476      2636840
cristianrecuiso@cristianrecuiso-VirtualBox:~$
```

vmstat: Muestra las estadísticas de la memoria del sistema.

```
cristianrecuiso@cristianrecuiso-VirtualBox:~$ vmstat
procs -----memoria----- ---swap-- -----io----- -sistema-- -----cpu-----
r  b   swpd  libre  búf caché   si   so    bi    bo   in   cs us sy id wa st
0  0  107476 138284  24672 1036188    3   93  2065   634  560 1139  3  6 91  1  0
cristianrecuiso@cristianrecuiso-VirtualBox:~$
```

nice: Con este comando puedes cambiar la prioridad de ejecución de un proceso.

```
cristianrecuiso@cristianrecuiso-VirtualBox:~$ nice
0
cristianrecuiso@cristianrecuiso-VirtualBox:~$
```

4.2 Licencias 1

OEM: Estas licencias están diseñadas para ser preinstaladas durante la fabricación de tu equipo, solo son propias de cada equipo. Por ejemplo, un fabricante de ordenadores puede fabricarlas antes de venderlas y el sistema operativo está asociado a ese equipo y no puede transferirlo.

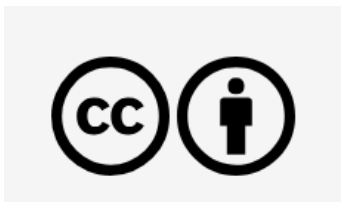
Retail: Estas licencias permiten la instalación del software en una máquina y que permite transferir la licencia a otro equipo, aunque con algunas restricciones. Por ejemplo, si un usuario compra la licencia lo puede instalar en su ordenador, pero si decide transferir sigue con restricciones generadas por la licencia.

Volumen: Estas se utilizan para instalar en múltiples ordenadores y estas sí se pueden transferir a otros dispositivos. Por ejemplo, una empresa que contrata esta licencia para instalar en todos los ordenadores de la empresa y así mejorar la productividad.

4.3 Licencias 2

Existen seis tipos de **Creative Commons**, las cuales vamos a pasar a poner una pequeña descripción de cada una de ellas con una imagen respectiva haciendo referencia a ellas:

1. **Atribución (CC BY):** Con esta licencia permite copiar, distribuir, mostrar y ejecutar con fines comerciales o no comerciales, es la más flexible de todas.



2. **Atribución-Compartir Igual (CC BY-SA):** Esta licencia permite mezclar, modificar y construir para cualquier propósito, siempre que se de crédito o se mencione al autor del trabajo.



3. **Atribución-Sin Derivadas (CC BY-ND):** Esta licencia es parecida a la anterior, pero solo puede descargar el trabajo, tiene que dar crédito y NO puede modificar su contenido ni utilizarla.



4. **Atribución-No comercial (CC BY-NC):** Permite realizar todas las tareas con el trabajo para fines no comerciales y se debe de dar créditos al autor.



5. **Atribución-No comercial-Compartir Igual (CC BY-NC-SA):** Permite mezclar, modificar y construir con fines no comerciales, tienen que dar créditos al autor y crear licencias con los mismos términos.



6. **Atribución-No comercial-Sin Derivadas (CC BY-NC-ND):** Es la más restrictiva, permite descargar y compartir, pero no pueden cambiarlas de ninguna manera.



5. VIRTUALIZACIÓN

5.1 Virtualización dual

https://github.com/colival03/Oliva_Cristian_ReclSO/blob/main/Virtualización/Virtualización%20dual.mp4

5.2 Red NAT

https://github.com/colival03/Oliva_Cristian_ReclSO/blob/main/Virtualización/Red%20Nat.mp4