

Práctica 03

| DOCENTE | CARRERA | CURSO |
|-----------------------------|---|---------------------|
| MSc. Maribel Molina Barriga | Escuela Profesional de Ingeniería de Software | Sistemas Operativos |

| GRUPO | TEMA | DURACIÓN |
|-------|---------------------|----------|
| 6 | Instalación Distros | 5 horas |

Integrantes

- José Carlos Machaca Vera
- Jhosep Alonso Mollapaza Morocco
- Patrick Andres Ramirez Santos

Índice

| | |
|---------------------------------------|-----------|
| 1. Objetivos | 2 |
| 2. Instalación Manjaro minimal | 2 |
| 2.1. VirtualBox | 2 |
| 2.2. Instalacion | 3 |
| 3. Instalacion de paquetes | 4 |
| 4. Instalación UwUntu | 6 |
| 4.1. VirtualBox | 6 |
| 4.2. Instalacion | 7 |
| 5. Instalacion de paquetes | 8 |
| 6. Compilacion C++ | 10 |
| 6.1. Compilacion Manjaro | 10 |
| 6.2. Compilación UwUntu | 12 |
| 7. Conclusiones | 13 |
| 8. Recomendaciones | 13 |

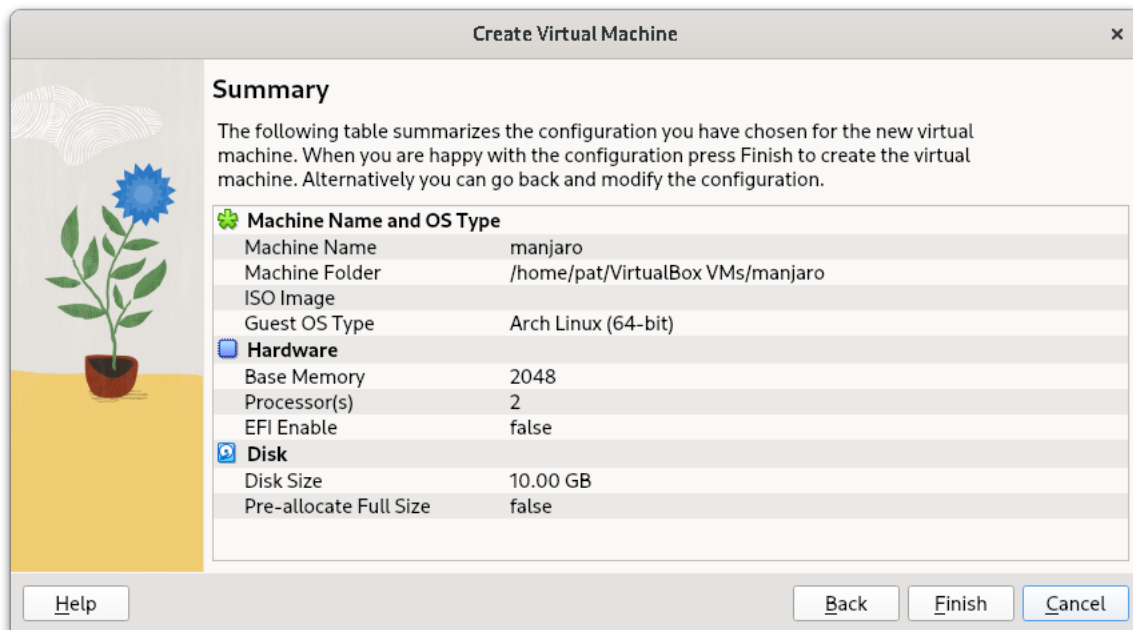
1. Objetivos

- Instalar 2 distribuciones de Linux, en este caso Manjaro y Uwuntu utilizando máquinas virtuales
- Crear y ejecutar programas en las distribuciones instaladas
- Crear una guía de instalacion eficiente y concisa

2. Instalación Manjaro minimal

2.1. VirtualBox

Manjaro no dispone de una version minima para arquitecturas x86, solamente para ARM y debido a la dificultad de virtualizar dicho entorno se procedio a instalar la version grafica minimal, para lo cual se configuró la máquina virtual de la siguiente forma:

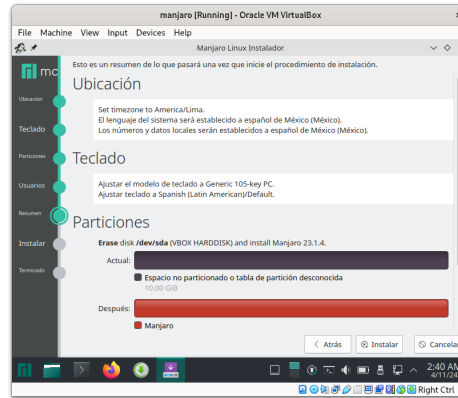


2.2. Instalacion

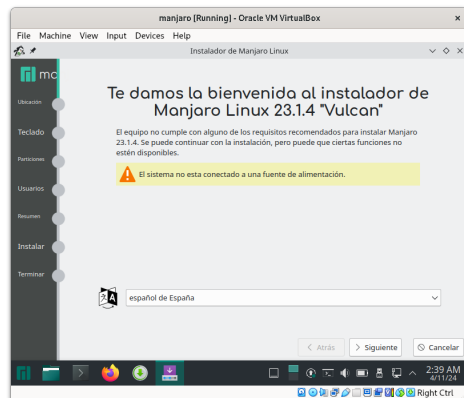
Se asignó 10 GB de almacenamiento, 2 GB de RAM y 2 procesores. Todo esto se realizó a lo largo de varias interfaces descritas así:



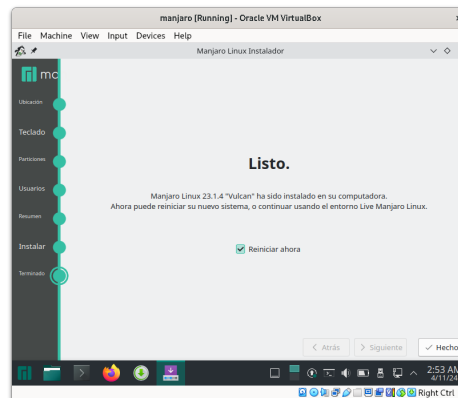
Boot



Resumen



Menu inicial

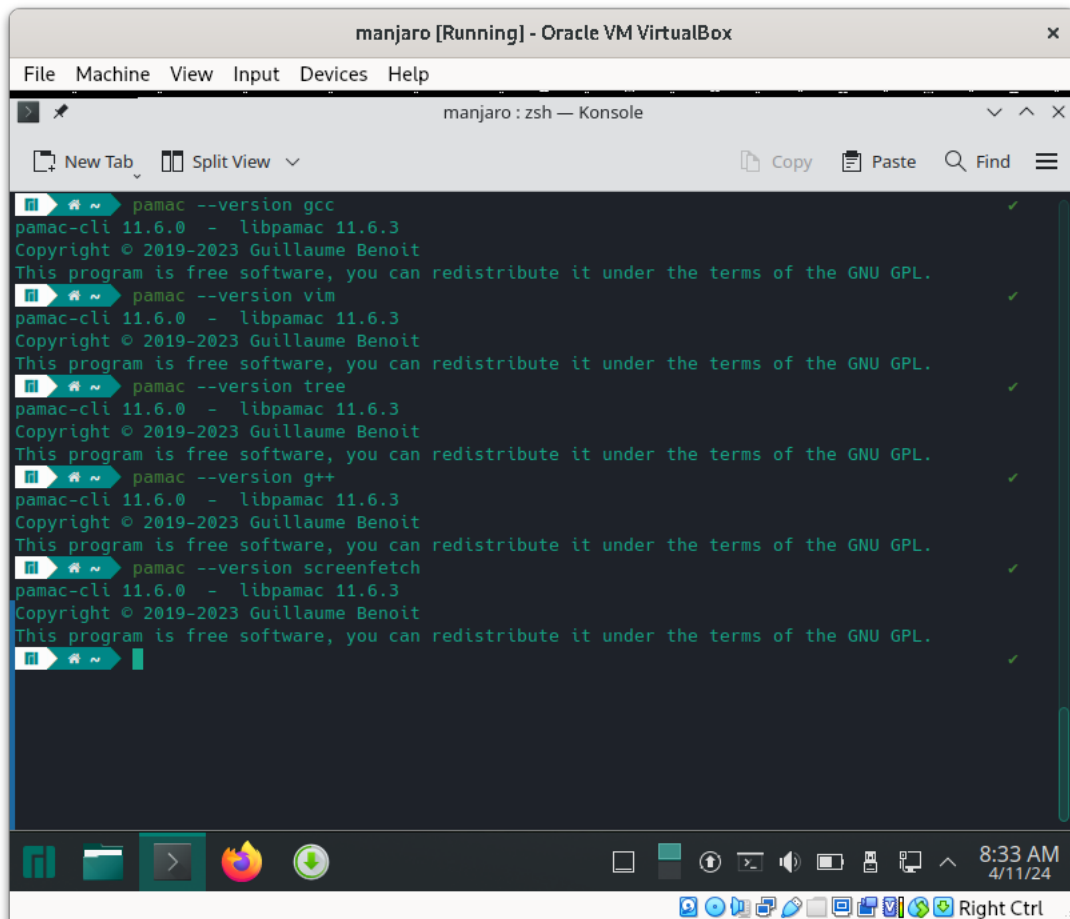


Menu cierre

3. Instalacion de paquetes

Se instala los compiladores usando base-devel, se añade vim, tree y screenfetch. TOdo esto se realiza con los siguientes comandos y finalmente se muestra una imagen con los paquetes instalados usando la flag -Q de pamac:

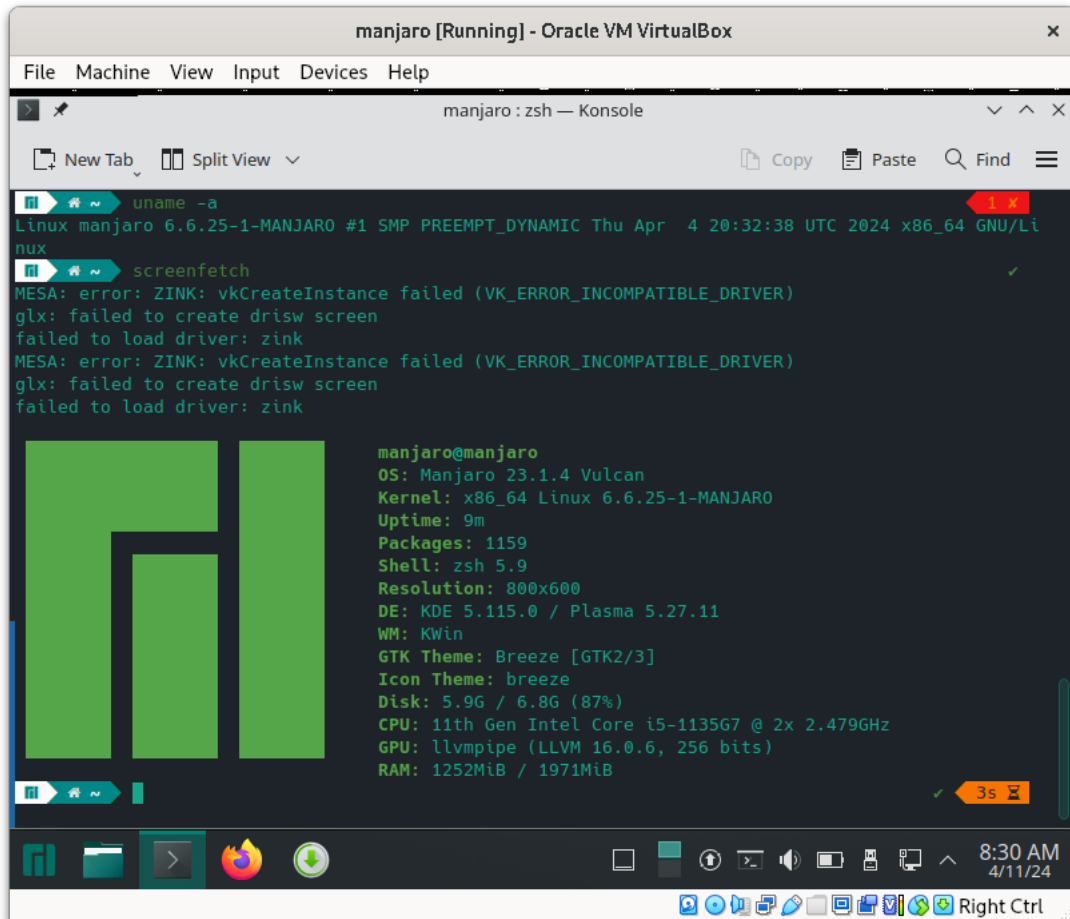
```
$ pamac install base-devel
$ pamac install vim
$ pamac install tree
$ pamac install screenfetch
$ pamac -Q [Nombre del paquete]
```



```
manjaro [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
manjaro : zsh — Konsole
New Tab Split View Copy Paste Find
$ pamac --version gcc
pamac-cli 11.6.0 - libpamac 11.6.3
Copyright © 2019-2023 Guillaume Benoit
This program is free software, you can redistribute it under the terms of the GNU GPL.
$ pamac --version vim
pamac-cli 11.6.0 - libpamac 11.6.3
Copyright © 2019-2023 Guillaume Benoit
This program is free software, you can redistribute it under the terms of the GNU GPL.
$ pamac --version tree
pamac-cli 11.6.0 - libpamac 11.6.3
Copyright © 2019-2023 Guillaume Benoit
This program is free software, you can redistribute it under the terms of the GNU GPL.
$ pamac --version g++
pamac-cli 11.6.0 - libpamac 11.6.3
Copyright © 2019-2023 Guillaume Benoit
This program is free software, you can redistribute it under the terms of the GNU GPL.
$ pamac --version screenfetch
pamac-cli 11.6.0 - libpamac 11.6.3
Copyright © 2019-2023 Guillaume Benoit
This program is free software, you can redistribute it under the terms of the GNU GPL.
$
```

Ademas se muestran los resultados de los comandos:

```
$ uname -a  
$ screenfetch
```



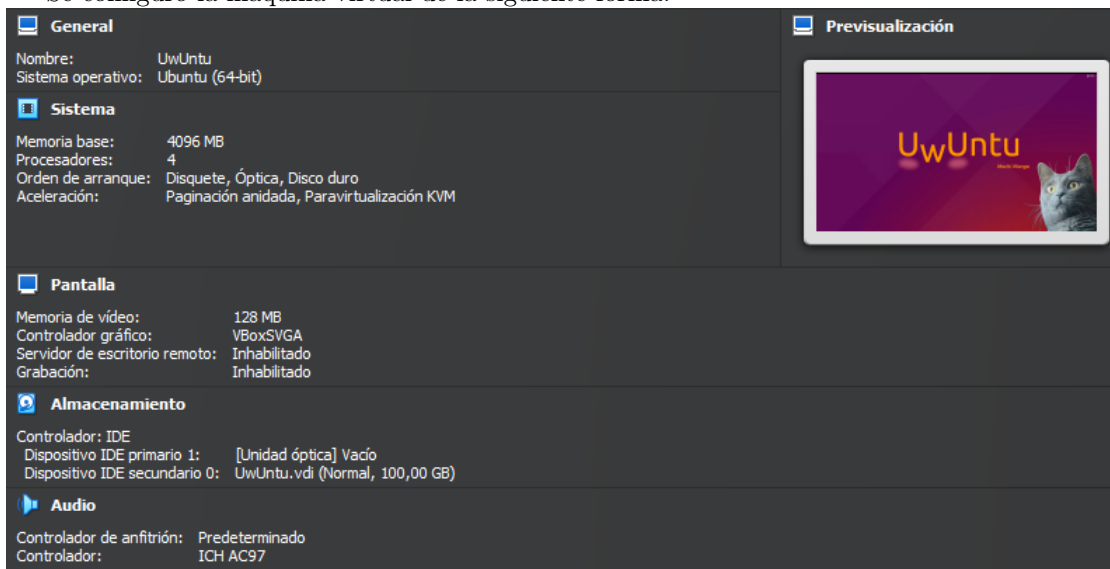
```
manjaro [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
manjaro : zsh — Konsole  
New Tab Split View Copy Paste Find  
$ uname -a  
Linux manjaro 6.6.25-1-MANJARO #1 SMP PREEMPT_DYNAMIC Thu Apr 4 20:32:38 UTC 2024 x86_64 GNU/Linux  
$ screenfetch  
MESA: error: ZINK: vkCreateInstance failed (VK_ERROR_INCOMPATIBLE_DRIVER)  
glx: failed to create drisw screen  
failed to load driver: zink  
MESA: error: ZINK: vkCreateInstance failed (VK_ERROR_INCOMPATIBLE_DRIVER)  
glx: failed to create drisw screen  
failed to load driver: zink  
manjaro@manjaro  
OS: Manjaro 23.1.4 Vulcan  
Kernel: x86_64 Linux 6.6.25-1-MANJARO  
Uptime: 9m  
Packages: 1159  
Shell: zsh 5.9  
Resolution: 800x600  
DE: KDE 5.115.0 / Plasma 5.27.11  
WM: KWin  
GTK Theme: Breeze [GTK2/3]  
Icon Theme: breeze  
Disk: 5.9G / 6.8G (87%)  
CPU: 11th Gen Intel Core i5-1135G7 @ 2x 2.479GHz  
GPU: llvmpipe (LLVM 16.0.6, 256 bits)  
RAM: 1252MiB / 1971MiB  
8:30 AM  
4/11/24  
Right Ctrl
```

4. Instalación UwUntu

4.1. VirtualBox

UwUntu al ser una distro de Ubuntu y que usa Ubuntu Budgie 10.6.1 que es conocido por ser simple y elegante, diseñado para ser fácil de usar, entonces la instalación de UwUntu por medio de VirtualBox no ha sido un problema solo hay que tener espacio ya que cuenta con varios paquetes y programas preinstalados, porque esta pensado para usuarios dedicados a los videojuegos y a la cultura popular.

Se configuro la máquina virtual de la siguiente forma.

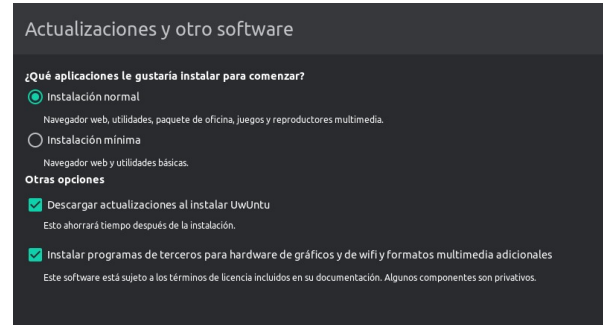


4.2. Instalacion

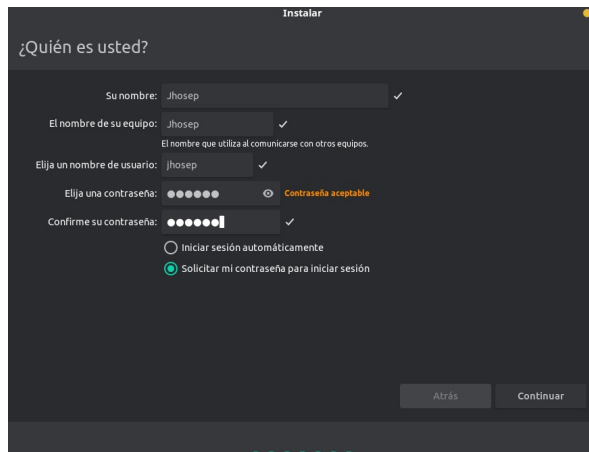
Para este sistema se asigno 100 GB de almacenamiento, 4 GB de RAM y 4 procesadores. Estas son los procesos de instalación que realiza UwUntu:



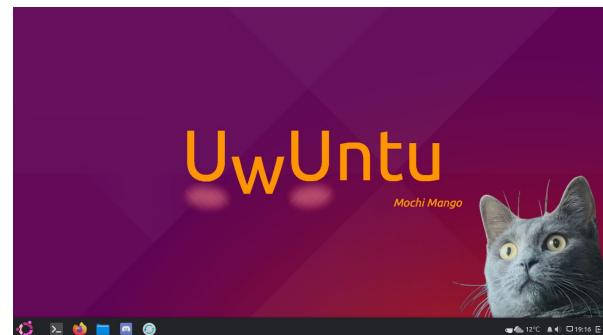
Menu Inicial



Instalación



Usuario



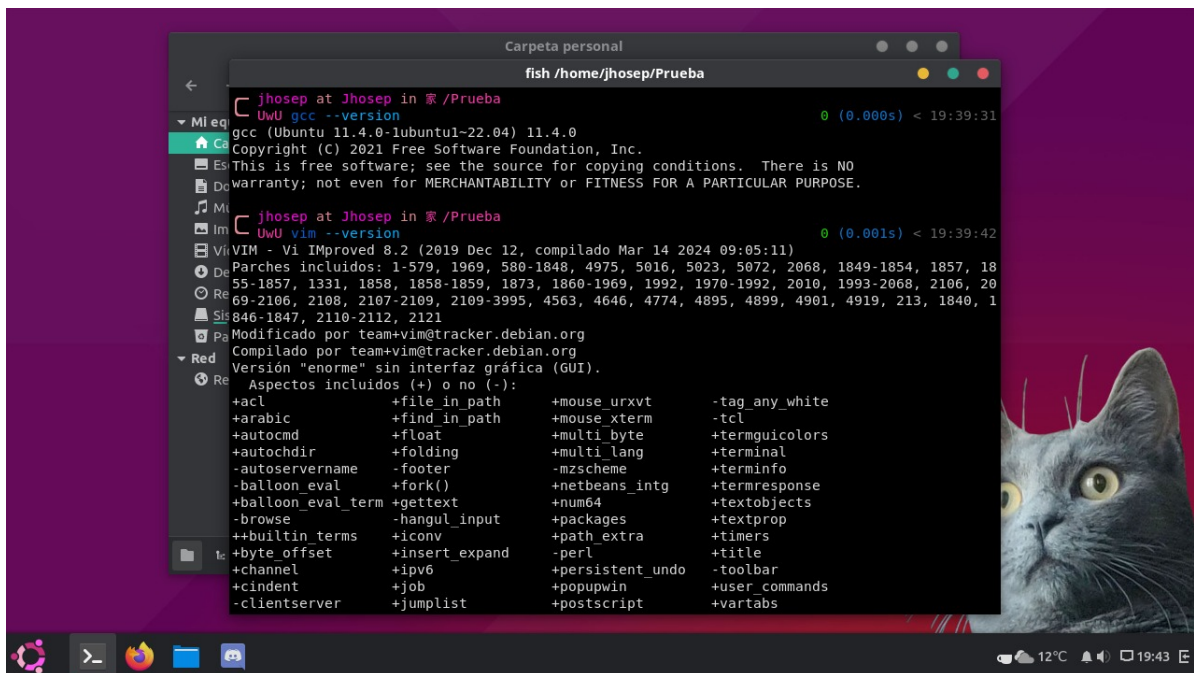
Escritorio

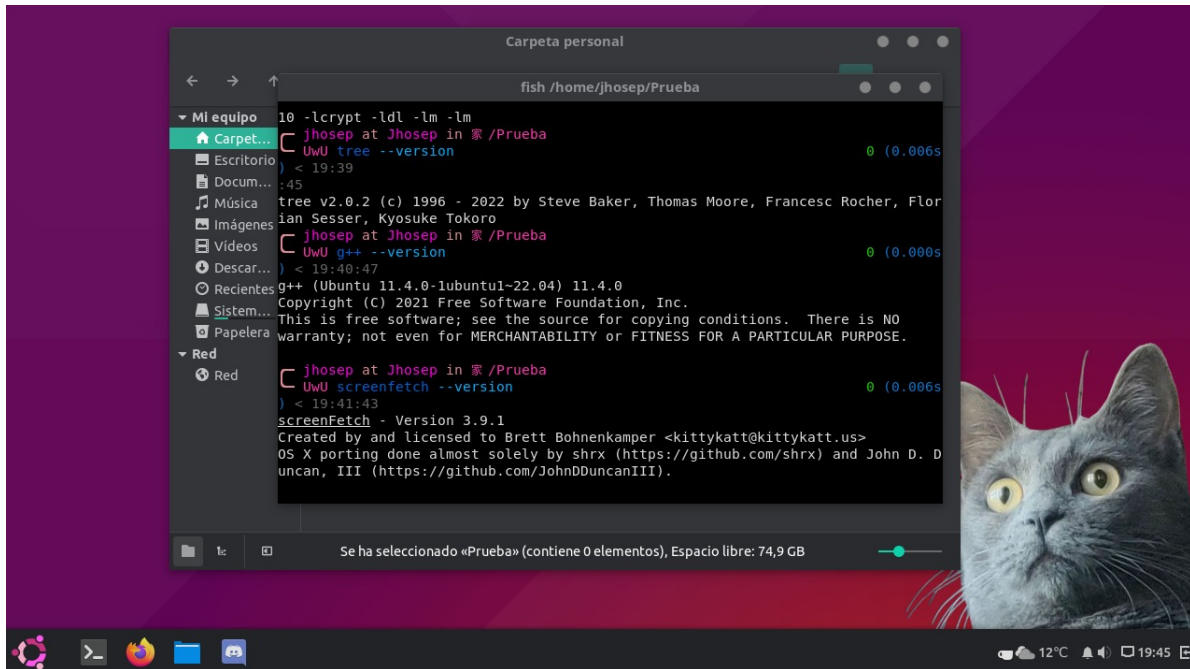
5. Instalacion de paquetes

Se instala los compiladores, vim, tree y screenfetch solicitados mediante la consola:

```
UwU sudo apt install gcc
UwU sudo apt install vim
UwU sudo apt install tree
UwU sudo apt install g++
UwU sudo apt install screenfetch
```

Y se muestran las versiones para demostrar que fueron instaladas.

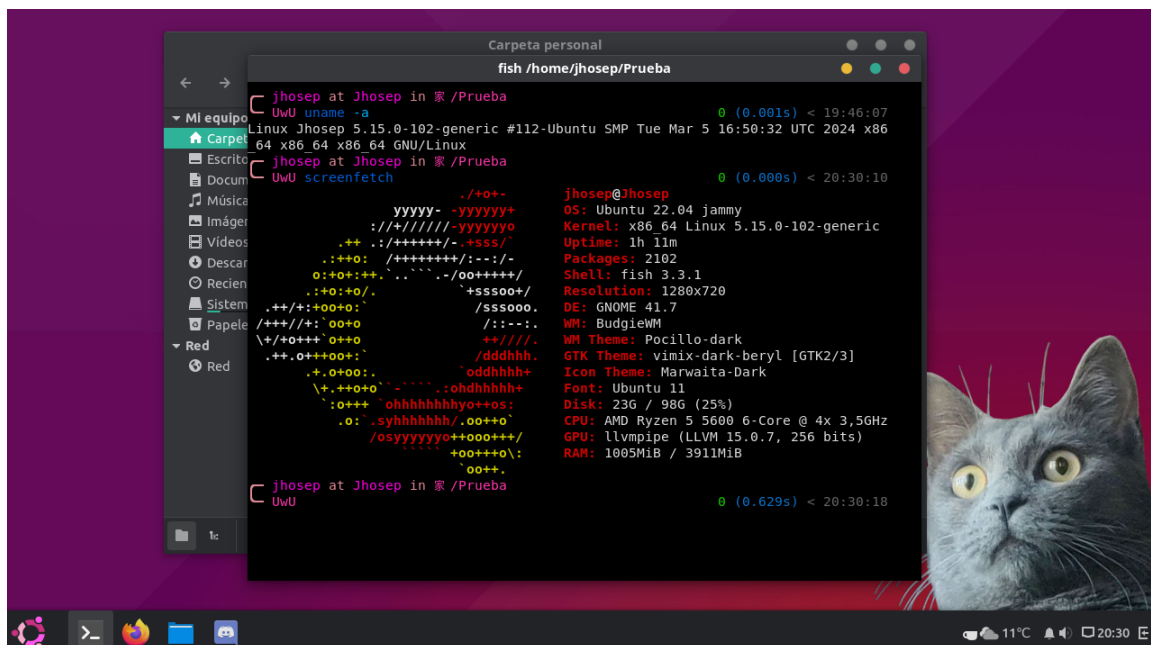




Ademas se muestran los resultados de los comandos:

UwU uname -a

UwU screenfetch

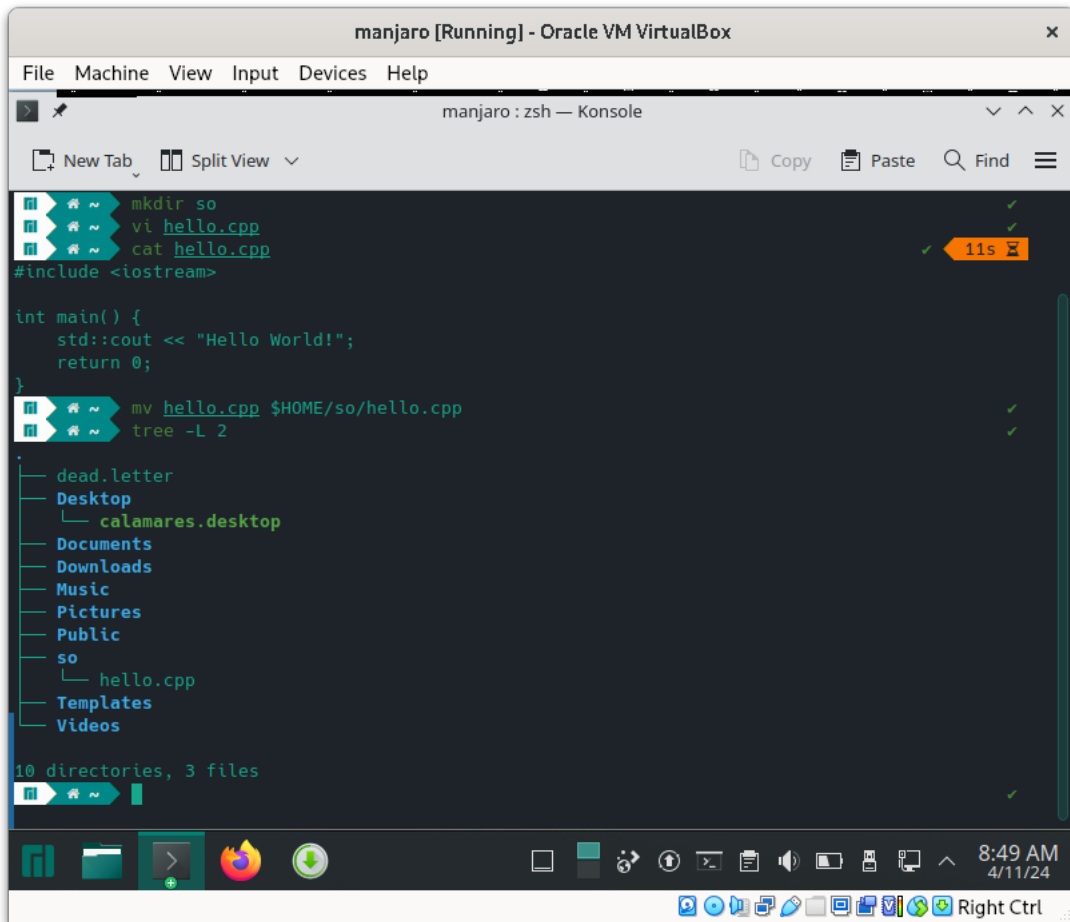


6. Compilacion C++

6.1. Compilacion Manjaro

Se crea el directorio, se crea el archivo en \$HOME y luego se mueve este al directorio so.

```
$ mkdir $HOME/so
$ vi hello.cpp
$ mv hello.cpp /so/hello.cpp
$ tree
```



The screenshot shows a terminal window titled "manjaro [Running] - Oracle VM VirtualBox". The terminal is running a zsh shell. The commands and their outputs are as follows:

```
manjaro: zsh — Konsole
New Tab Split View
Copy Paste Find
mkdir so
vi hello.cpp
cat hello.cpp
#include <iostream>

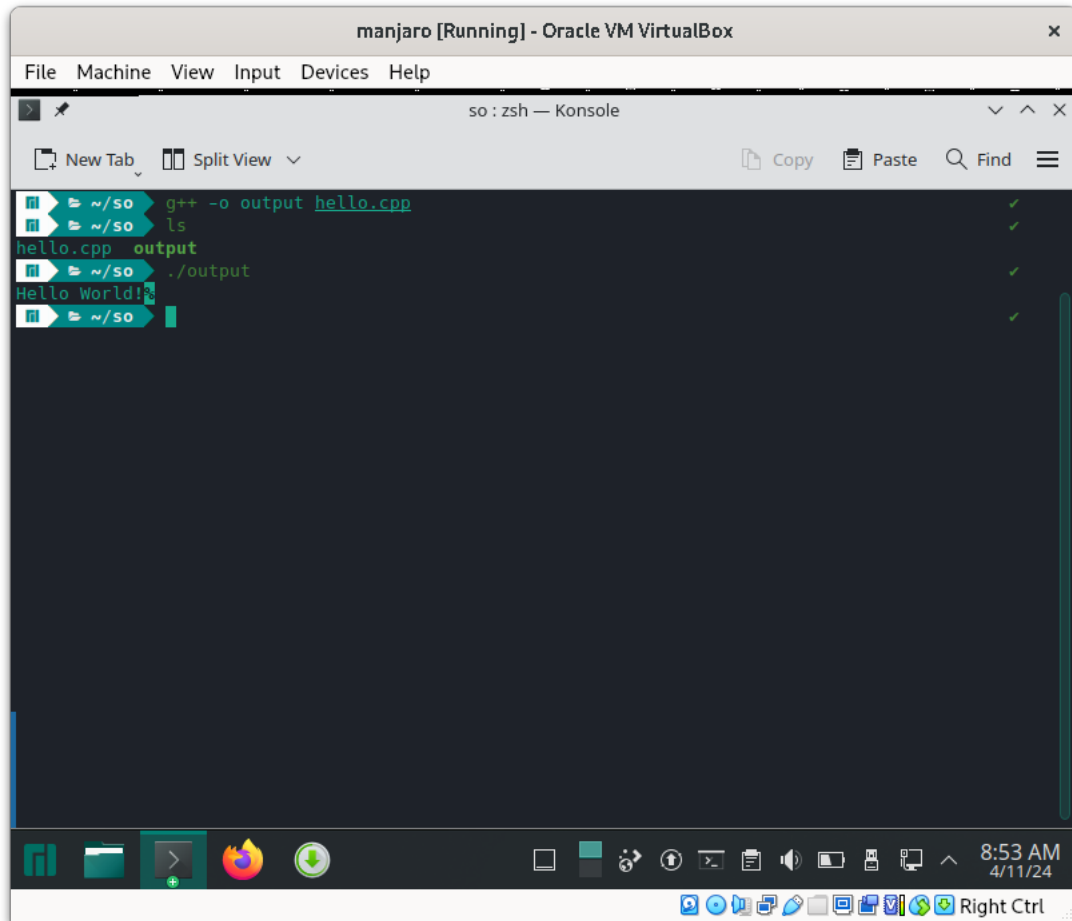
int main() {
    std::cout << "Hello World!";
    return 0;
}
mv hello.cpp $HOME/so/hello.cpp
tree -L 2
.
├── dead.letter
├── Desktop
│   └── calamares.desktop
├── Documents
├── Downloads
├── Music
├── Pictures
├── Public
├── so
│   └── hello.cpp
├── Templates
└── Videos

10 directories, 3 files
```

The terminal window also shows a taskbar at the bottom with various icons and a system clock indicating 8:49 AM on 4/11/24.

Para ejecutar el programa primero se compila y luego se ejecuta de la siguiente forma:

```
$ g++ -o output hello.cpp  
$ ./output
```



The screenshot shows a terminal window titled "manjaro [Running] - Oracle VM VirtualBox". The terminal is running a zsh shell. The user has entered the following commands and received the following output:

```
so: zsh — Konsole  
New Tab Split View  
Copy Paste Find  
~/so$ g++ -o output hello.cpp  
~/so$ ls  
hello.cpp  output  
~/so$ ./output  
Hello World!  
~/so$
```

The terminal window has a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". The status bar at the bottom shows the time as 8:53 AM on 4/11/24 and the text "Right Ctrl".

6.2. Compilación UwUntu

Creamos el directorio en \$Home y creamos el archivo hello.c

```
UwU $HOME
UwU mkdir SO
UwU cd SO
UwU vim hello.c
```

```
fish /home/jhosep/SO
jhosep at Jhosep in 家
UwU $HOME 0 (0.000s) < 20:33:09
jhosep at Jhosep in 家
UwU mkdir SO 0 (0.000s) < 20:33:12
jhosep at Jhosep in 家
UwU cd SO 0 (0.002s) < 20:33:23
jhosep at Jhosep in 家 /SO
UwU vim hello.c 0 (0.000s) < 20:33:25
```

```
#include <stdio.h>

int main() {
    printf("Hola Mundo\n");
    return 0;
}
```

```
vim hello.c /home/jhosep/SO
#include <stdio.h>

int main() {
    printf("Hola mundo");
    return 0;
}
~
~
~
```

Y luego compilamos el programa y lo ejecutamos:

```
UwU gcc hello.c -o hello
UwU ./hello
```

```
jhosep at Jhosep in 家 /SO
UwU gcc hello.c -o hello 0 (0.024s) < 20:56:41
jhosep at Jhosep in 家 /SO
UwU ./hello 0 (0.020s) < 20:57:18
Hola mundo
jhosep at Jhosep in 家 /SO
UwU 0 (0.000s) < 20:57:24
```

7. Conclusiones

8. Recomendaciones

- No escoger Manjaro para una instalacion minima puesto que al ser una version modificada de Arch su valor agregado esta en la GUI, sin ella sería mejor instalar Arch directamente. Además la version minima solo esta disponible para arquitecturas ARM que son bastante difíciles de virtualizar.
-