

KERNEL GNU/LINUX

Manual de Compilación

Índice

I. Introducción	1
II. Objetivos	2
a) <i>General</i>	2
b) <i>Específicos</i>	2
III. Kernel Linux	3
a) <i>Descripción</i>	3
b) <i>Topología del Equipo de Prueba</i>	3
c) <i>Preparación</i>	4
1. Descargar	5
2. Carpeta de Trabajo.....	5
3. Descomprimir	6
4. Cambiar Directorio.....	6
5. Archivo .CONFIG.....	7
d) <i>Configuración</i>	7
1. Menuconfig	8
2. Nconfig	8
3. Xconfig.....	9
e) <i>Compilación</i>	10
1. # make -j [número_núcleos]	10
2. # make -j [número_núcleos] modules.....	12
3. # make.....	13
4. # sudo make install.....	14
5. # sudo make install modules	15

<i>f) Instalación</i>	16
1. Mover el Kernel	16
2. Generar INITRAMFS	17
3. Gestor de Arranque	18
<i>g) Uso</i>	18
IV. Contacto	22

I. Introducción

Para la primera fase del proyecto de laboratorio en el curso de Sistemas Operativos 1, se desarrolló el presente manual de ayuda que abarca aspectos básicos descriptivos, preparación y configuración sobre el proceso para realizar una compilación manual del kernel de GNU/Linux, su debida instalación y uso.

II. Objetivos

a) General

- Brindar un tutorial intuitivo, informativo y conciso sobre la compilación de un kernel GNU-Linux.

b) Específicos

- Describir aspectos generales del kernel GNU/Linux.
- Informar sobre una correcta preparación antes de compilar.
- Detallar sobre la configuración básica de la compilación.
- Indicar la instalación y uso del nuevo kernel compilado.

III. Kernel Linux

a) Descripción

Linux es un sistema operativo libre, basado en Unix. Es uno de los principales ejemplos de software libre y de código abierto. Linux está licenciado bajo la GPL v2 y está desarrollado por colaboradores de todo el mundo. El desarrollo del día a día tiene lugar en la Linux Kernel Mailing List Archive.

El núcleo Linux fue concebido por el entonces estudiante de ciencias de la computación finlandés Linus Torvalds en 1991. Linux consiguió rápidamente desarrolladores y usuarios que adoptaron códigos de otros proyectos de software libre para usarlos con el nuevo sistema operativo. El núcleo Linux ha recibido contribuciones de miles de programadores de todo el mundo. Normalmente Linux se utiliza junto a un empaquetado de software, llamado distribución GNU/Linux y servidores.

Debido a que el kernel de GNU/Linux es de código abierto, fácilmente se puede acceder al mismo y emplearlo sin ninguna consecuencia.

<https://github.com/torvalds/linux>

b) Topología del Equipo de Prueba

Para desarrollar el presente manual se utilizó como host una Notebook con las siguientes características:

- Manjaro Linux 16.08 x64 bits KDE edition
- Intel® Coore™ i3-4005U CPU @1.70 GHz @1.70 GHz
- 96.0 Gb de almacenamiento particionado
- 4.00 Gb Memoria Swap
- 8.00 Gb DDR3
- Video Integra Intel HD Graphics

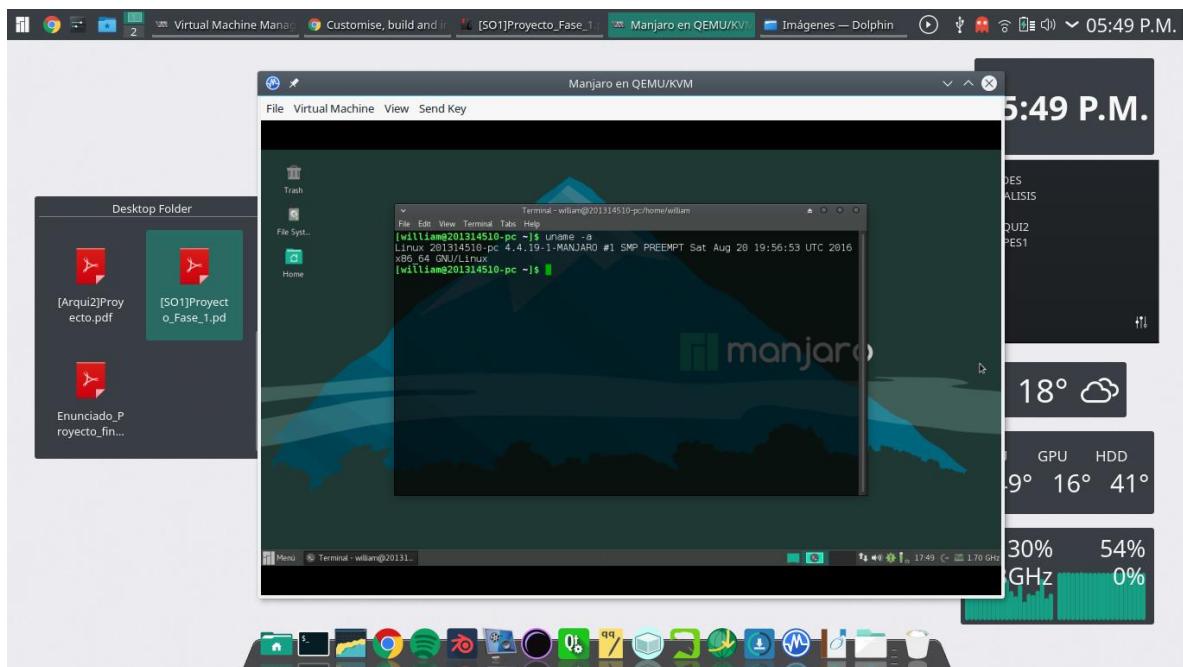
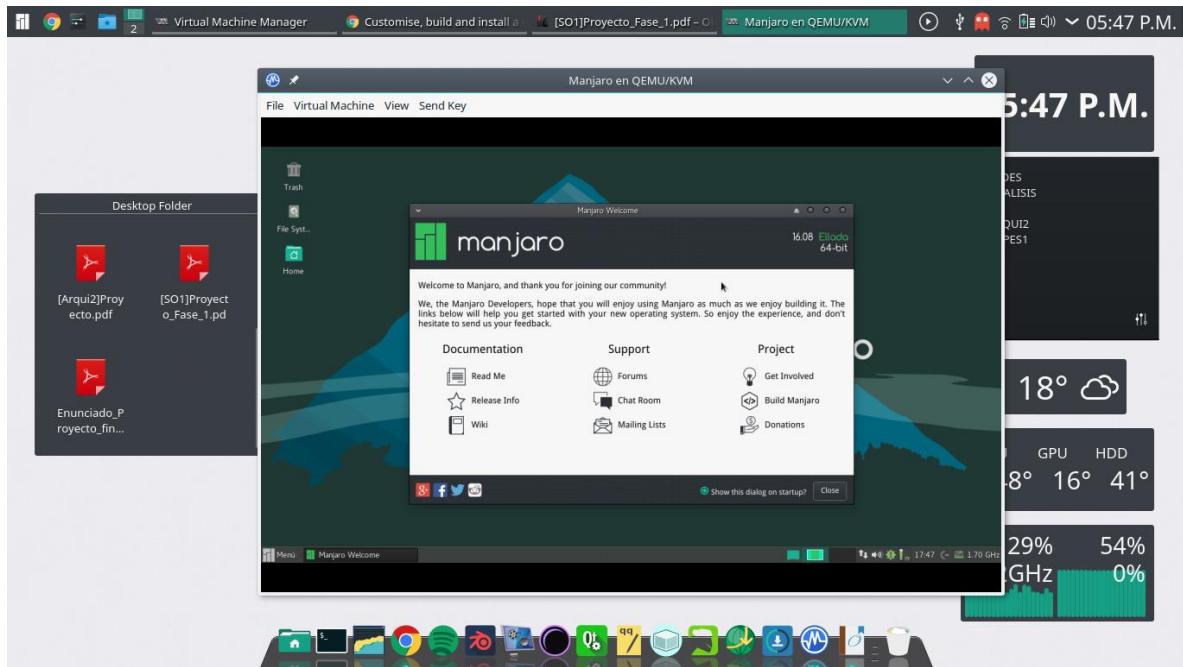
La cual utilizando el hipervisor KVM se virtualizó una máquina virtual guest con las propiedades siguientes:

- Manjaro Linux 16.08 x64 bits XFCE edition
- Intel® Coore™ i3-4005U CPU @1.70 GHz @1.70 GHz
- 16.0 Gb de almacenamiento particionado
- 4.00 Gb Memoria Swap
- 4.00 Gb DDR3
- Video VGA Genérico

c) Preparación

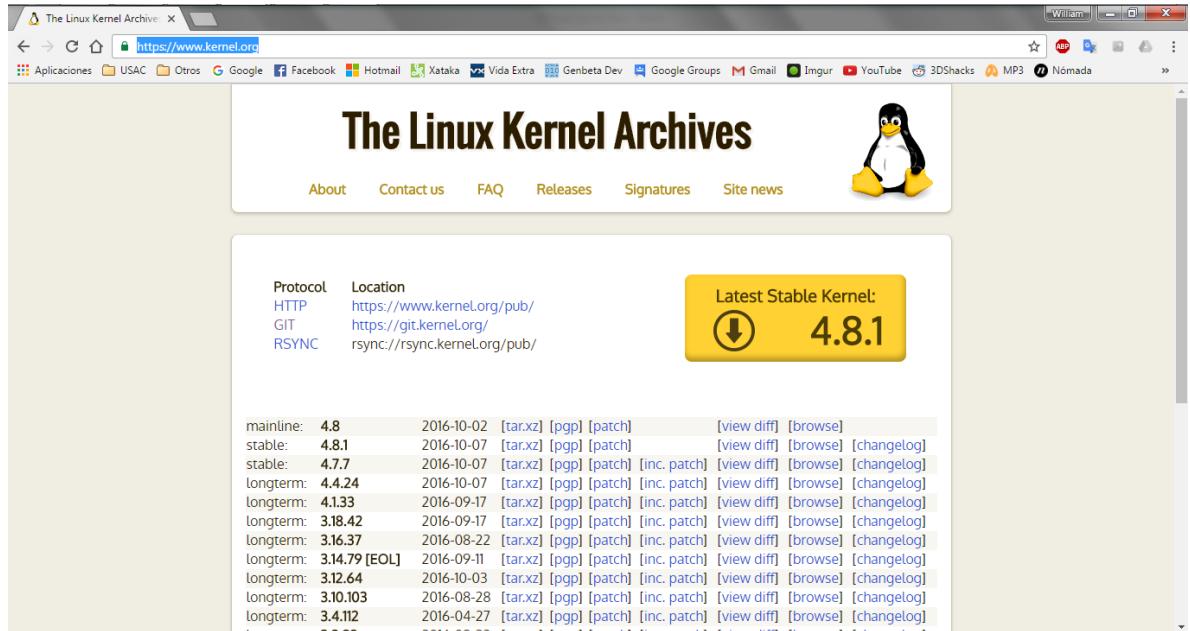
Por motivos de seguridad y accesibilidad se utilizó la topología descrita previamente, realizando el proceso de compilación principal en la máquina virtualizada guest.

Esta máquina guest utilizó el kernel genérico proporcionado por el sistema operativo:



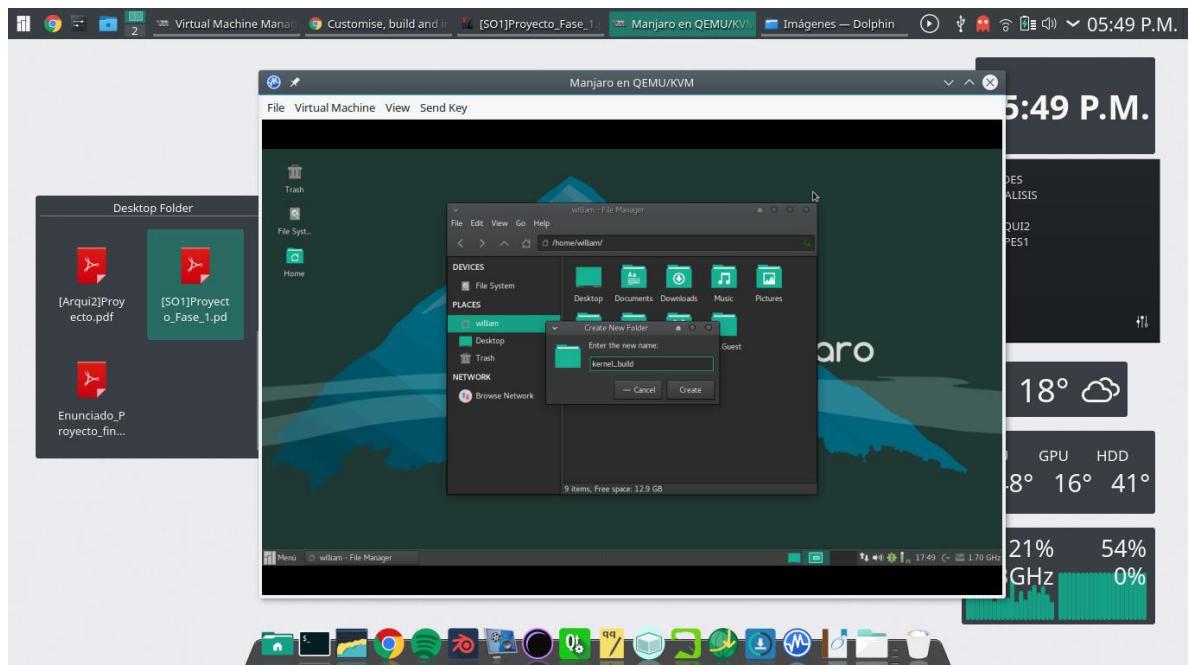
1. Descargar

Dirigirse al enlace: <https://www.kernel.org/> y descargar el código fuente del kernel elegido para compilar



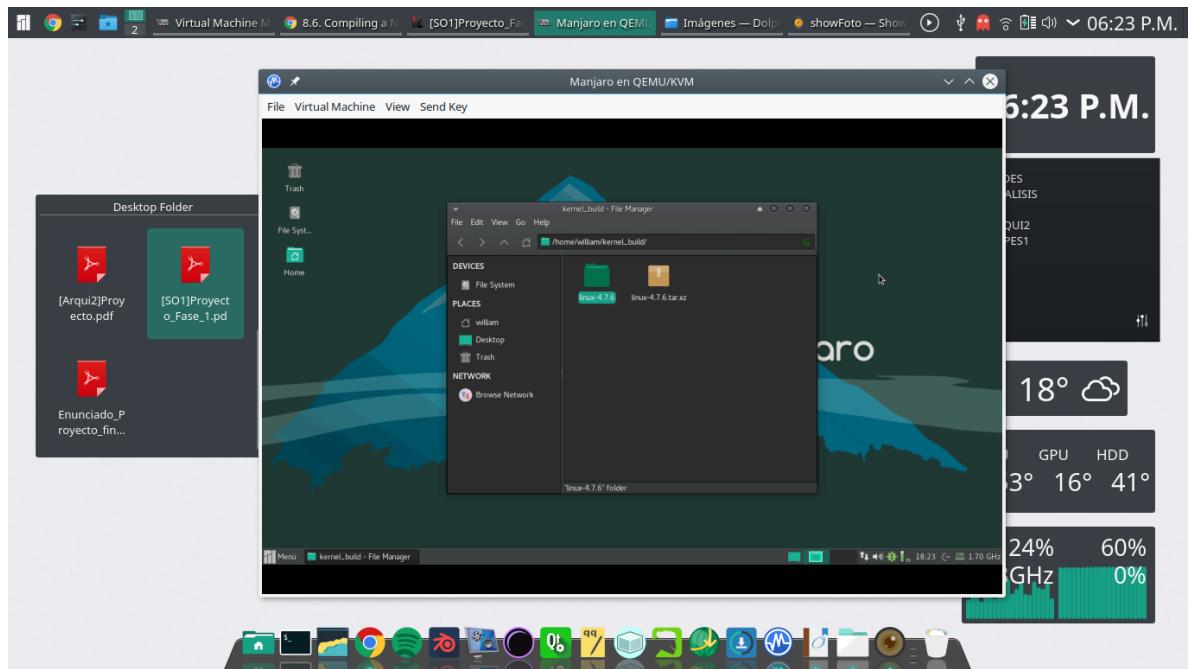
2. Carpeta de Trabajo

Crear una carpeta para utilizarla en el proceso de compilación.



3. Descomprimir

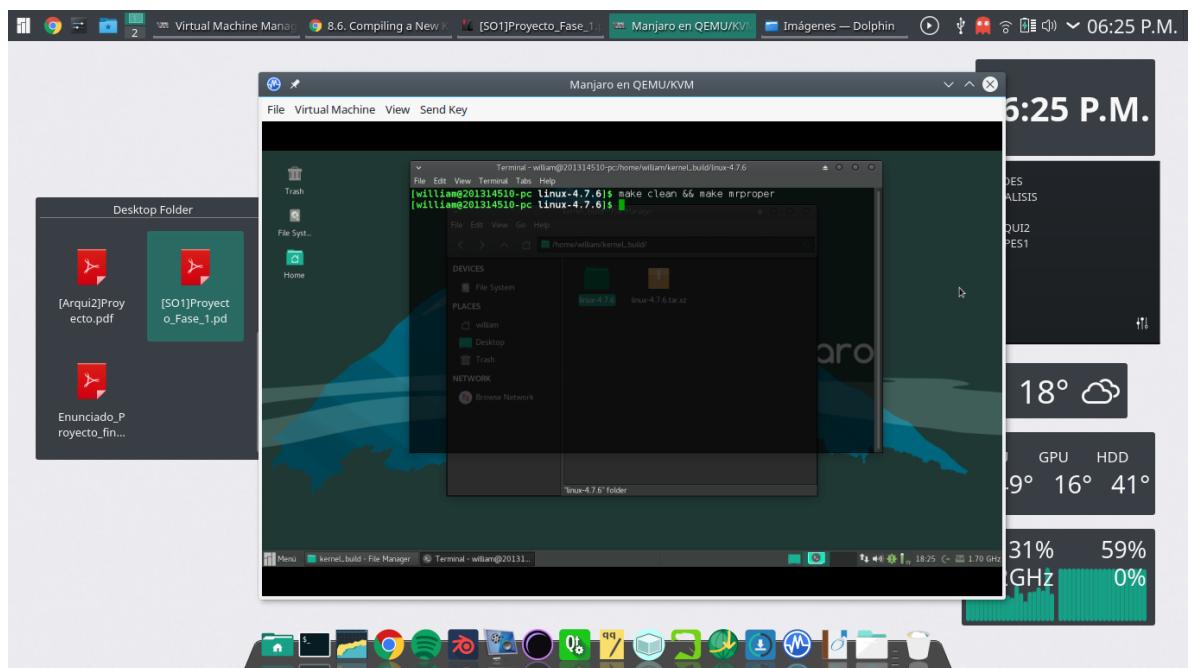
Descomprimir el código fuente del kernel descargado previamente



4. Cambiar Directorio

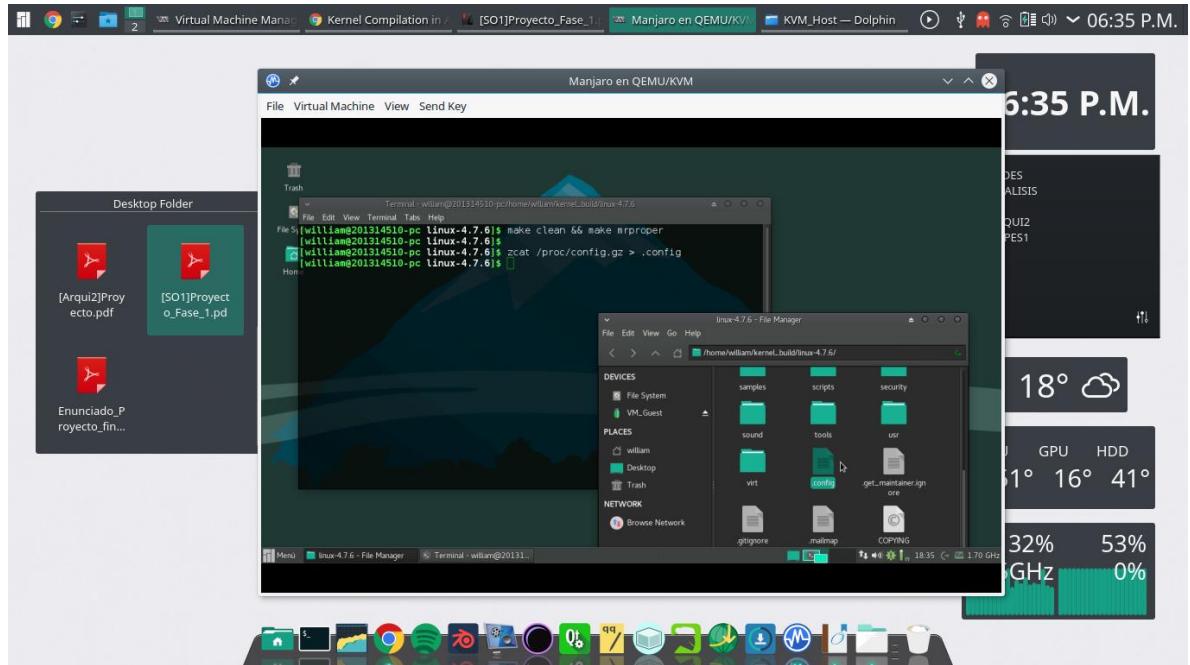
En la consola dirigirse a la carpeta descompresa del kernel y ejecutar:

```
# make clean && make mrproper
```



5. Archivo .CONFIG

Generar el archivo .config basado en el archivo de configuración del sistema operativo en ejecución



d) Configuración

La configuración del kernel a compilar es la parte más importante y delicada del proceso, se debe seleccionar múltiples aspectos del kernel y tomar en cuenta el incluir todos los módulos que se planean utilizar, como drivers de dispositivos por ejemplo; así mismo elegir las opciones compatibles con el equipo para así evitar problemas futuros.

Para aumentar la probabilidad de realizar una compilación exitosa del kernel seleccionado, se utilizó como base la configuración del kernel genérico del sistema en ejecución.

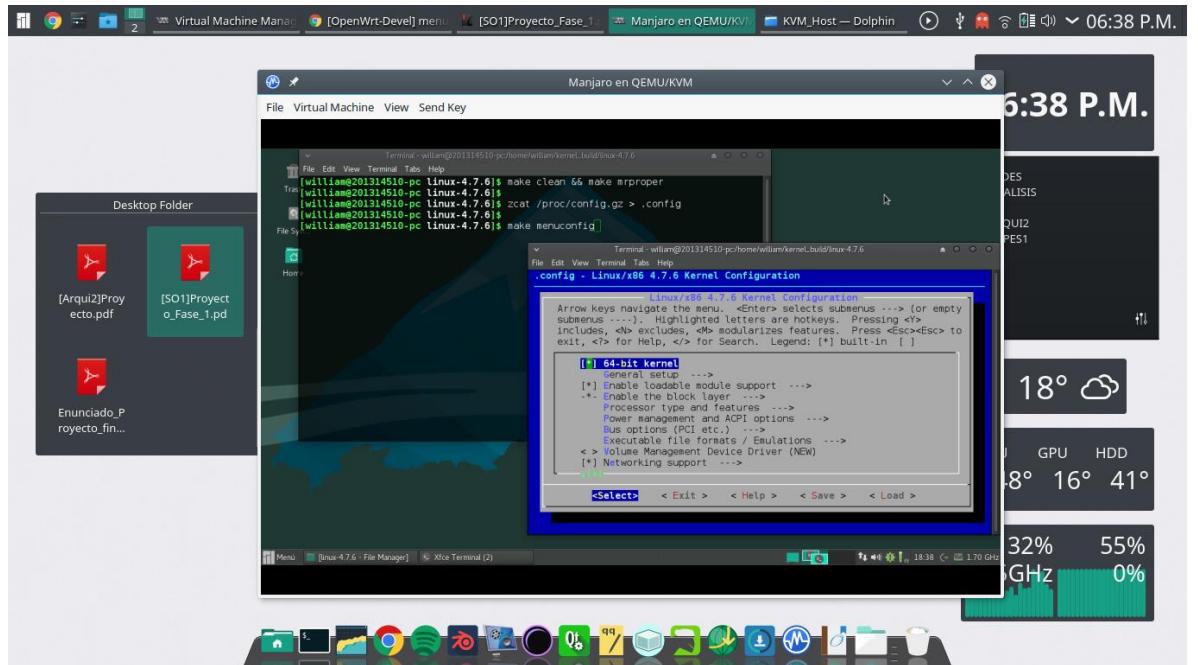
Seguidamente se debe modificar la configuración del kernel si se desea agregar o quitar funcionalidades, módulos y demás; para ello se dispone de diferentes métodos pero el que se recomienda y desarrolla en este manual es el método por interfaz gráfica.

Entre los métodos para configurar el kernel mediante interfaz gráfica están:

1. Menuconfig
2. Nconfig
3. Xconfig

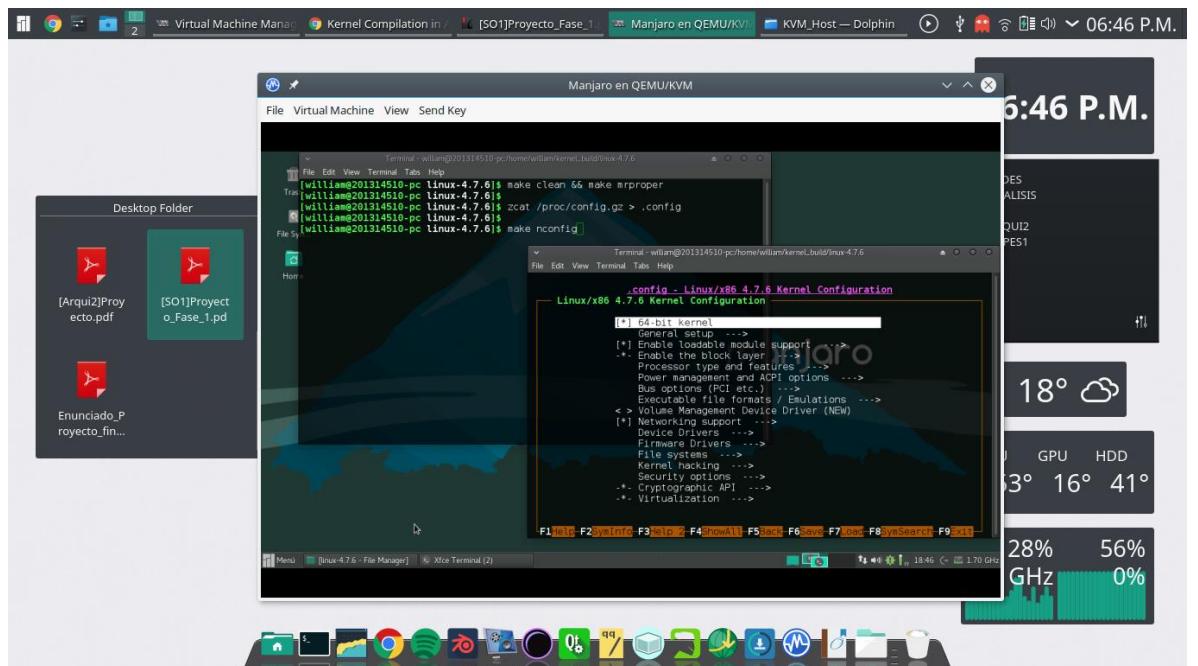
1. Menuconfig

make menuconfig



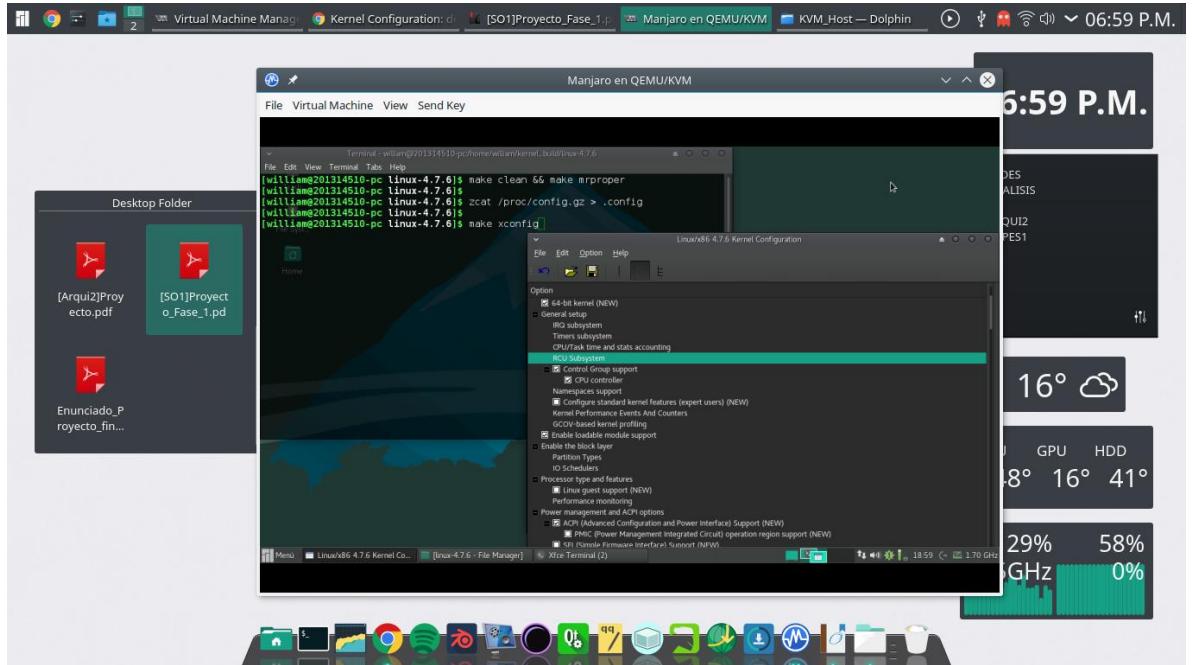
2. Nconfig

make nconfig



3. Xconfig

```
# make xconfig
```



Al utilizar la interfaz de usuario se puede navegar rápidamente entre los apartados a seleccionar de la configuración, volviendo el proceso más rápido y con mejores resultados. Al finalizar la configuración se deben guardar los cambios, sobreescribiendo o creando el archivo `.config`.

Este método por interfaz gráfica se prefiere porque si se utiliza el método tradicional por consola, habría que responder a una basta cantidad de preguntas una tras otra, lo cual se vuelve tedioso y propenso a crear una configuración defectuosa o incompatible con el equipo.

IMPORTANTE:

Si se utilizan múltiples programas para editar el mismo archivo `.config`, provocaría una compilación defectuosa o incompatible. Por ello se recomienda ejecutar el comando del inciso número 4, de la sección de preparación:

```
# make clean && make mrproper
```

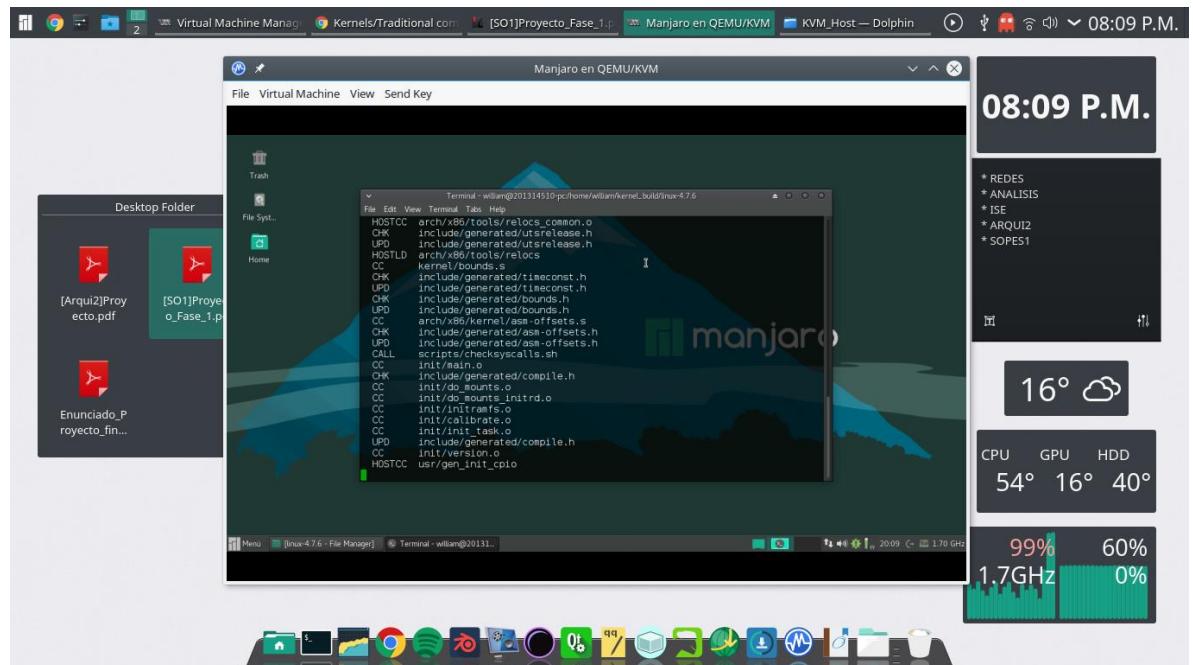
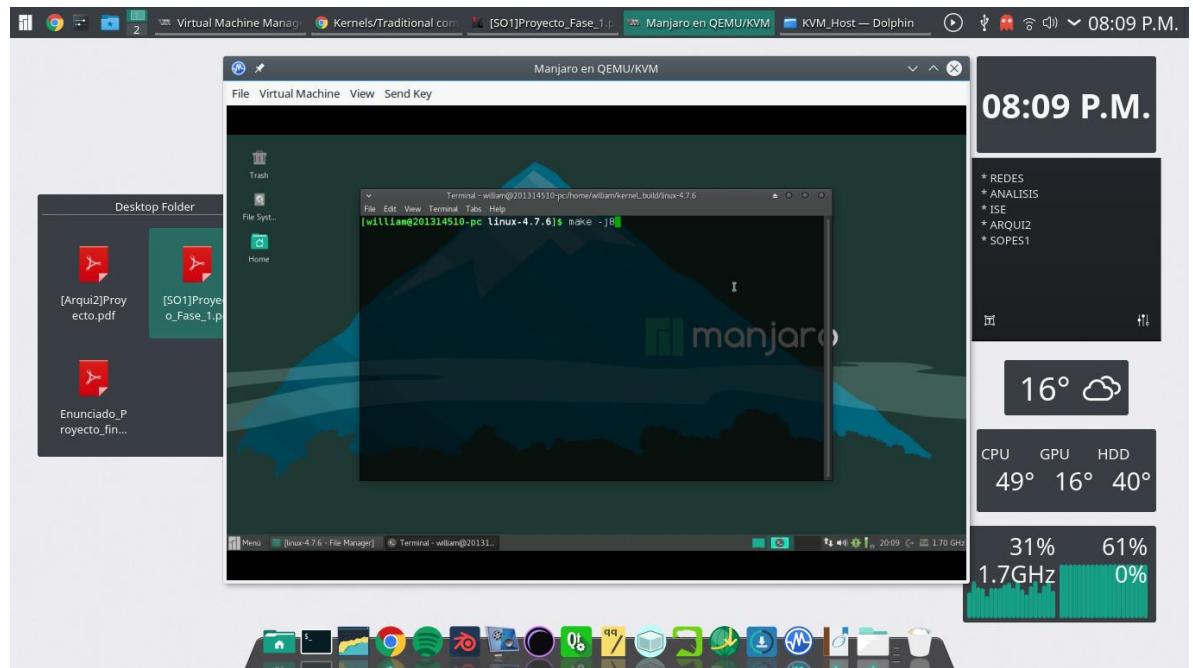
Así mismo, si se decide reconfigurar un `.config`, se recomienda ejecutar:

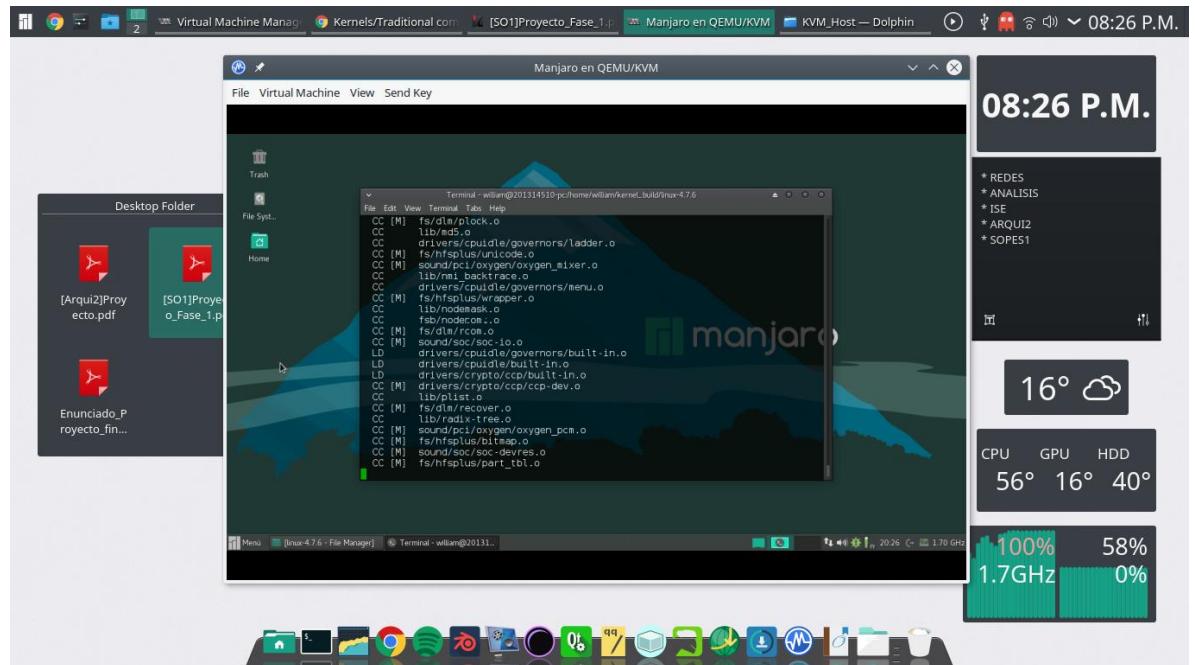
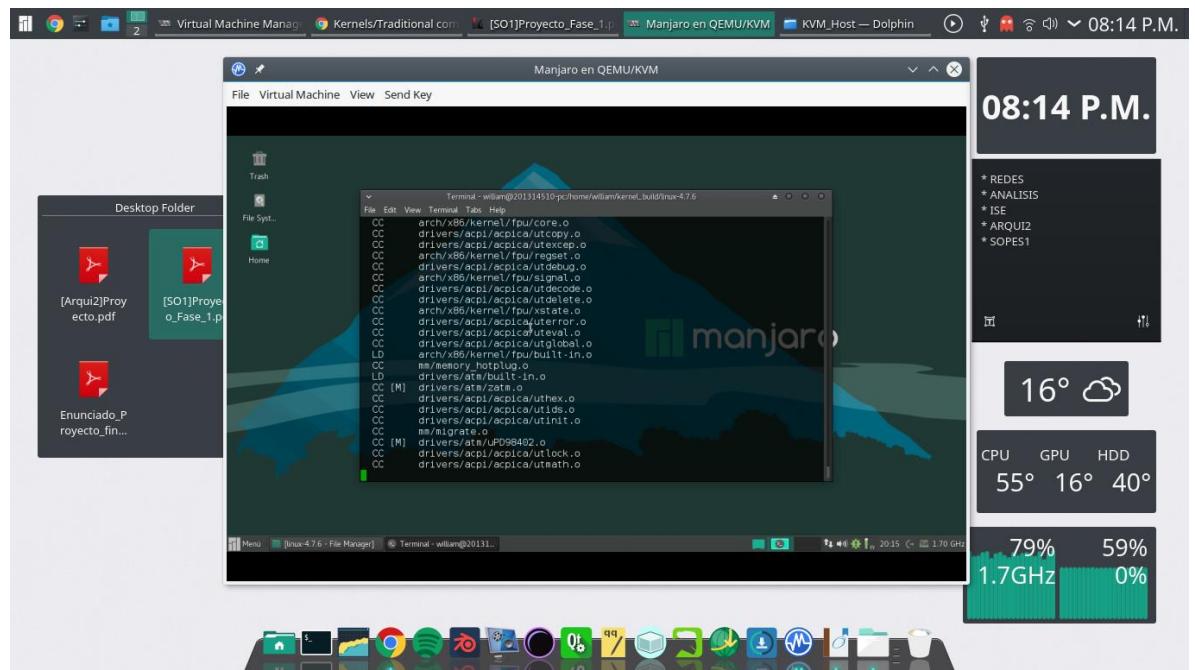
```
# make mrproper
```

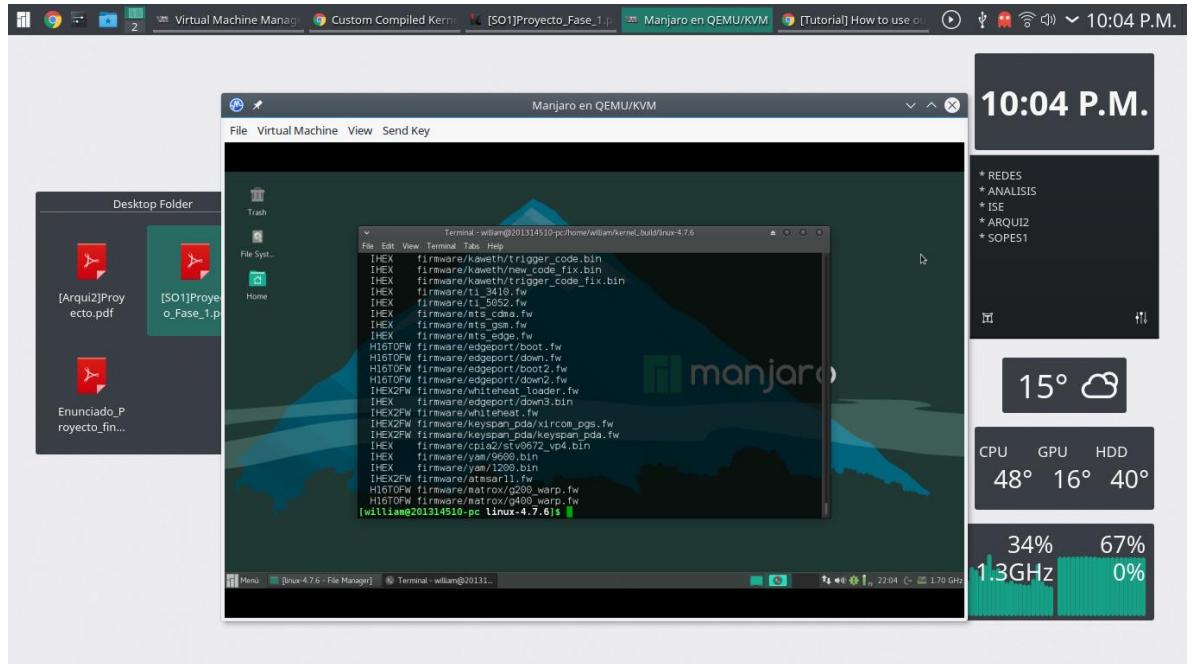
e) Compilación

La compilación del kernel suele durar varias horas, dependiendo de la capacidad del equipo en que se compila. Para realizarla se deben ejecutar los siguientes comandos:

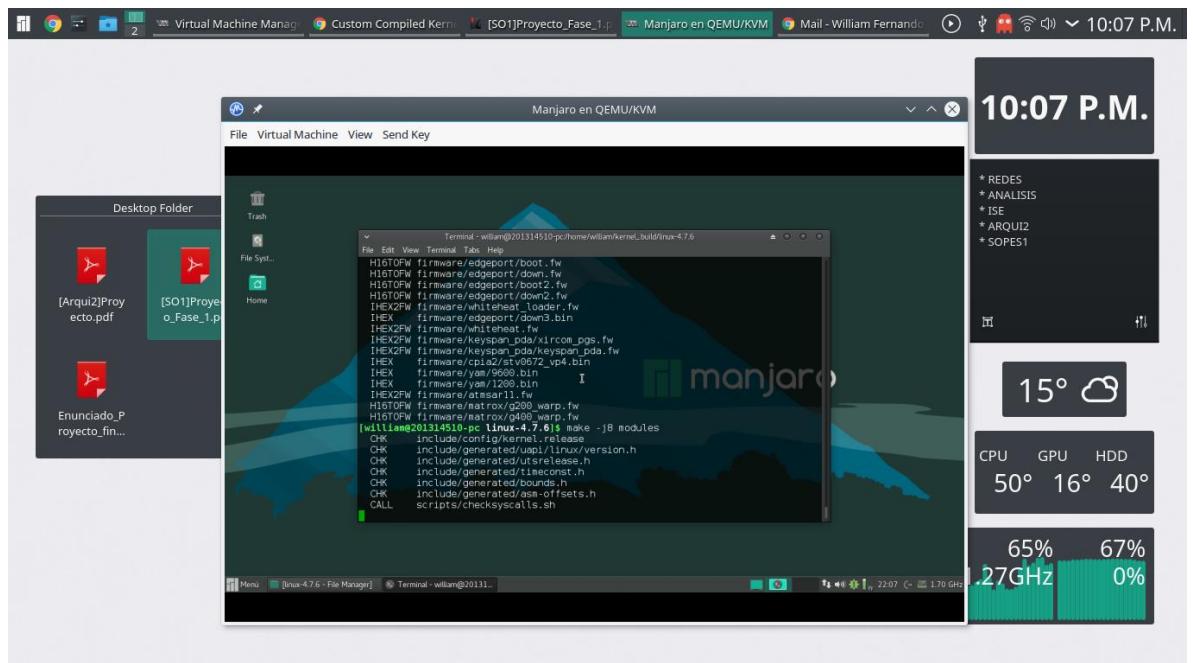
1. # make -j [número_núcleos]

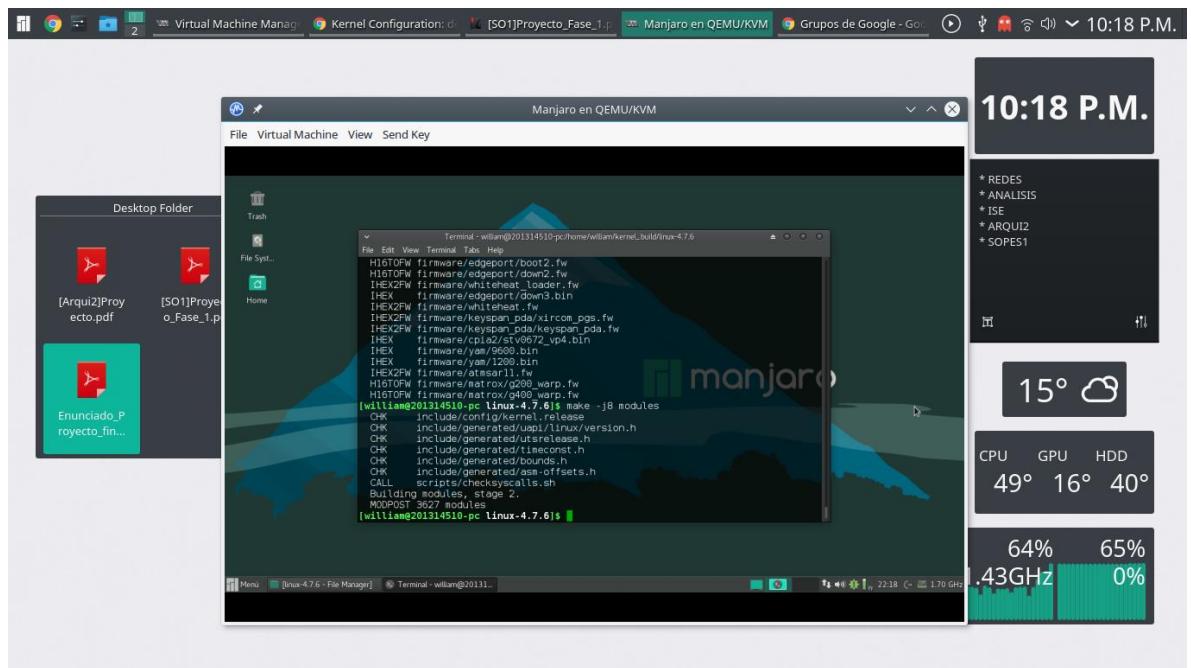




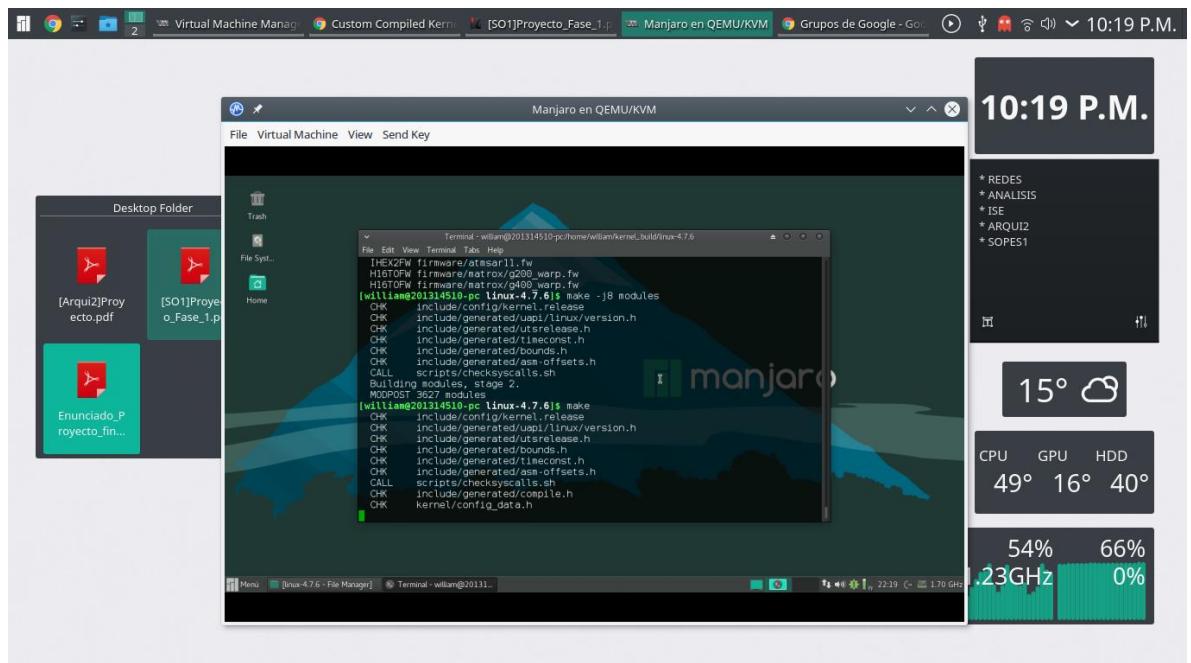


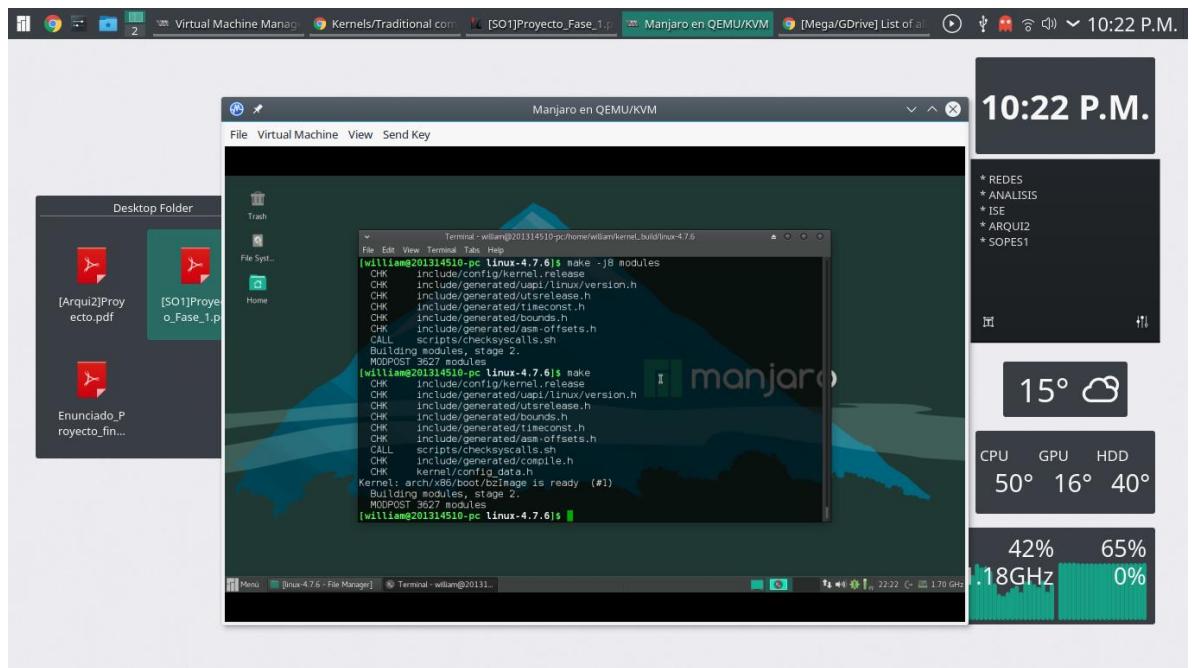
2. # make -j [número_núcleos] modules



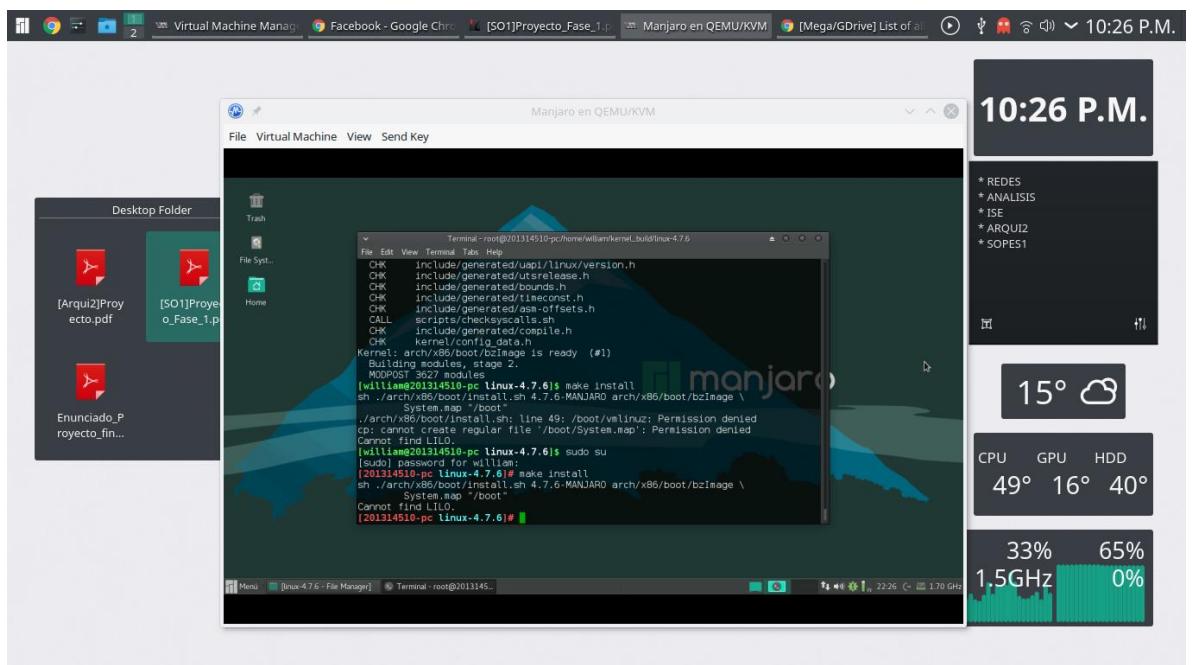


3. # make

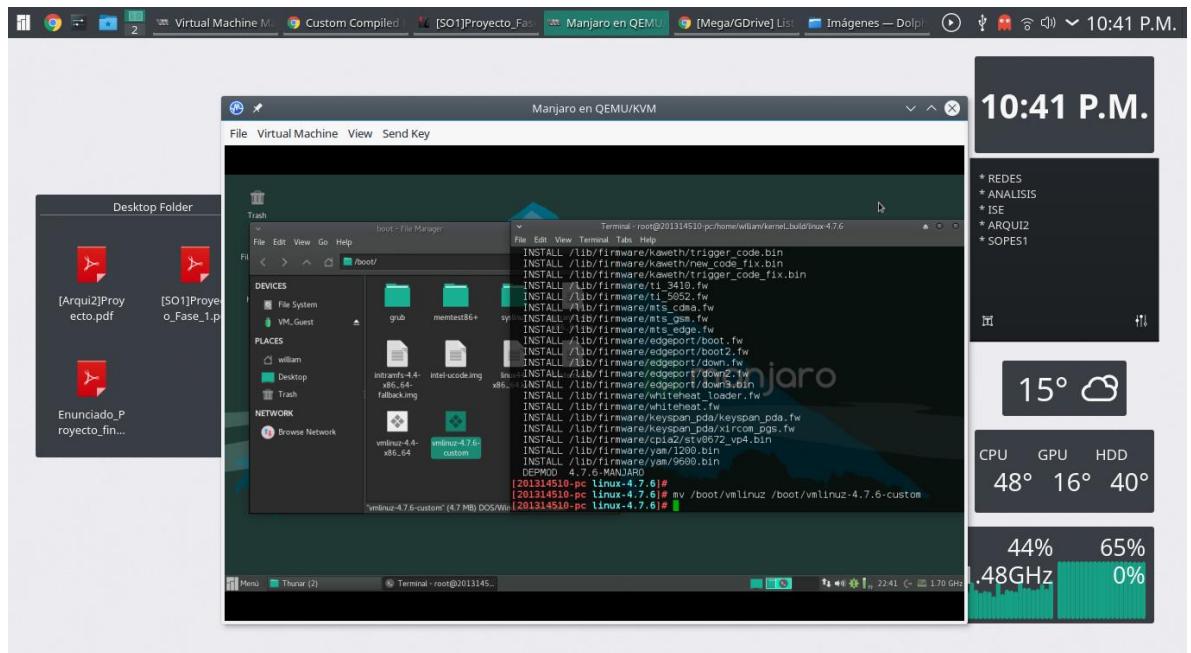
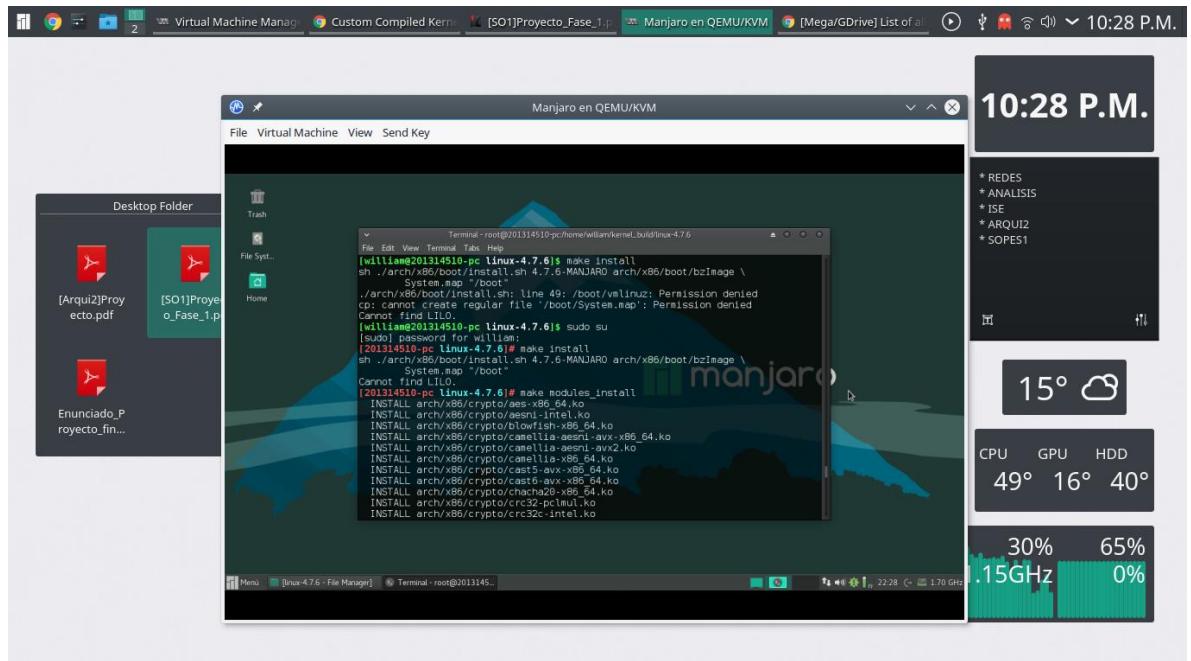




4. # sudo make install



5. # sudo make install modules



f) Instalación

Al finalizar la compilación, en la carpeta /boot/ del directorio en que se definió para descomprimir el kernel, aparecerá un archivo llamado *vmlinuz*.

Este archivo es el nuevo kernel basado en la configuración realizada previmente. Para que el sistema operativo reconozca el nuevo kernel y lo pueda montar, se siguen los siguientes pasos:

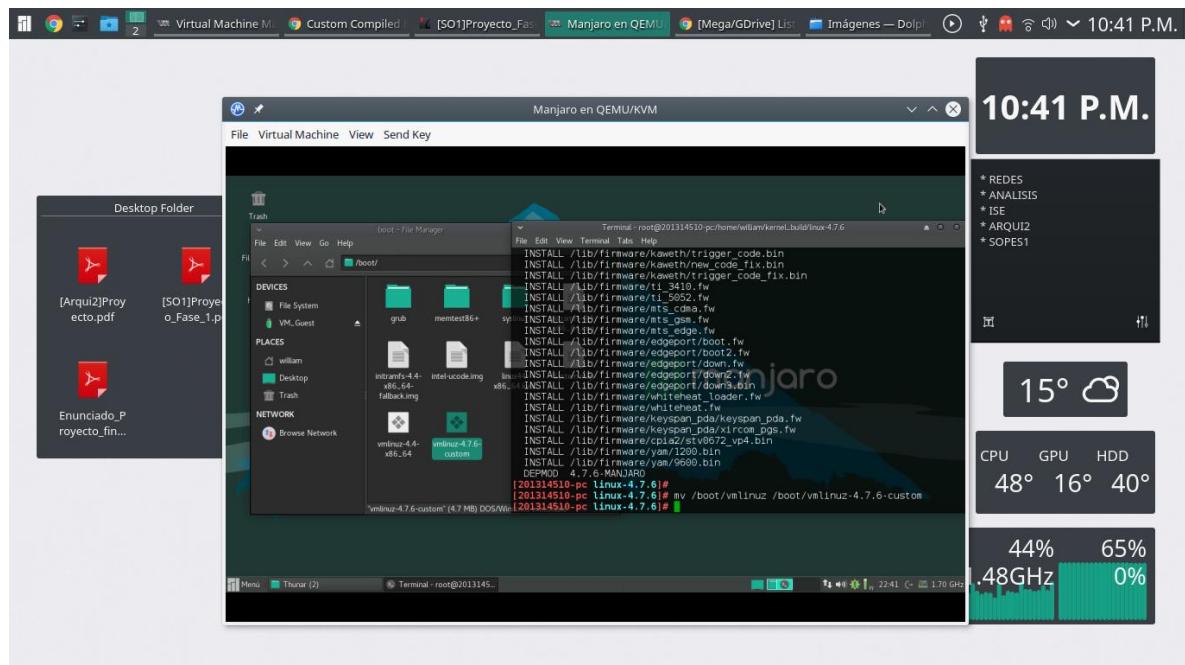
1. Mover el Kernel

Se debe colocar el nuevo kernel en la carpeta /boot/ del sistema de archivos del equipo y se recomienda cambiarle nombre respecto a la versión de kernel compilada.

El nombre del kernel debe apegarse a la convención *vmlinuz-x.x.x-[cualquier_caracter]*. Y debe iniciar con los caracteres “vmlinuz” para ser reconocido por el sistema al arrancar el equipo.

Para realizarlo se debe ejecutar el comando siguiente:

```
# sudo /boot/vmlinuz /boot/ vmlinuz-x.x.x-[cualquier_caracter]
```



2. Generar INITRAMFS

Para que el kernel sea montado correctamente al inicio debe estar acompañado de una imagen *Initial RAM File System*, lo que ayuda a que el kernel reconozca las particiones del sistema operativo.

Antes de generar este archivo, se debe ir a `/lib/modules/` en el sistema de archivos y verificar el nombre de la carpeta que contenga la versión del kernel compilado, con la convención `x.x.x[-mas_caracteres]`.

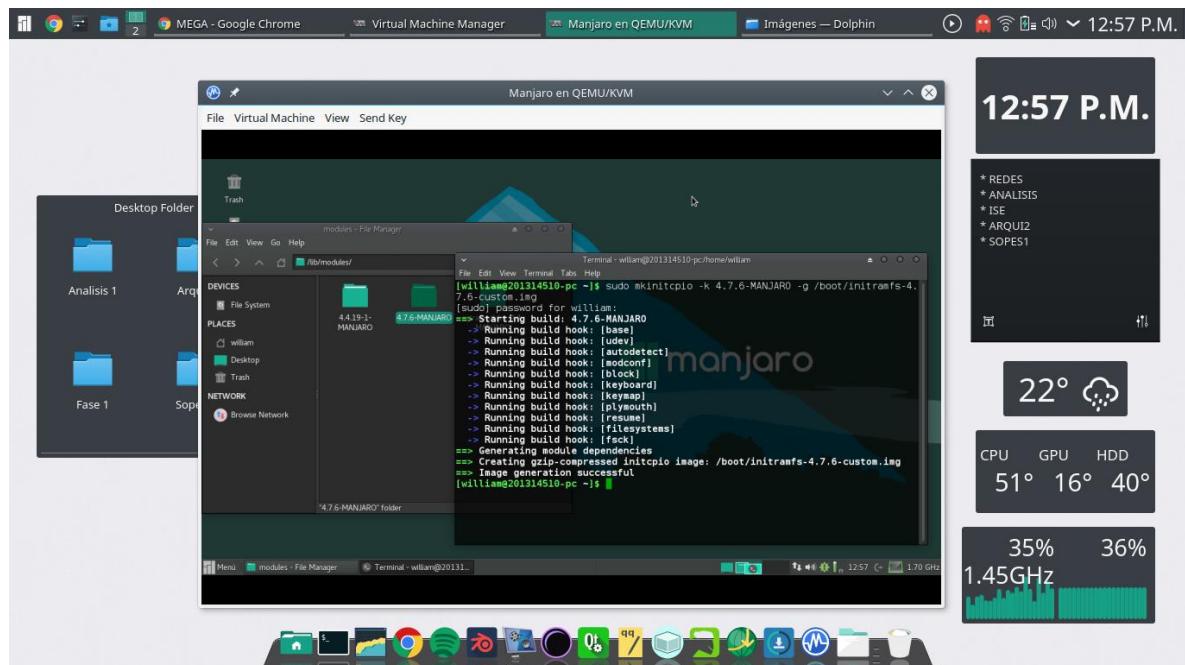
También, antes se debe definir el nombre del archivo initramfs, lo cual es de suma importancia para que esté vinculado al archivo del kernel. Por ello se debe colocar un nombre con la convención `initramfs[n_kernel]`, donde `n_kernel` no incluye los caracteres “vmlinuz”.

Ahora teniendo el nombre a colocar y la ruta de la carpeta de módulos, se puede generar el archivo initramfs desde la consola con el comando:

```
# sudo mkinitcpio -k x.x.x[-mas_caracteres] -g initramfs[n_kernel]
```

Por ejemplo:

- El kernel tiene el nombre vmlinuz-4.7.6-custom
- La carpeta de módulos corresponde a la `/4.7.6-MANJARO/`
- La imagen INITRAMFS se debe nombrar `initramfs-4.7.6-custom.img`.

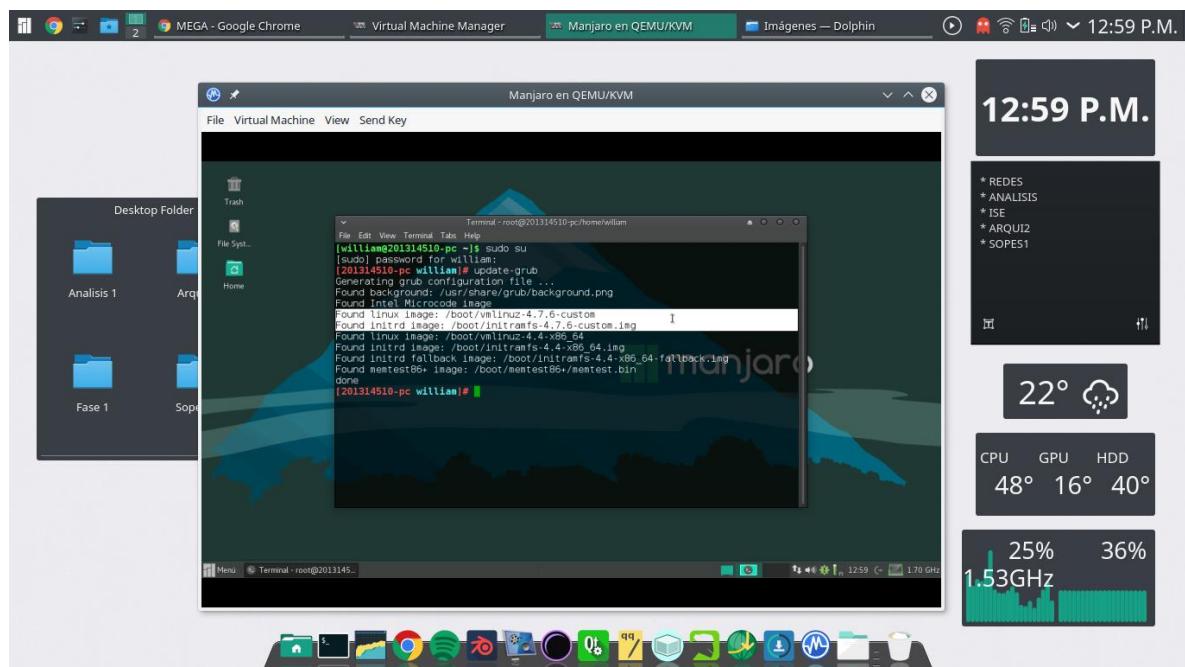


3. Gestor de Arranque

Finalmente solo queda actualizar el gestor de arranque del sistema. En este folleto se detalla utilizando como gesto a GRUB. Para ello solo se ejecuta el comando:

```
# sudo update-grub
```

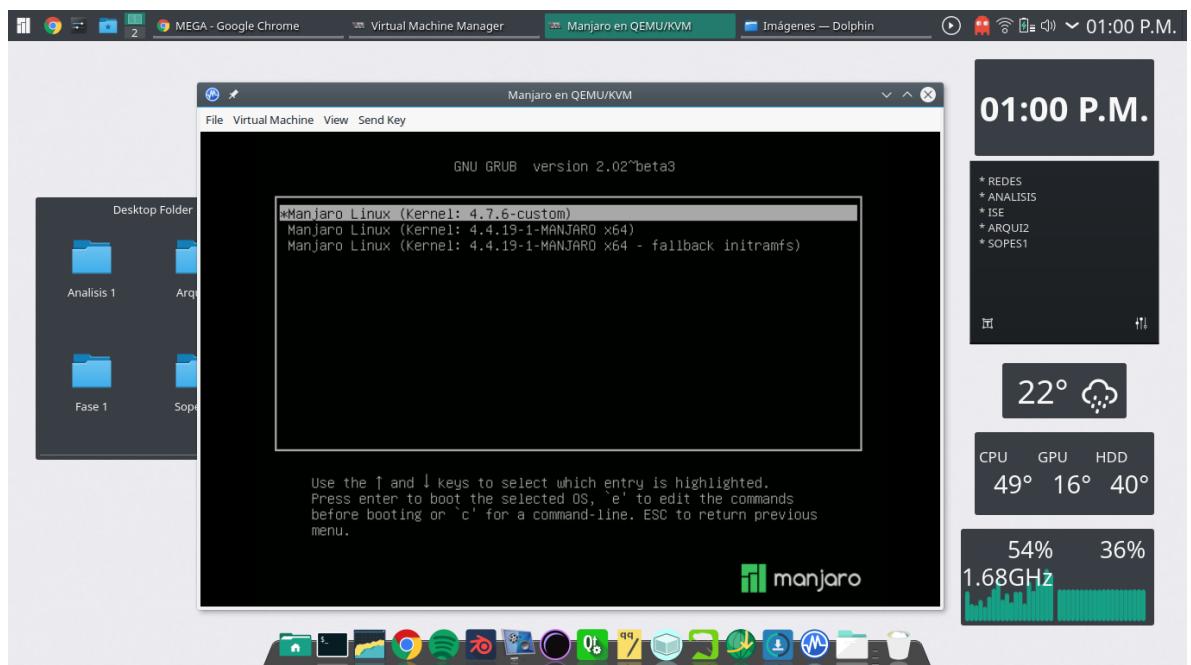
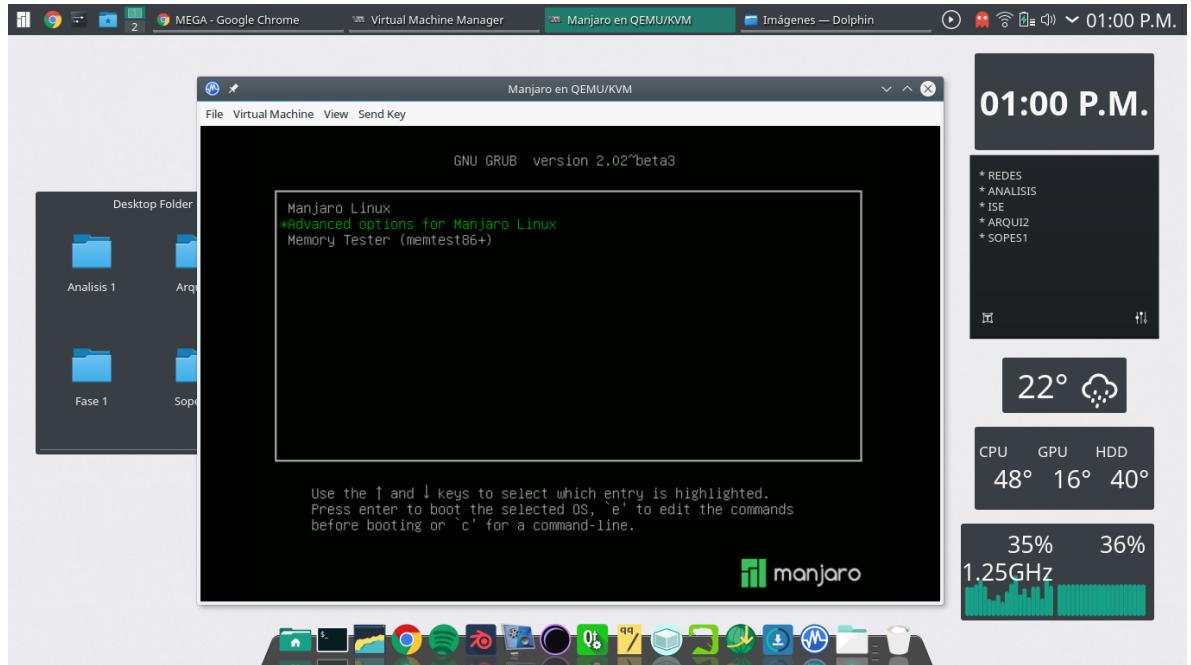
GRUB debe mostrar que encontró el archivo del kernel compilado y el archivo initramfs.

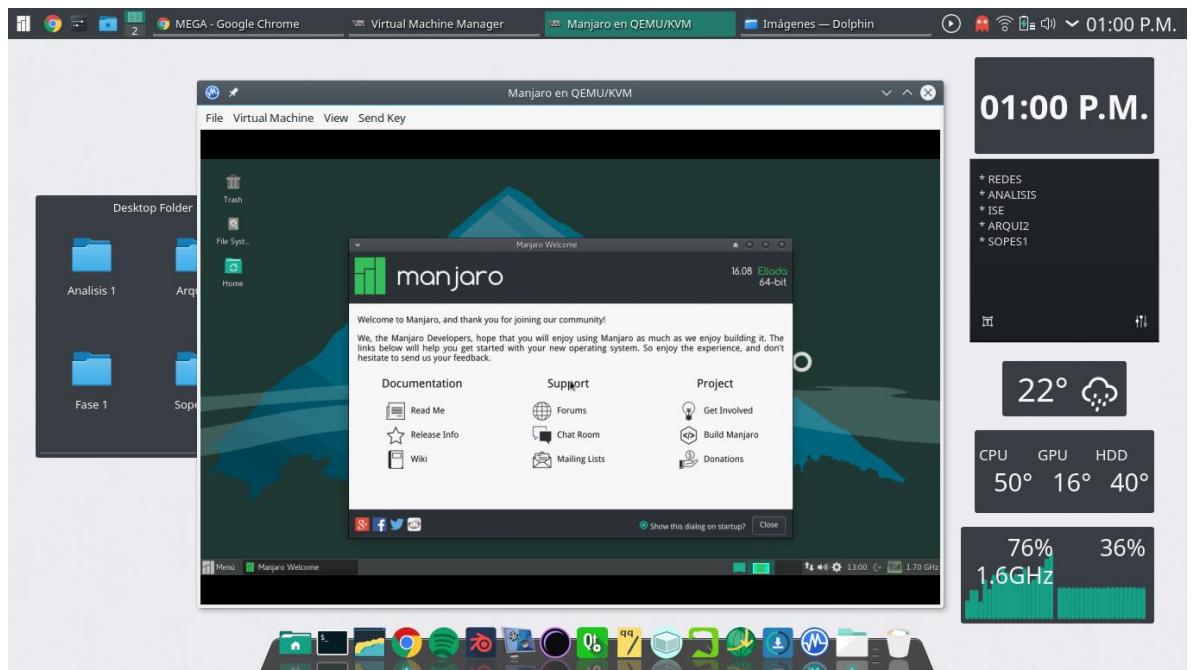
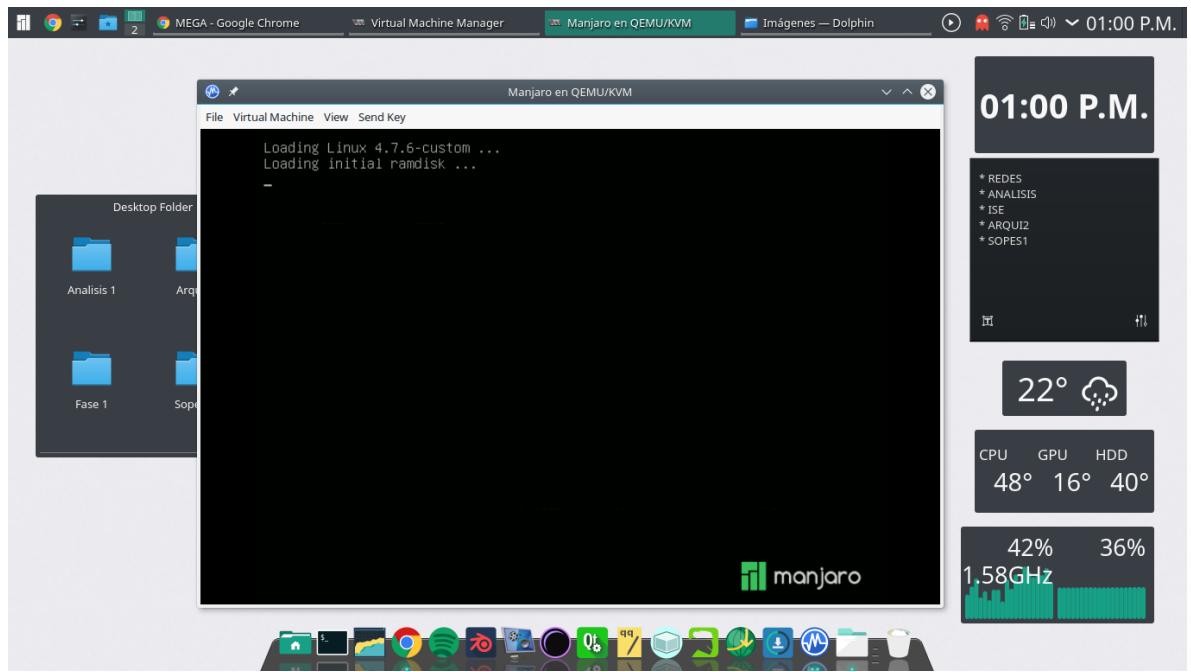


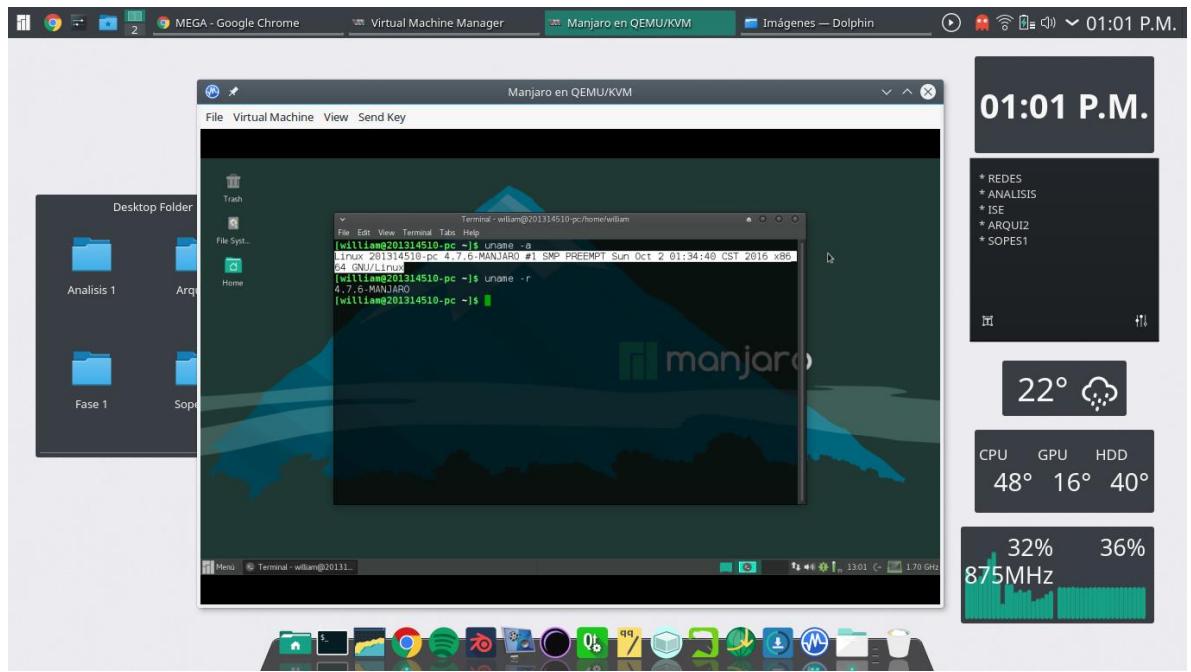
g) Uso

Para demostrar el uso del kernel compilado en este tutorial, se muestra el gestor de arranque, la selección del kernel compilado y dentro del sistema operativo, información técnica del mismo.

Si se compara la información de este kernel motando con el que se muestra en la sección de Preparación, se puede notar que no son el mismo y que el autor del segundo corresponde al usuario visible en la consola de mandos.







IV. Contacto

Nombre: William Fernando Valladares Muñoz

Ocupación: Estudiante Ingeniería en Sistemas en USAC

Correo: willfer9494@hotmail.com