
Manual Practica 3

Julia Argentina Sierra Herrera

Carné 201503865

Contenido

Requerimientos	3
¿Qué es KVM?	4
Pasos para la instalación	5
Creación de una maquina virtual	9
Compilación e instalación de kernel	14

Requerimientos

El manual fue realizado para una instalación en Ubuntu.

Almacenamiento: 600mb

Memoria RAM: 256Mb

Procesador: Disponer de extensiones para virtualización.

¿Qué es KVM?

La máquina virtual basada en el kernel (KVM) es una tecnología de virtualización de open source integrada a Linux®. En concreto, con KVM puede convertir a Linux en un hipervisor que permite que una máquina de host ejecute entornos virtuales múltiples y aislados llamados máquinas virtuales (VM) o huéspedes.

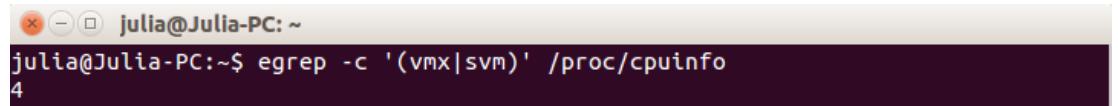
KVM convierte a Linux en un hipervisor de tipo 1 (sin sistema operativo). Todos los hipervisores necesitan algunos componentes al nivel del sistema operativo (por ejemplo, administrador de memoria, planificador de procesos, pila de entrada o salida (E/S), controladores de dispositivos, gestión de seguridad, pila de red y más) para ejecutar las máquinas virtuales. KVM cuenta con todos estos componentes porque es parte del kernel de Linux. Cada máquina virtual se implementa como un proceso regular de Linux, programada por el planificador estándar de Linux con hardware virtual dedicado como tarjeta de red, adaptador de gráficos, CPU, memoria y discos.

Pasos para la instalación

Paso 1:

Antes de instalar KVM se debe de comprobar que el sistema es compatible con KVM, para ello se necesita saber si la computadora admite VMX y virtualización de hardware SVM, para lo cual se utiliza el comando.

```
$ egrep -c '(vmx|svm)' /proc/cpuinfo
```

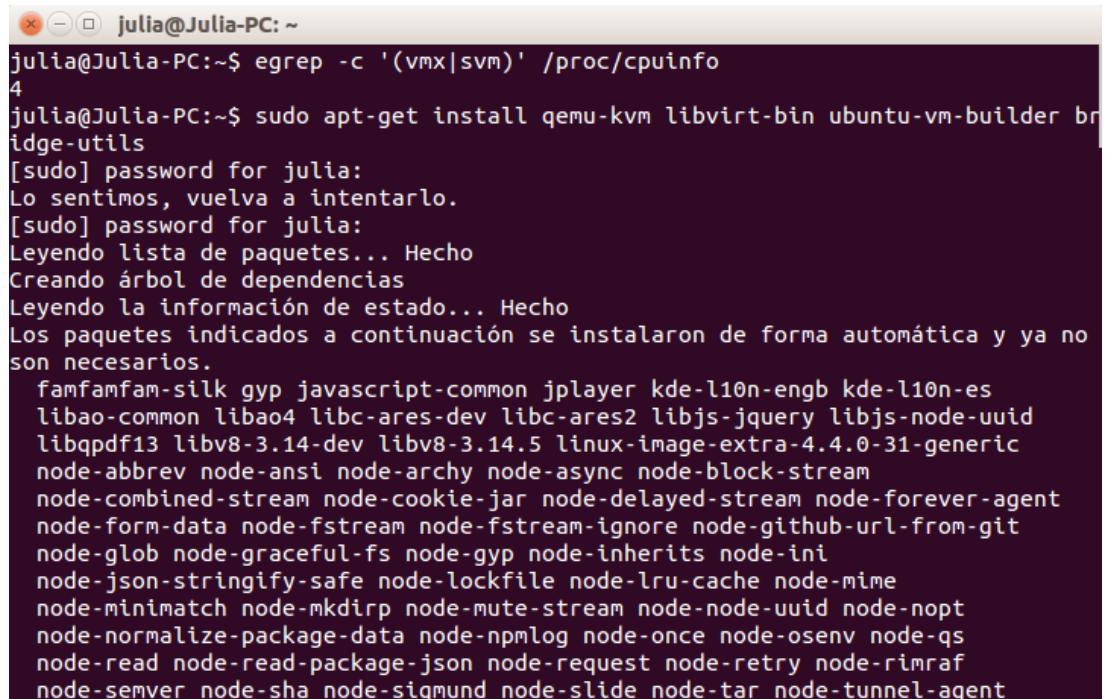


```
julia@Julia-PC:~$ egrep -c '(vmx|svm)' /proc/cpuinfo
4
```

Si el resultado que nos da es 0 entonces su maquina no soporta virtualización, ahora si es mayor o igual a 1 su computadora es apta para la virtualización.

Para la instalación de KVM se necesita instalar los paquetes que son kvm, libvirt-bin, Ubuntu-vm-builder y bridge-utils, para realizarlo usaremos el siguiente comando.

```
$ sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils
```



```
julia@Julia-PC:~$ egrep -c '(vmx|svm)' /proc/cpuinfo
4
julia@Julia-PC:~$ sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder br
idge-utils
[sudo] password for julia:
Lo sentimos, vuelva a intentarlo.
[sudo] password for julia:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
famfamfam-silk gyp javascript-common jplayer kde-l10n-engb kde-l10n-es
libao-common libao4 libc-ares-dev libc-ares2 libjs-jquery libjs-node-uuid
libqpdf13 libv8-3.14-dev libv8-3.14.5 linux-image-extra-4.4.0-31-generic
node-abbrev node-ansi node-archy node-async node-block-stream
node-combined-stream node-cookie-jar node-delayed-stream node-forever-agent
node-form-data node-fstream node-fstream-ignore node-github-url-from-git
node-glob node-graceful-fs node-gyp node-inherits node-inti
node-json-stringify-safe node-lockfile node-lru-cache node-mime
node-minimatch node-mkdirp node-mute-stream node-node-uuid node-nopt
node-normalize-package-data node-npmlog node-once node-osenv node-qs
node-read node-read-package-json node-request node-retry node-rimraf
node-semver node-sha node-sigmund node-slide node-tar node-tunnel-agent
```

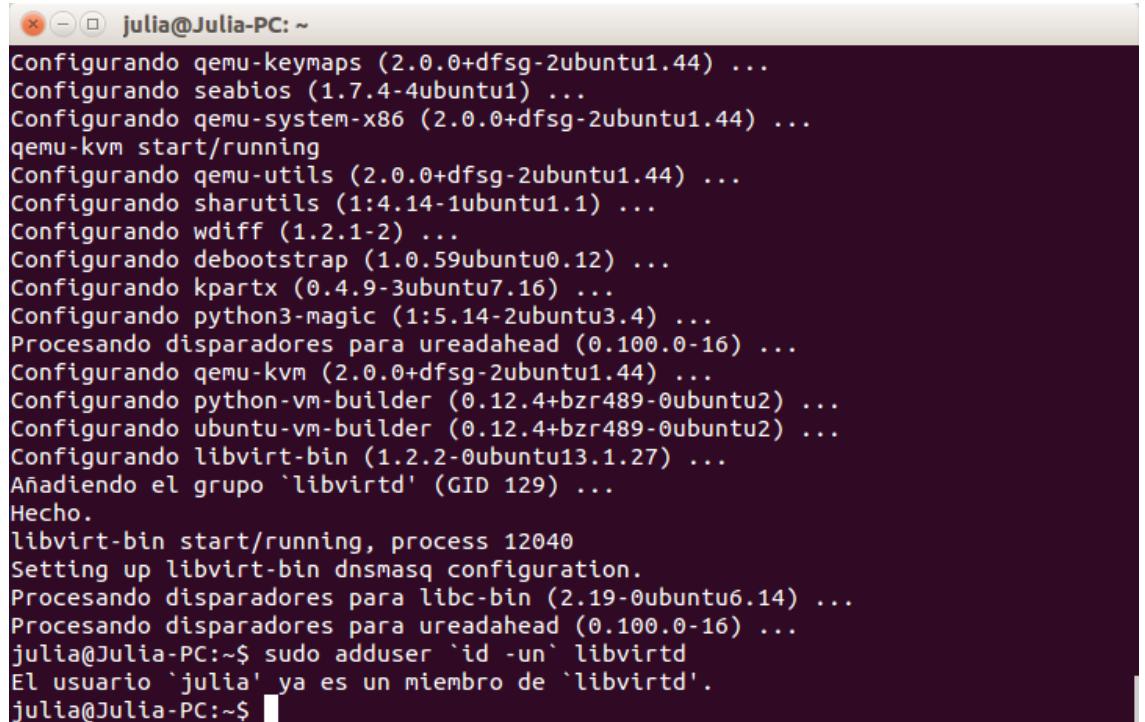
Lo que necesita permisos de administrador lo cual nos pedira la contraseña de administrador.

Paso 2

Colocar el usuario en el grupo libvирtd

Para que el KVM funcione de la manera correcta el usuario debe de estar en el grupo de libvирtd, para eso utilizaremos el siguiente comando.

```
$ sudo adduser `id -un` libvирtd
```



The screenshot shows a terminal window titled "julia@Julia-PC: ~". The user has run the command "sudo adduser `id -un` libvирtd". The terminal displays the output of the package configuration process, which includes dependencies like qemu-keymaps, seabios, qemu-system-x86, qemu-kvm, qemu-utils, sharutils, wdiff, debootstrap, kpartx, python3-magic, and others. It also shows the creation of the libvирtd group (GID 129) and the addition of the user 'julia' to it. The final message indicates that 'julia' is already a member of the libvирtd group.

```
julia@Julia-PC: ~
Configurando qemu-keymaps (2.0.0+dfsg-2ubuntu1.44) ...
Configurando seabios (1.7.4-4ubuntu1) ...
Configurando qemu-system-x86 (2.0.0+dfsg-2ubuntu1.44) ...
qemu-kvm start/running
Configurando qemu-utils (2.0.0+dfsg-2ubuntu1.44) ...
Configurando sharutils (1:4.14-1ubuntu1.1) ...
Configurando wdiff (1.2.1-2) ...
Configurando debootstrap (1.0.59ubuntu0.12) ...
Configurando kpartx (0.4.9-3ubuntu7.16) ...
Configurando python3-magic (1:5.14-2ubuntu3.4) ...
Procesando disparadores para ureadahead (0.100.0-16) ...
Configurando qemu-kvm (2.0.0+dfsg-2ubuntu1.44) ...
Configurando python-vm-builder (0.12.4+bzr489-0ubuntu2) ...
Configurando ubuntu-vm-builder (0.12.4+bzr489-0ubuntu2) ...
Configurando libvirt-bin (1.2.2-0ubuntu13.1.27) ...
Añadiendo el grupo 'libvirt-bin' (GID 129) ...
Hecho.
libvirt-bin start/running, process 12040
Setting up libvirt-bin dnsmasq configuration.
Procesando disparadores para libc-bin (2.19-0ubuntu6.14) ...
Procesando disparadores para ureadahead (0.100.0-16) ...
julia@Julia-PC: ~$ sudo adduser `id -un` libvirt-bin
El usuario 'julia' ya es un miembro de 'libvirt-bin'.
julia@Julia-PC: ~$
```

En este caso al instalar las dependencias se visualiza como el usuario es agregado al grupo libvирtd, por lo que al colocar el comando nos dice que el usuario ya es miembro de libvирtd.

Paso 3

Instalar la interfaz grafica de KVM

Ahora que ya esta instalado se necesita tener una interfaz grafica para lo que necesitamos instalar virt-manager, para ello utilizaremos el siguiente comando.

```
$sudo apt-get install virt-manager
```

```
julia@Julia-PC: ~
Julia@Julia-PC:~$ sudo adduser `id -un` llvmbirtd
El usuario 'Julia' ya es un miembro de 'llvmbirtd'.
julia@Julia-PC:~$ sudo apt-get install virt-manager
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
  fanfamfan-stlk gyp javascript-common jplayer kde-l10n-engb kde-l10n-es
  liblao-common liblao4 libc-ares-dev liblbc-aresz libljs-jquery libljs-node-uid
  liblbpdf13 liblv8-3.14-dev liblv8-3.14.5 linux-image-extra-4.4.0-31-generic
  node-abbrev node-ansi node-archy node-async node-block-stream
  node-combined-stream node-cookie-jar node-delayed-stream node-forever-agent
  node-form-data node-fstream node-fstream-ignore node-github-url-from-git
  node-glob node-graceful-fs node-gyp node-inherits node-ini
  node-json-stringify-safe node-lockfile node-lru-cache node-mime
  node-minimatch node-mkdirp node-mute-stream node-node-uid node-nop
  node-normalize-package-data node-npmlog node-once node-osenv node-q
  node-read-read-package-jason node-request node-retry node-rimraf
  node-semver node-sha node-signum node-slide node-tar node-tunnel-agent
  node-which vorbis-tools
Use 'apt-get autoremove' to remove them...
se instalarán los siguientes paquetes extras:
libbonoboi2-0 libbonoboi2-common libbonoboui2-0 libbonoboui2-common
libglade2-0 libgnome2-0 libgnome2-bin libgnome2-common libgnomecanvas2-0
libgnomecanvas2-common libgnomeui-0 libgnomeui-common libgnomevfs2-0
libgnomevfs2-common libgtk-vnc-1.0-0 libgvnc-1.0-0 libt1l-common libt1l0
liborbit-2-0 liborbit2 libvte-common libvte9 python-appindicator
python-glade2 python-gnome python-gtk+2 vnc python-pyorbit python-urllibgrabb
python-vte virtinst
Paquetes sugeridos:
libbonoboi2-bin desktop-base libgnomevfs2-bin libgnomevfs2-extra gamin fam
gnome-mime-data python-gtk2-doc python-gnome2-doc python-pyorbit-dbg
virt-vviewer python-guestfs python-spice-client-gtk
Se instalarán los siguientes paquetes NUEVOS:
libbonoboi2-0 libbonoboi2-common libbonoboui2-0 libbonoboui2-common
libglade2-0 libgnome2-0 libgnome2-bin libgnome2-common libgnomecanvas2-0
libgnomecanvas2-common libgnomeui-0 libgnomeui-common libgnomevfs2-0
libgnomevfs2-common libgtk-vnc-1.0-0 libgvnc-1.0-0 libt1l-common libt1l0
liborbit-2-0 liborbit2 libvte-common libvte9 python-appindicator
python-glade2 python-gnome python-gtk+2 vnc python-pyorbit python-urllibgrabb
```

Para verificar la instalación utilizaremos el comando

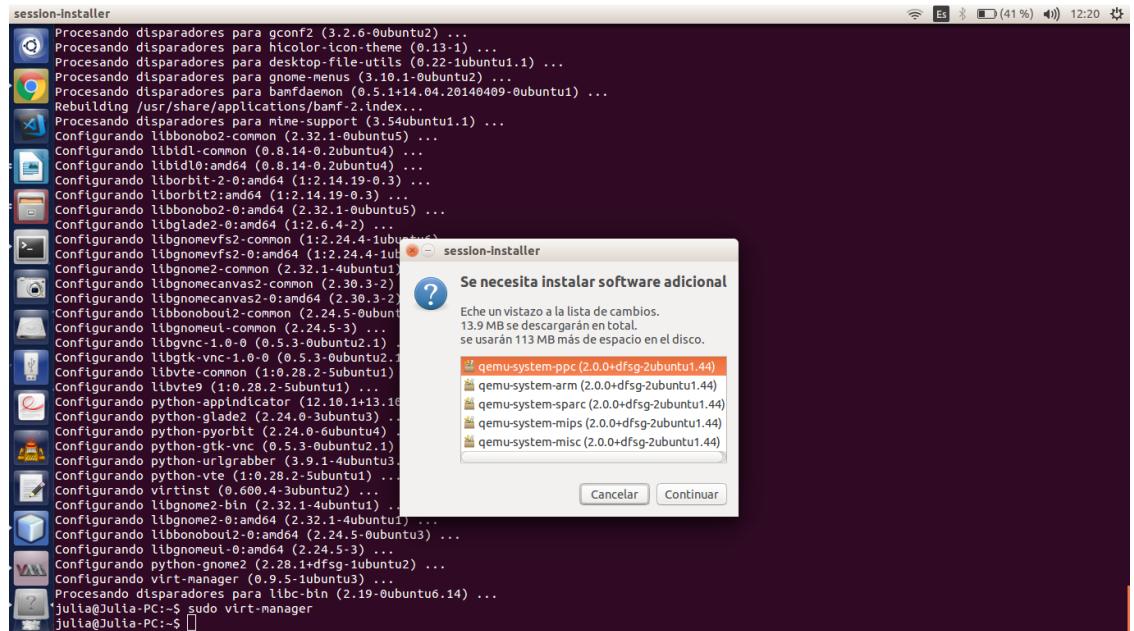
\$Sudo virt-manager

The screenshot shows a Linux desktop environment with several windows open. In the foreground, a terminal window displays a log of package installations:

```
Procesando disparadores para gconf2 (3.2.6-0ubuntu2) ...
Procesando disparadores para libgnome-themes (0.18.1-1) ...
Procesando disparadores para desktop-file-utils (0.22-1ubuntu1.1) ...
Procesando disparadores para gnome-menu (3.10.1-0ubuntu1.1) ...
Procesando disparadores para bamfdaemon (0.5.1+14.04.20140409-0ubuntu1) ...
Rebuilding /usr/share/applications/bamf-2.index...
Procesando disparadores para mime-support (3.54ubuntu1.1) ...
Configurando libbonoboz-common (2.24.1-0ubuntu5) ...
Configurando libltdl-common (0.8.14-0.2ubuntu4) ...
Configurando libbonoboz-0:amd64 (2.24.1-0ubuntu5) ...
Configurando libltdl-0:amd64 (0.8.14-0.2ubuntu4) ...
Configurando libbonoboz2:amd64 (1:2.24.1-0ubuntu5) ...
Configurando libbonoboz2:amd64 (2.24.1-0ubuntu5) ...
Configurando libglade2-0:amd64 (1:2.24.1-0ubuntu5) ...
Configurando libgnomevfs2-common (0.9.5-1) ...
Configurando libgnomevfs2-0:amd64 (0.9.5-1) ...
Configurando libgnomevfs2-common (0.9.5-1) ...
Configurando libgnomecanvas2-common (0.9.5-1) ...
Configurando libgnomecanvas2-0:amd64 (0.9.5-1) ...
Configurando libgnomebouilts-common (0.9.5-1) ...
Configurando libgnome-1.0-common (0.9.5-1) ...
Configurando libgvnc-1.0.0 (0.5.3-1) ...
Configurando libgvtk-vnc-1.0.0 (0.5.3-1) ...
Configurando libvte-common (1:0.28.2-1) ...
Configurando libvte9 (1:0.28.2-1) ...
Configurando python-apportindicator (0.1.1-1) ...
Configurando python-glade2 (2.24.0-1) ...
Configurando python-gtk2 (2.24.0-1) ...
Configurando python-ttk-vnc (0.9.5-1) ...
Configurando python-urllibraber (3.1-1) ...
Configurando python-vte (1:0.28.2-1) ...
Configurando virtinst (0.600.4-3ubunt) ...
Configurando libgnome2-bin (2.32.1-4ubuntu1) ...
Configurando libgnome2-0:amd64 (2.32.1-4ubuntu1) ...
Configurando libbonobouilt-0:amd64 (2.24.5-0ubuntu3) ...
Configurando libgnomeui-0:amd64 (2.24.5-3) ...
Configurando libgnomeui-common (2.24.5-3) ...
Configurando virt-nanomouse (0.9.5-1ubuntu2) ...
Procesando disparadores para libtbc-bin (2.19-0ubuntu6.14) ...
julia@Julia-PC:~$ sudo virt-manager
julia@Julia-PC:~$ [ ]
```

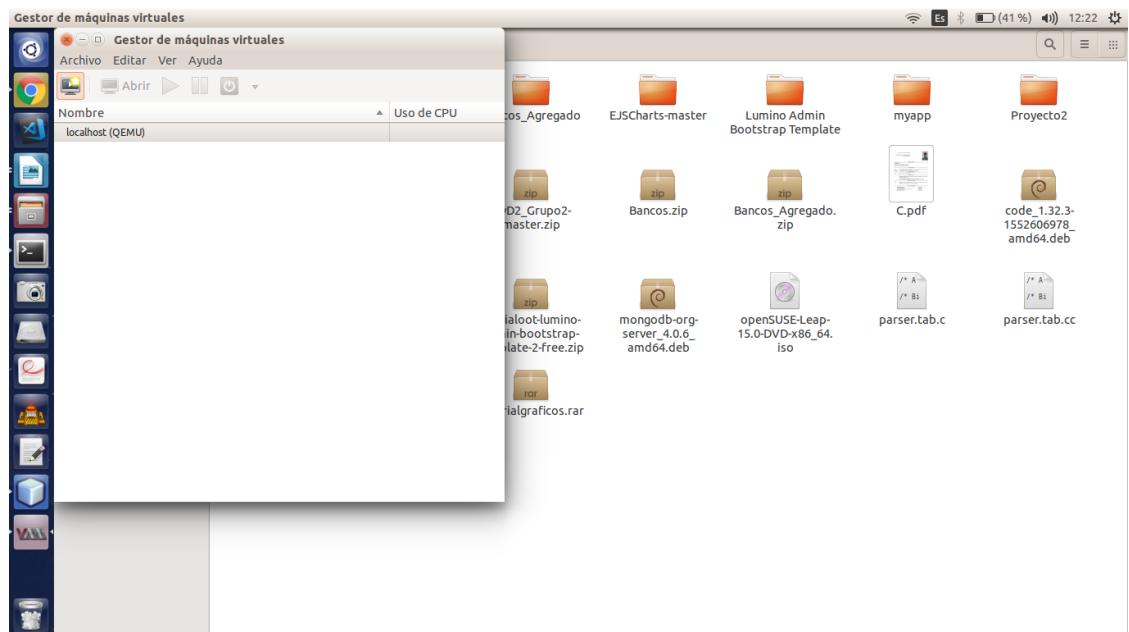
A 'Install package' dialog box is overlaid on the desktop, prompting for a package named 'qemu-system'. The dialog includes fields for 'Search' and 'Install', and a note about requiring a full system emulation binary.

Nos desplegará mensajes de paquetes que se deben instalar.



Paso 4

Ahora el virtual manager está listo para poder utilizarse.



Creación de una maquina virtual

Paso 1

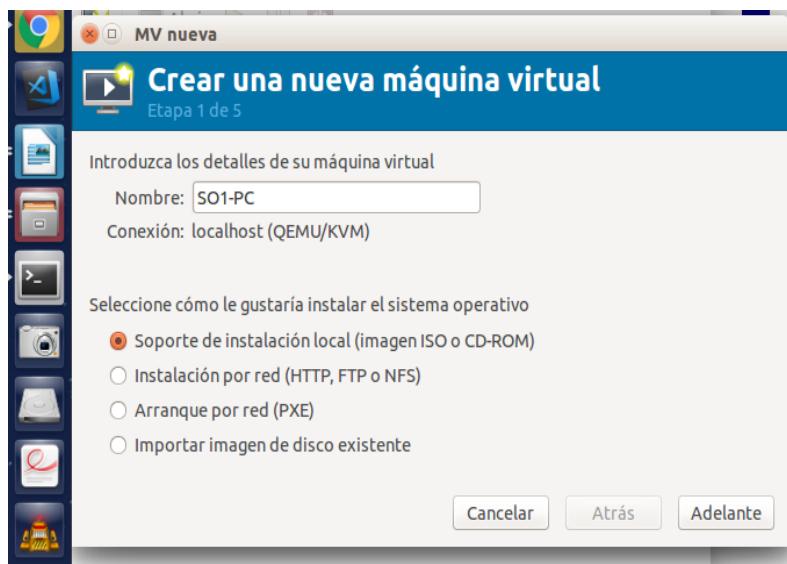
Revisar que la maquina tenga el espacio suficiente de disco duro para crear una nueva maquina virtual, se realizara una instalación de Fedora.

Paso 2

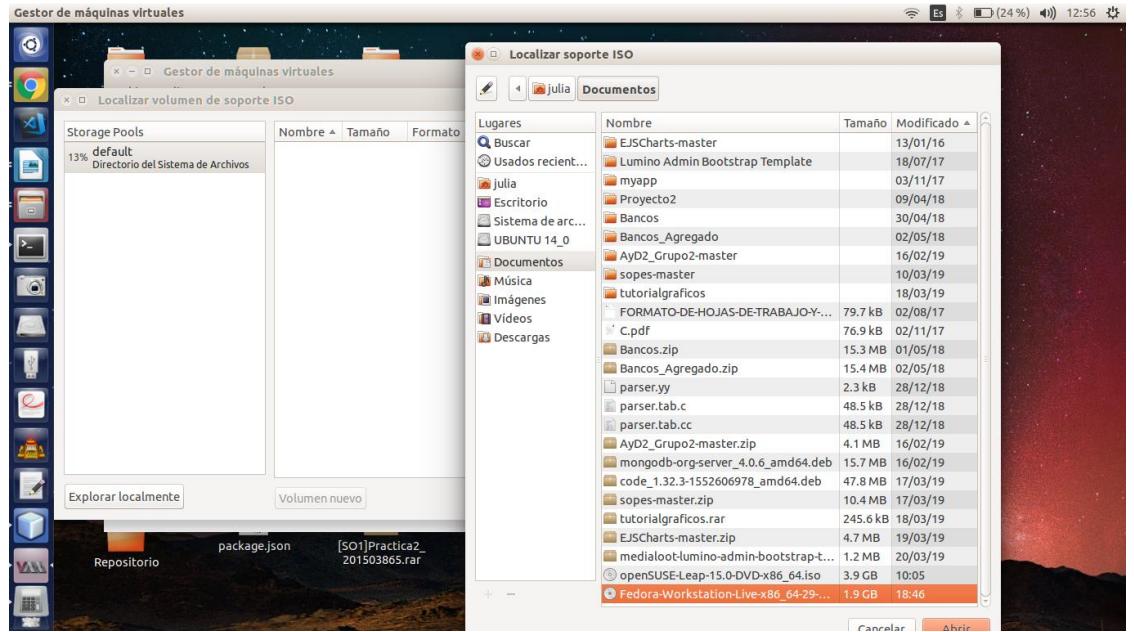
Le daremos clic al icono para crear una maquina virtual.



Nos mostrara la siguiente ventana en la que definiremos el nombre de la maquina de que manera la realizaremos que en este caso será por medio de una imagen ISO

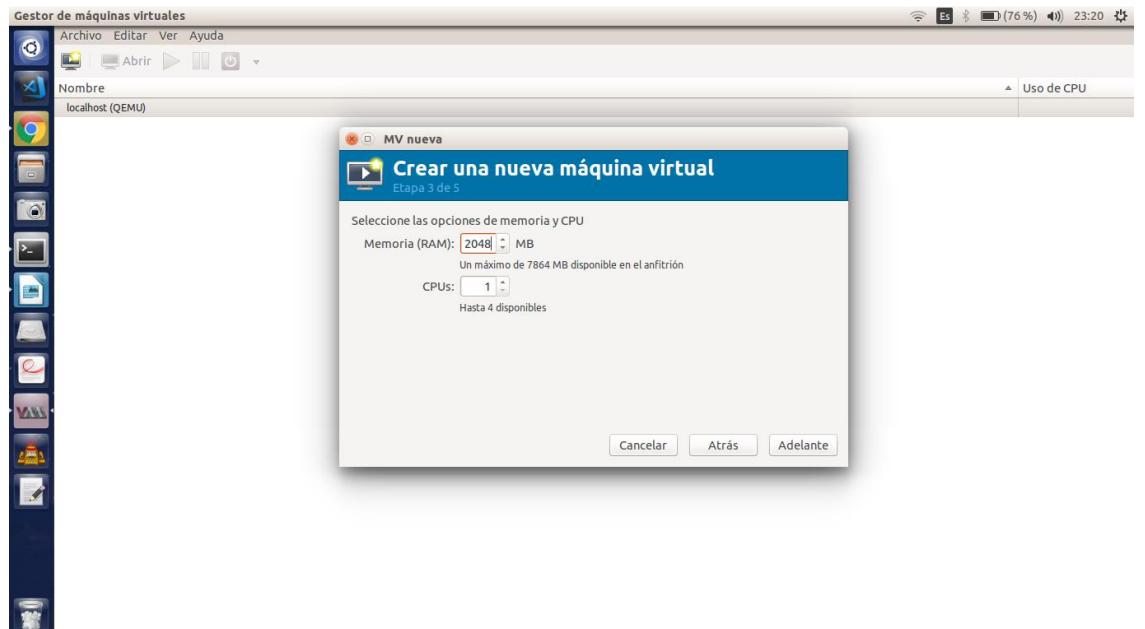


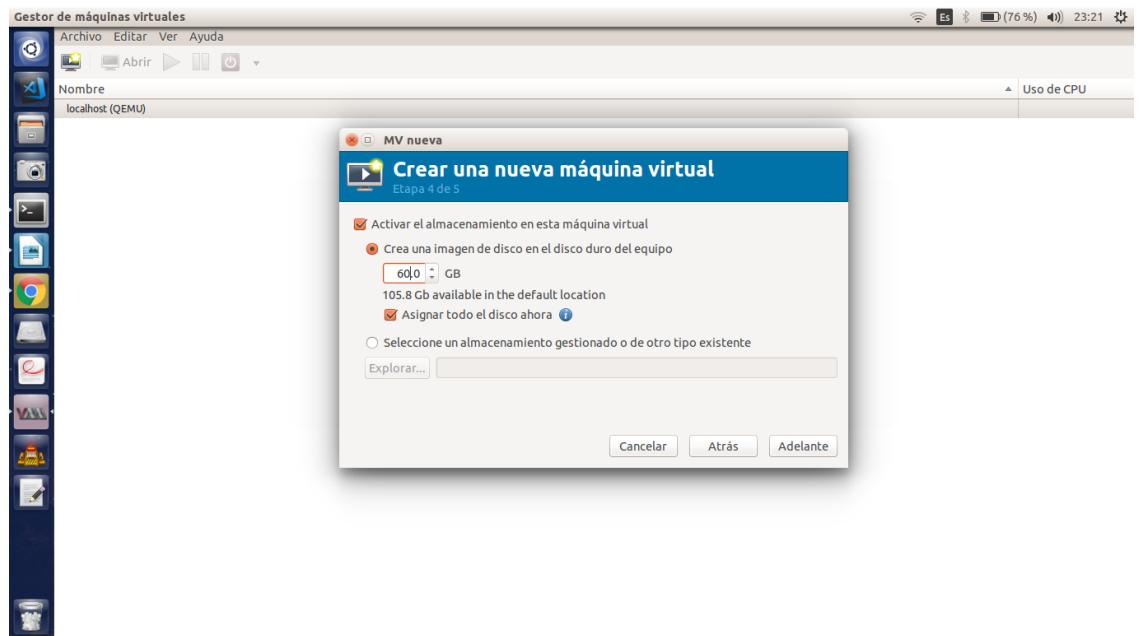
Luego de dar siguiente seleccionaremos la imagen iso desde nuestra computadora.



Paso 2

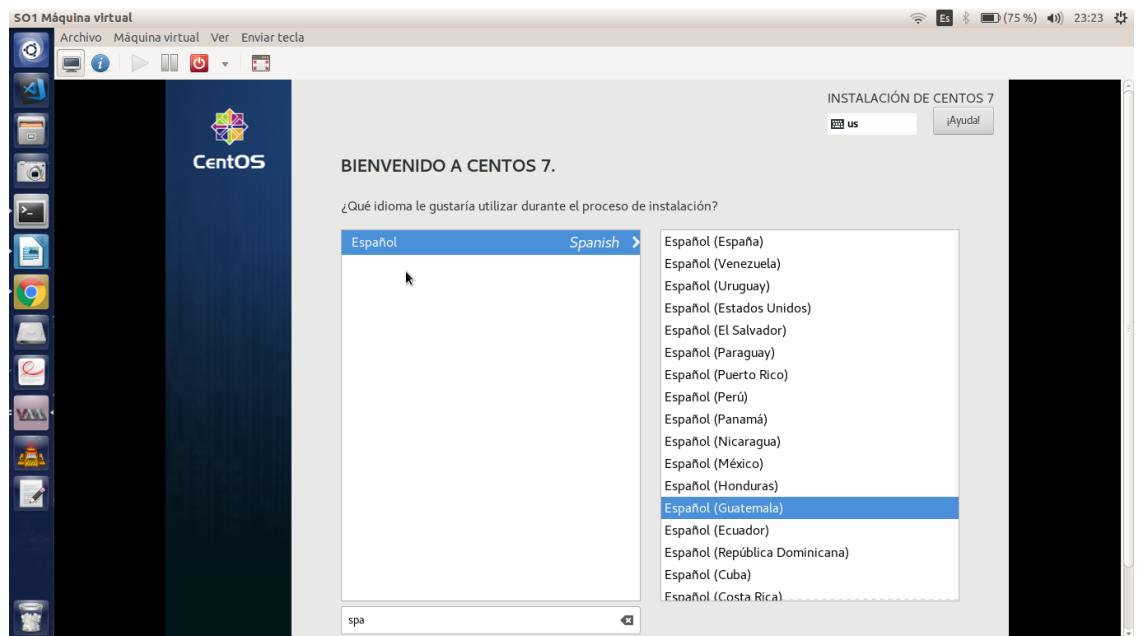
Ahora debemos de colocarle la RAM, CPU y tamaño del disco duro de nuestra nueva máquina virtual.



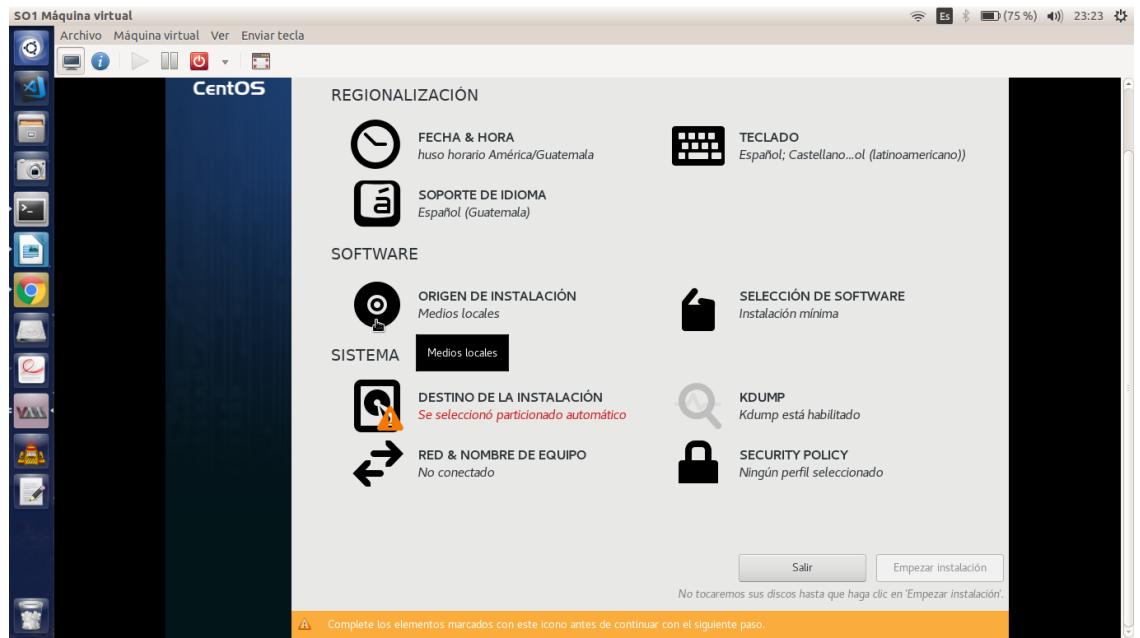


Paso 3

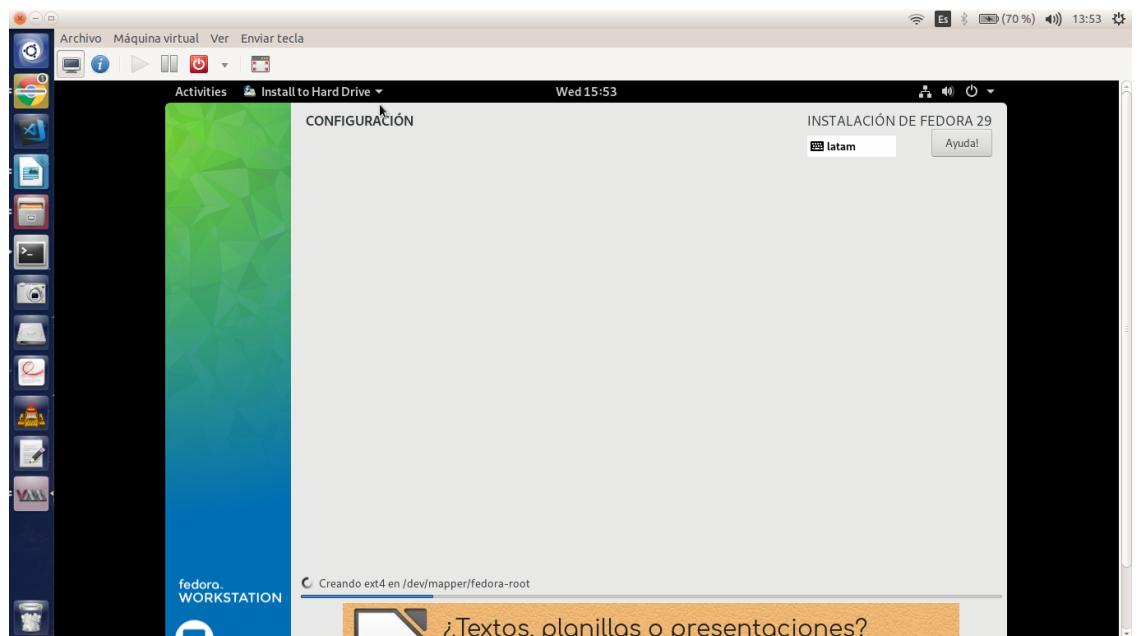
Instalación de Centos



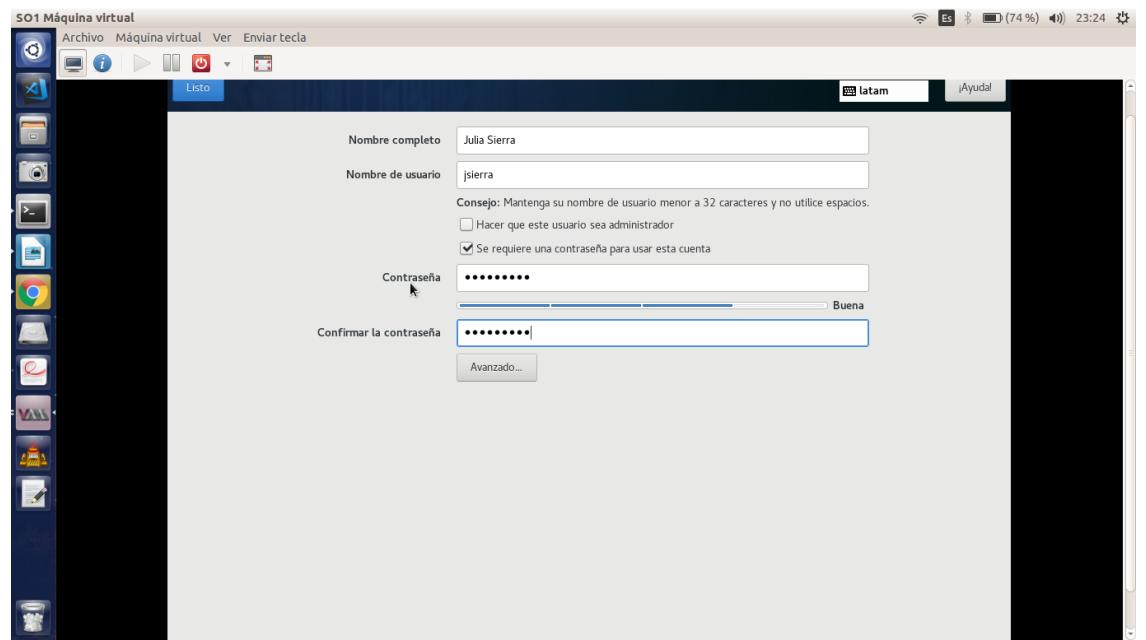
En la siguiente debemos de escoger el idioma



Seleccionar el disco duro en que deseamos instalarlo.



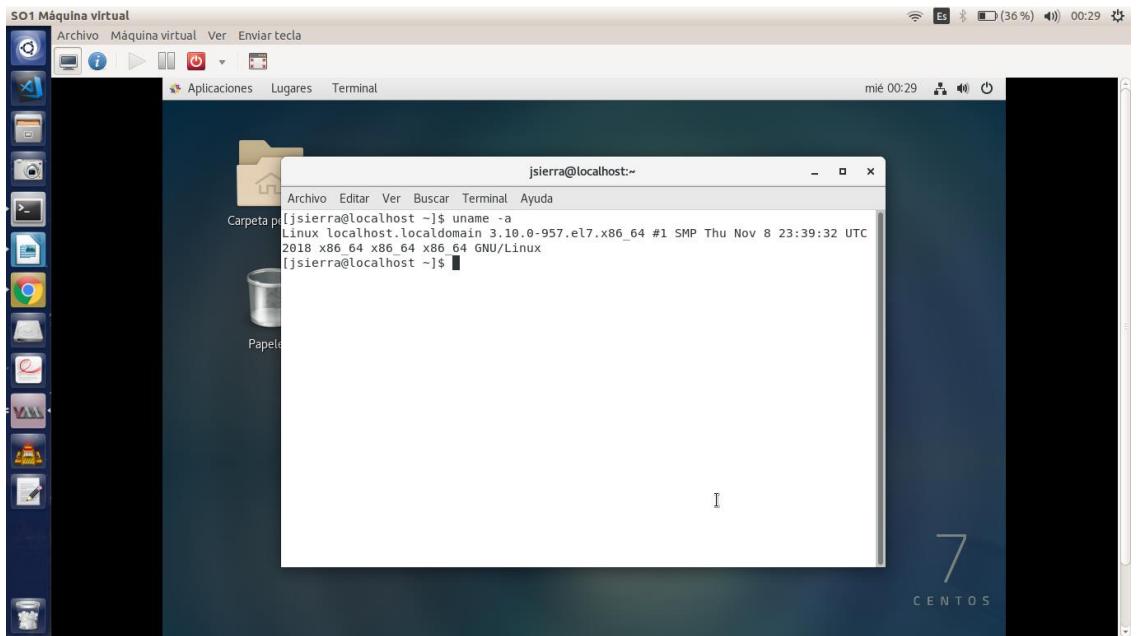
Se nos mostraran una serie de configuraciones que se deben de colocar.



Ahora se nos pedirán las credenciales que utilizara nuestra máquina. Se establecerá una contraseña para poder accesar al equipo.

Y ahora ya está finalizada nuestra instalación de Centos.

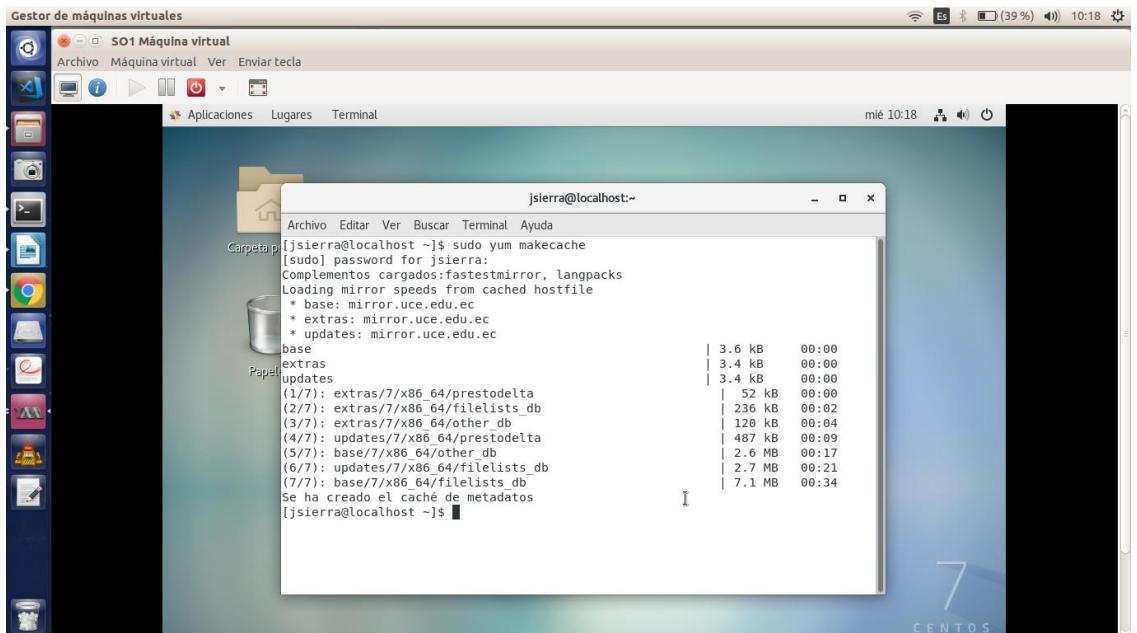
Compilación e instalación de kernel



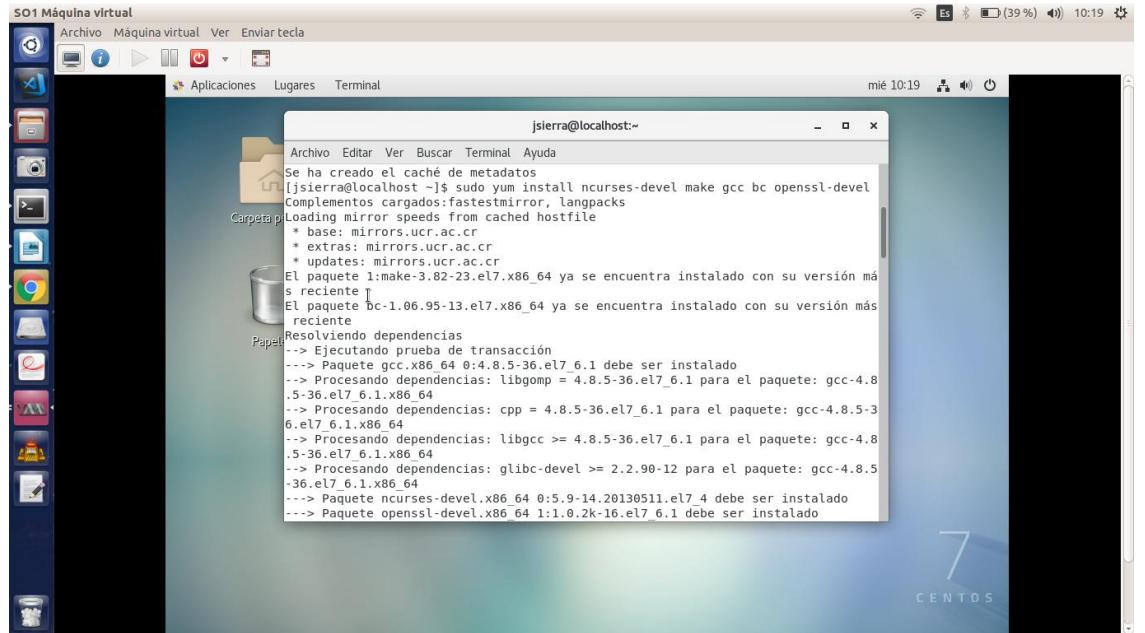
Primero se verificará la versión de kernel que posee la máquina. Para ello se utiliza el comando : **uname -a**

Paso 1

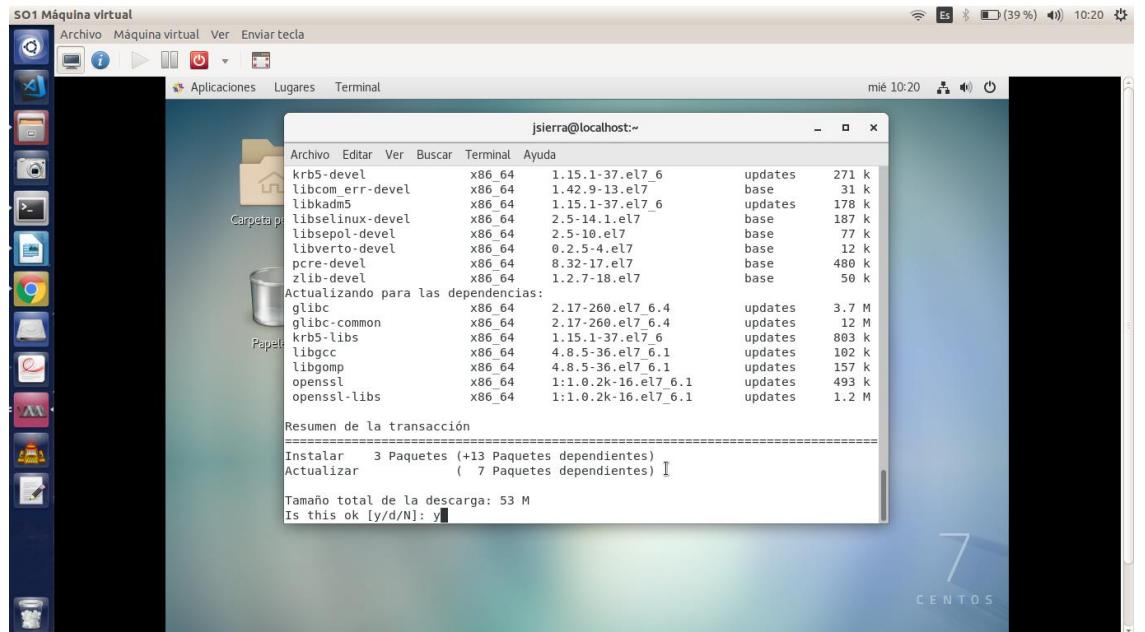
Se necesita tener instalados previamente paquetes y herramientas



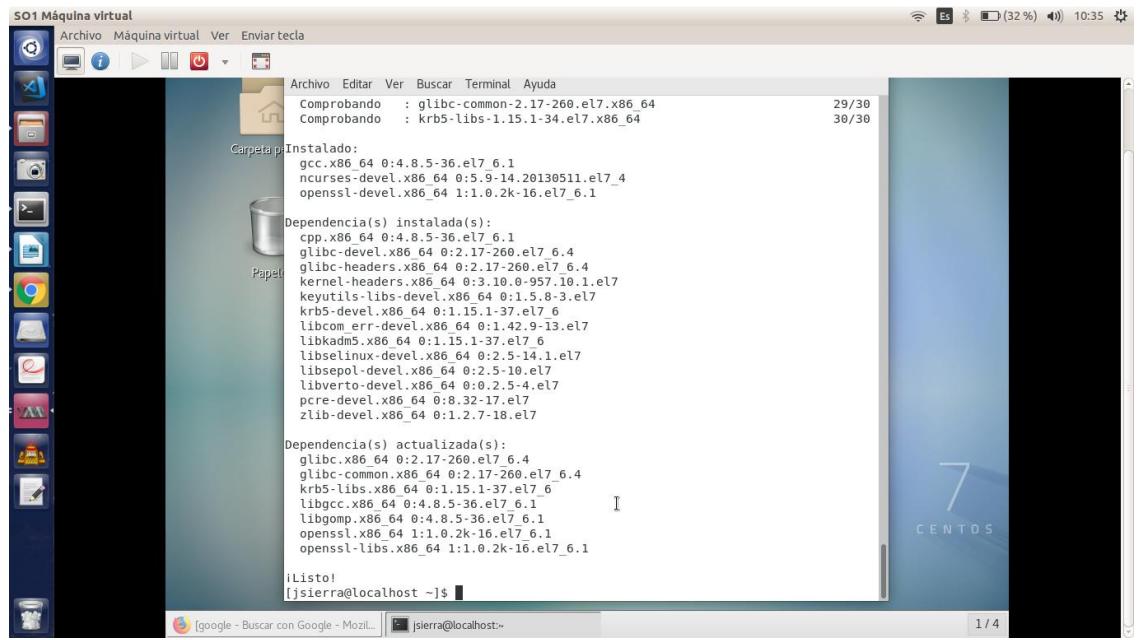
El comando **sudo yum makecache** actualizara el cache de los paquetes del sistema operativo.



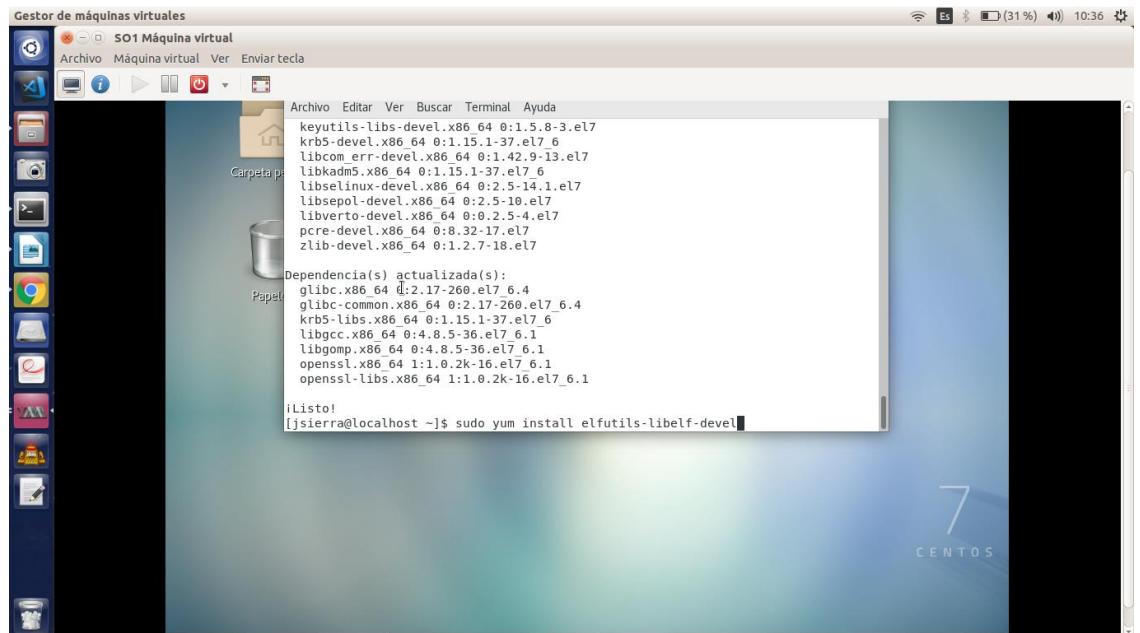
Se necesita instalar los compiladores y bibliotecas para lo que se utiliza el comando
sudo yum install ncurses-devel make gcc bc openssl-devel



Nos preguntara si estamos de acuerdo para lo que necesitamos colocar Y y luego enter.

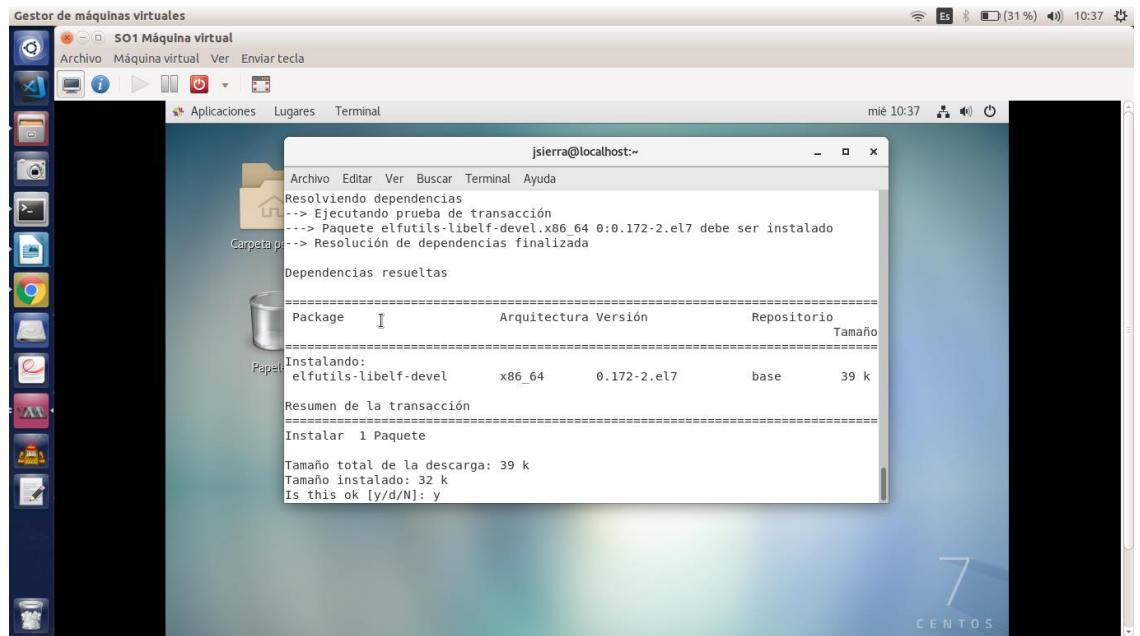


Nos mostrara todas las dependencias que han sido instaladas y actualizadas.

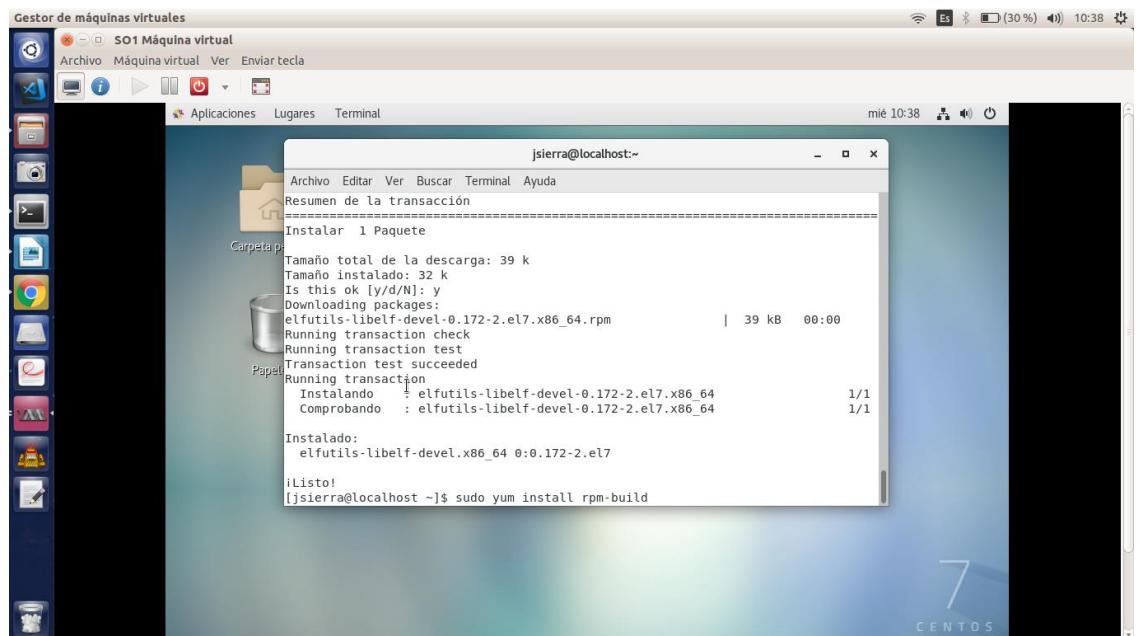


Necesitamos instalar el paquete elfutils para ello utilizaremos el siguiente comando

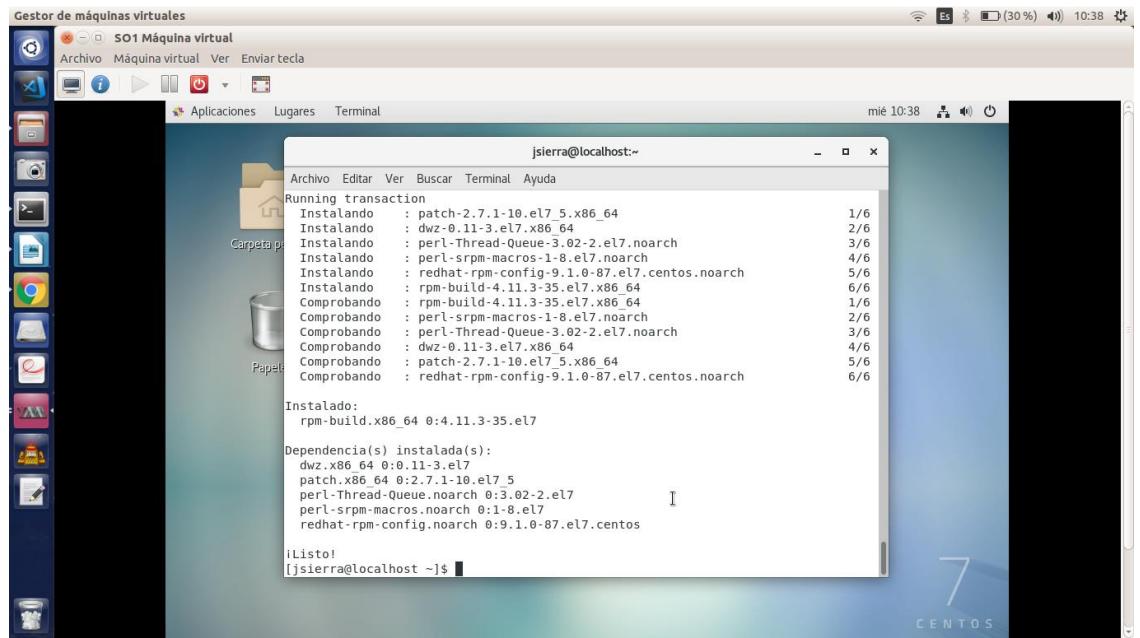
\$ sudo yum install elfutils-libelf-devel



Al igual que en el comando anterior nos preguntara si estamos de acuerdo le daremos y y enter para que siga la instalación



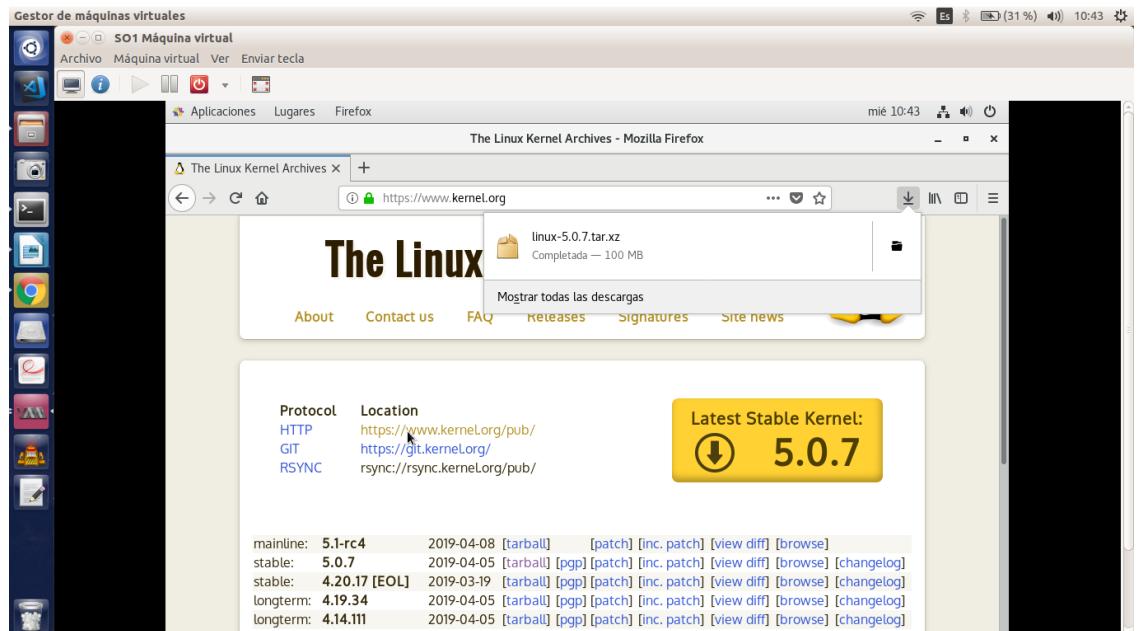
Se deberá de instalar otro paquete que será el rpm-build para lo que se utilizará el siguiente comando **sudo yum install rpm-build**



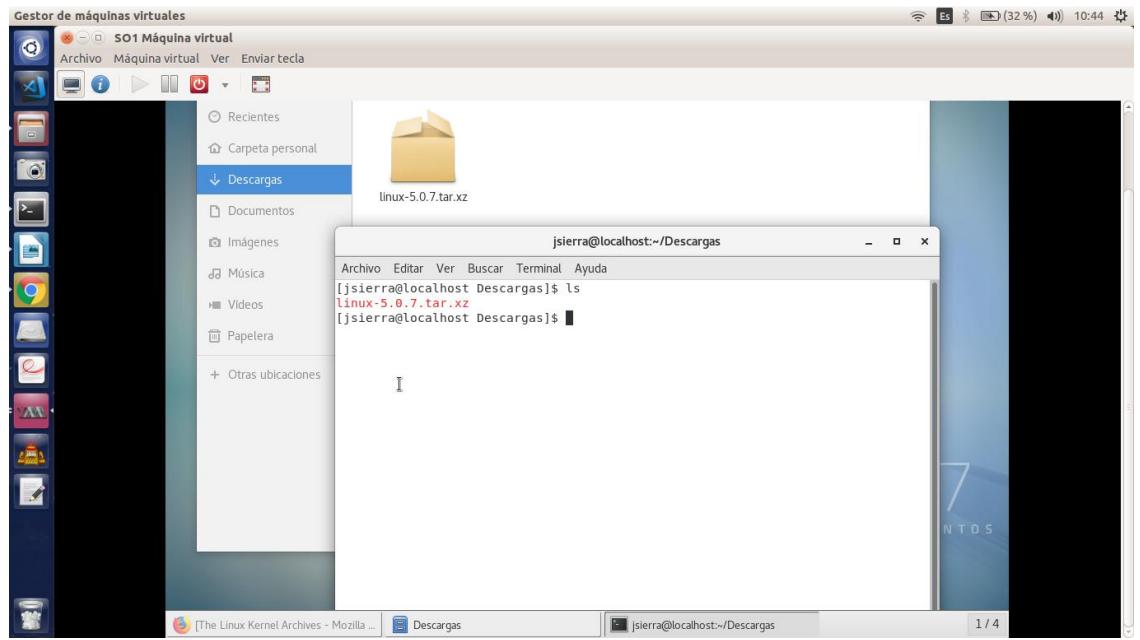
De la misma manera nos pedirá la confirmación, luego nos mostrará que dependencias hemos instalado.

Paso 2

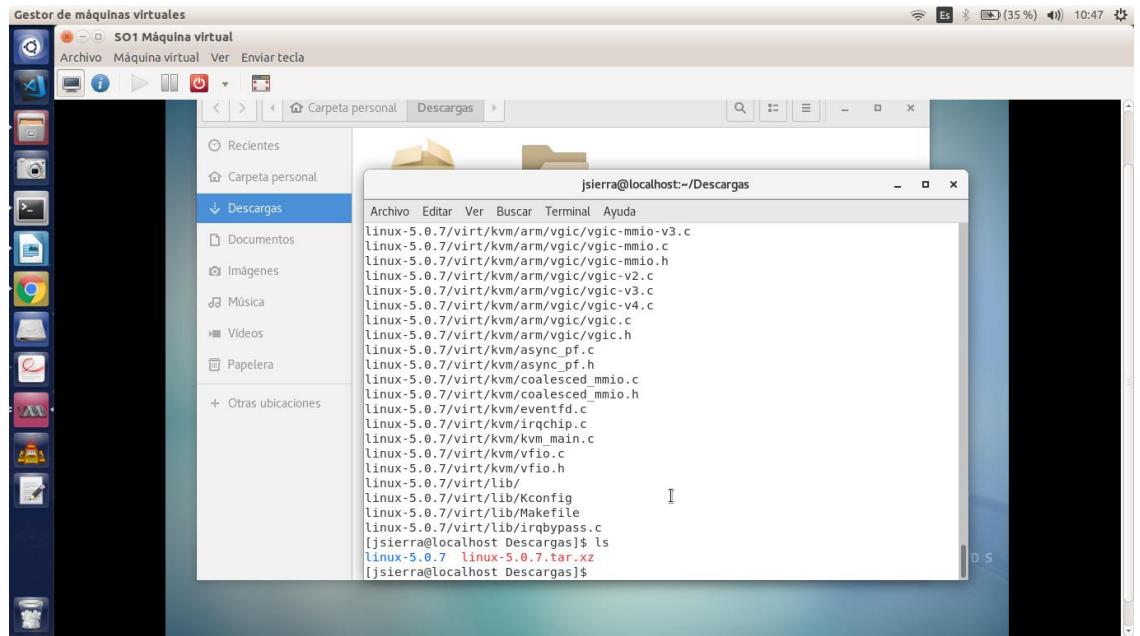
Descargaremos la versión mas estable del kernel de Linux en su pagina oficial



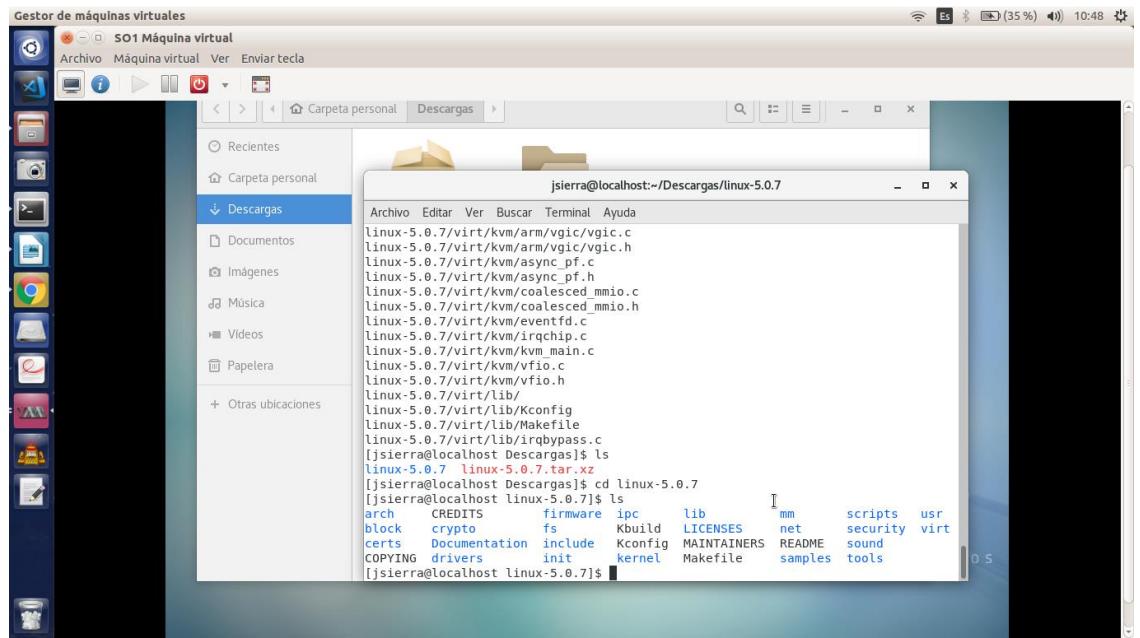
El Kernel mas estable es el 5.0.7 en este caso.



Ahora lo descomprimiremos con el siguiente comando **tar xvf linux-5.0.7.tar.xz** el nombre del archive depende de nuestro tar.



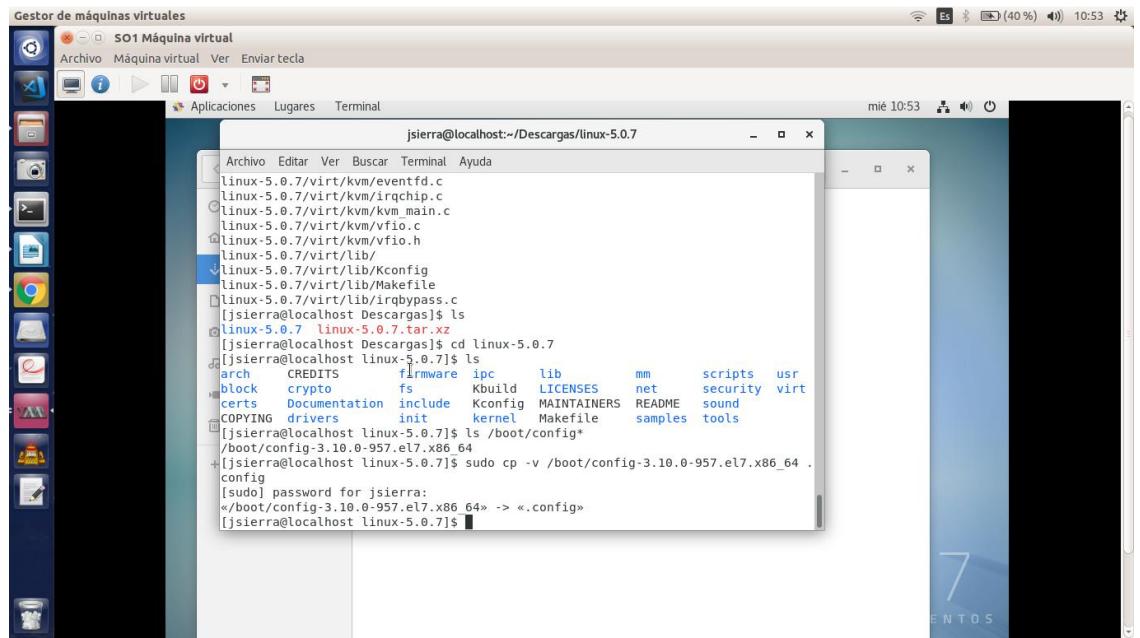
Después de descomprimir se podrá mostrar que aparece una carpeta con el nombre del tar.



Acá se muestran los archivos que tienen el kernel.

Paso 3

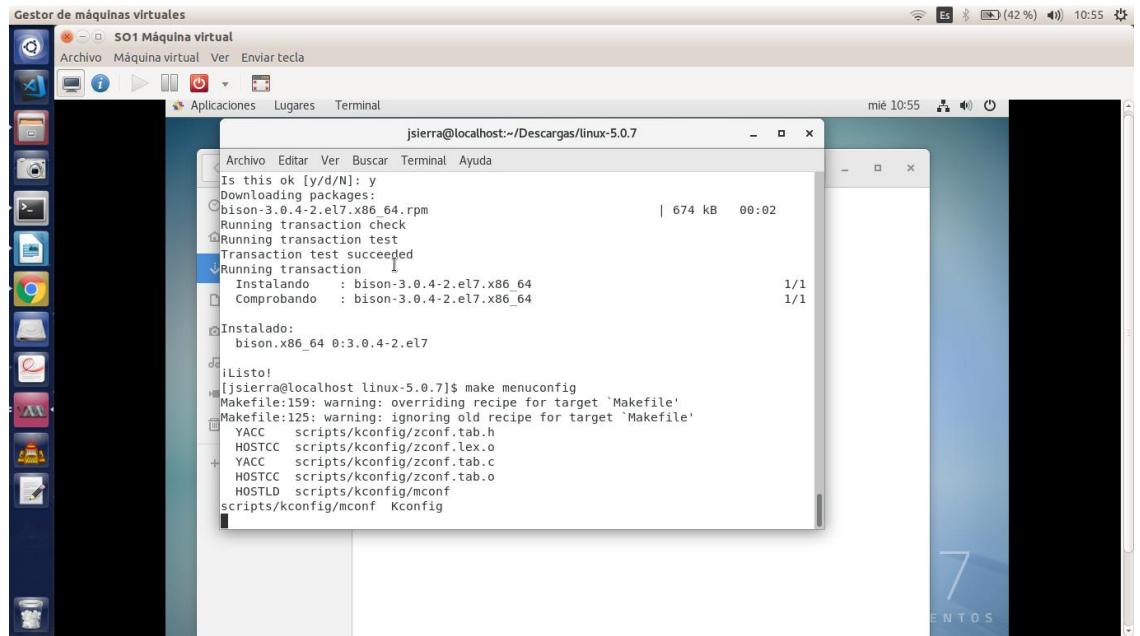
Copiar el archivo de configuraciones del kernel actual en el kernel que instalaremos.



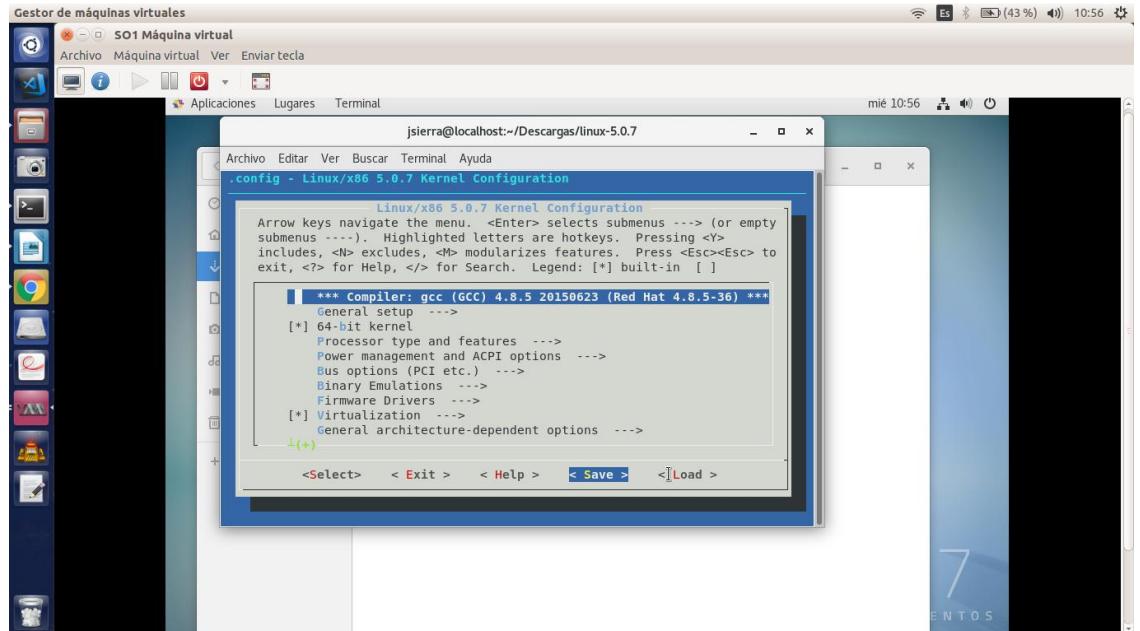
Para ello se utiliza el siguiente comando.

```
sudo cp -v /boot/config-3.10.0-957.el7.x86_64 .config
```

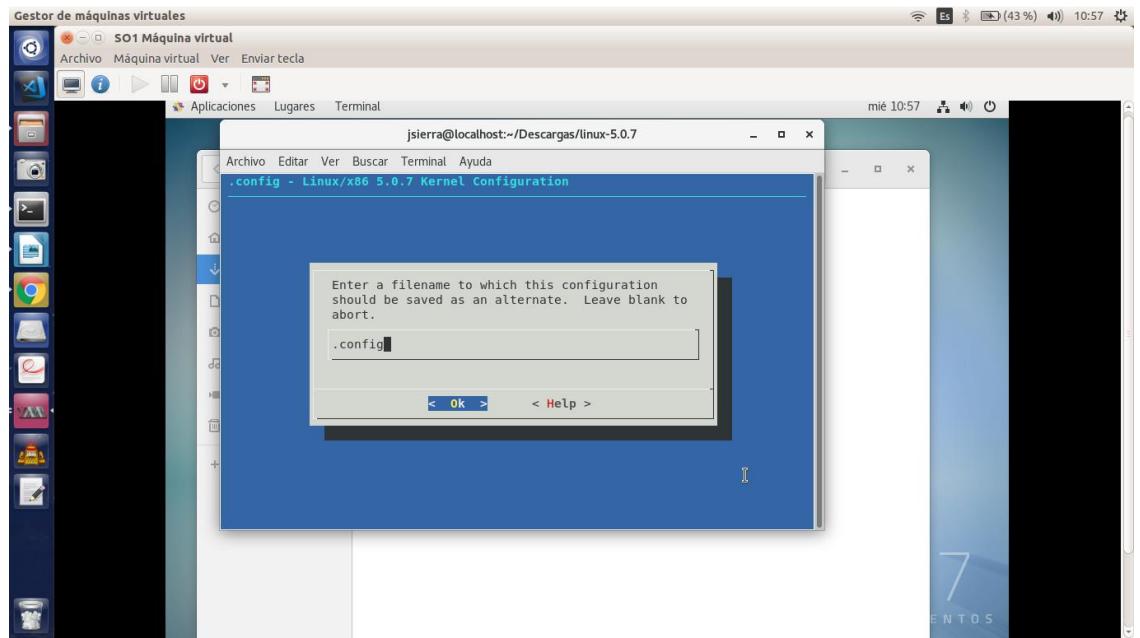
Paso 3



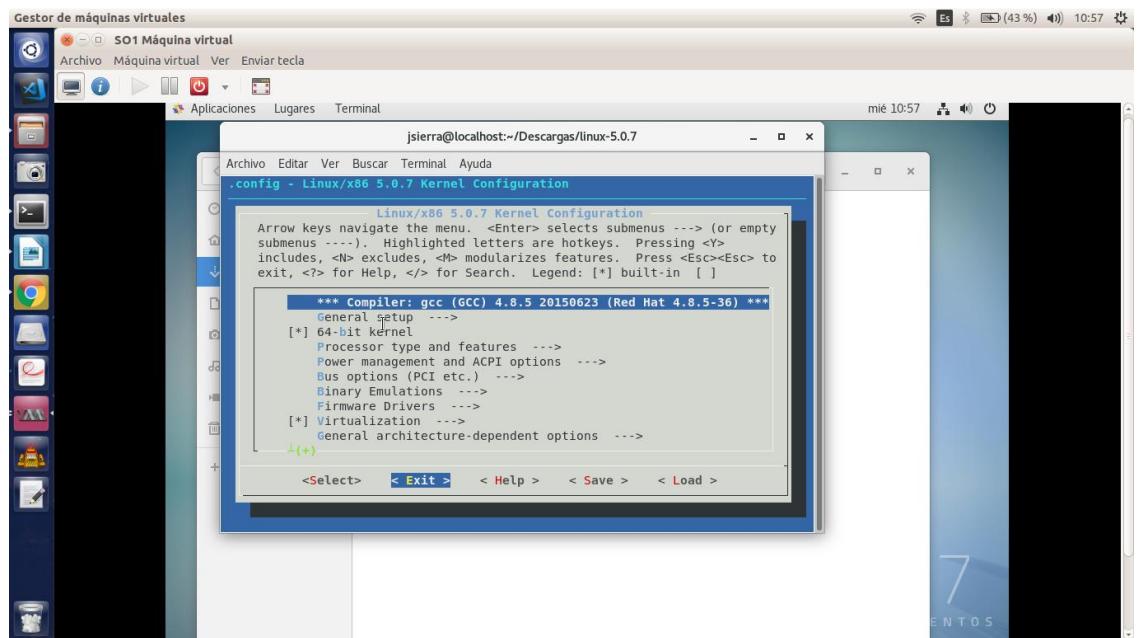
Ahora utilizaremos el siguiente comando **make menuconfig** para mostrar las configuraciones del sistema.



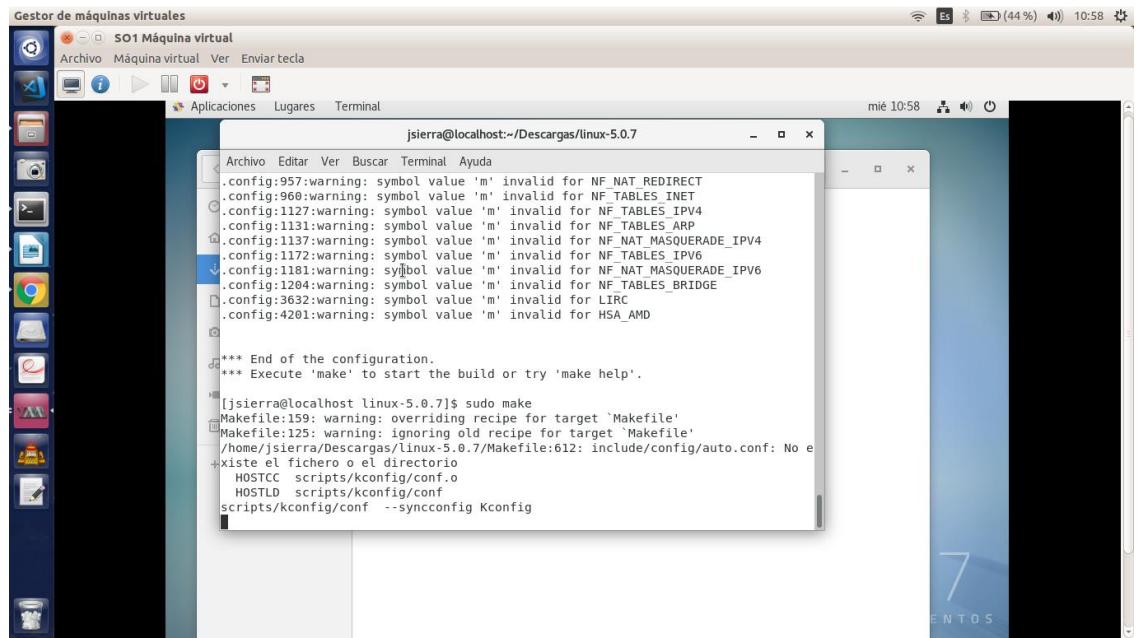
Este es el menú de configuraciones en donde se puede cambiar lo que se desee, en caso de que se tenga que cambiar algo del kernel, finalizadas las configuraciones se le dará save.



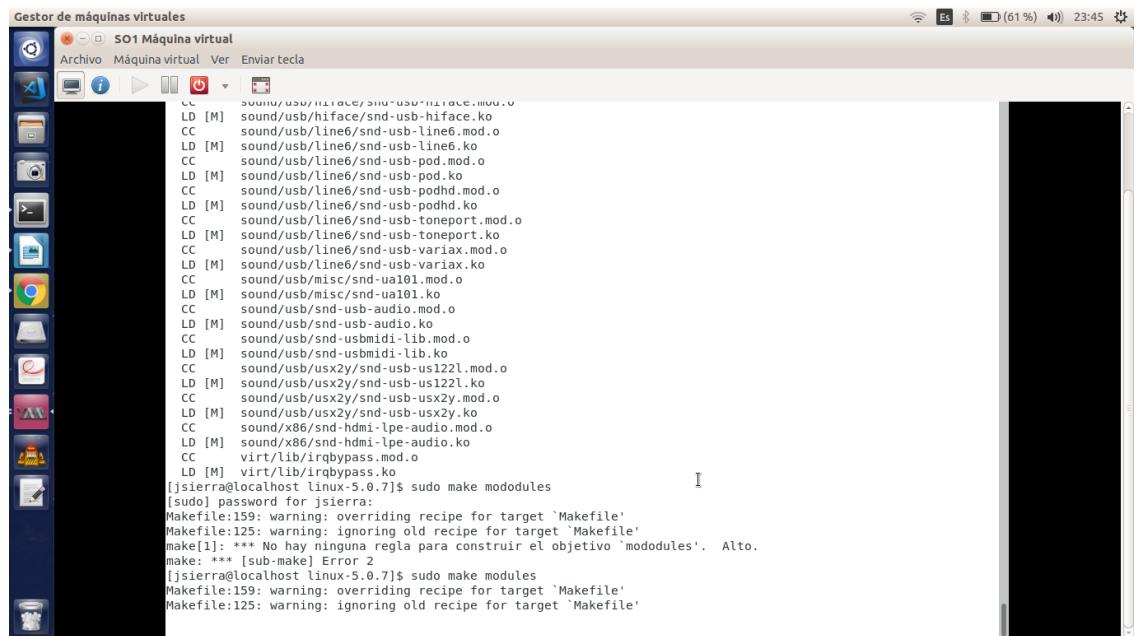
Por lo que ahora nos preguntara si queremos escribir las configuraciones en el archivo ,conf a lo que le daremos ok.



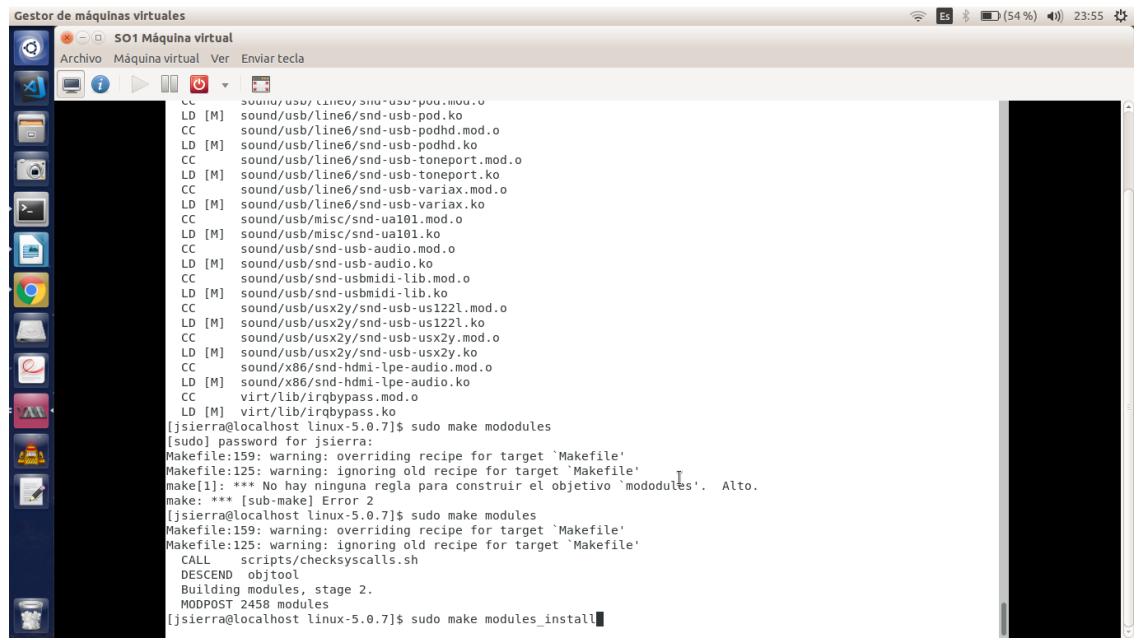
Luego saldremos del menú de configuraciones.



Después de esto colocaremos el comando **sudo make** el cual puede tardar varios minutos.

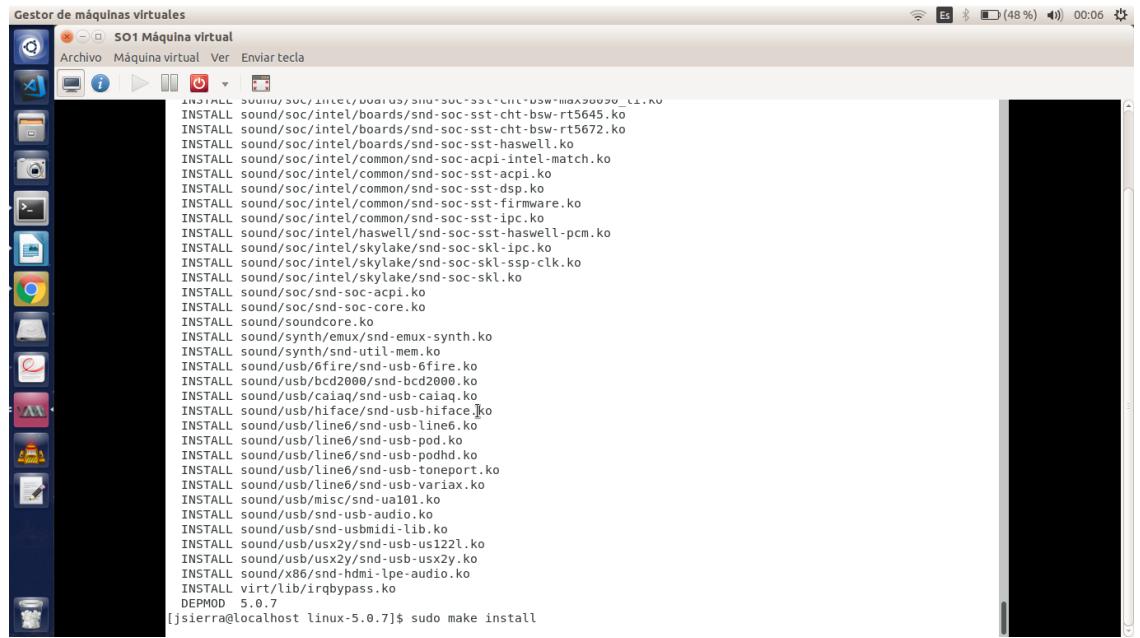


Luego el comando **sudo make modules** lo que también podría tardar bastante tiempo.



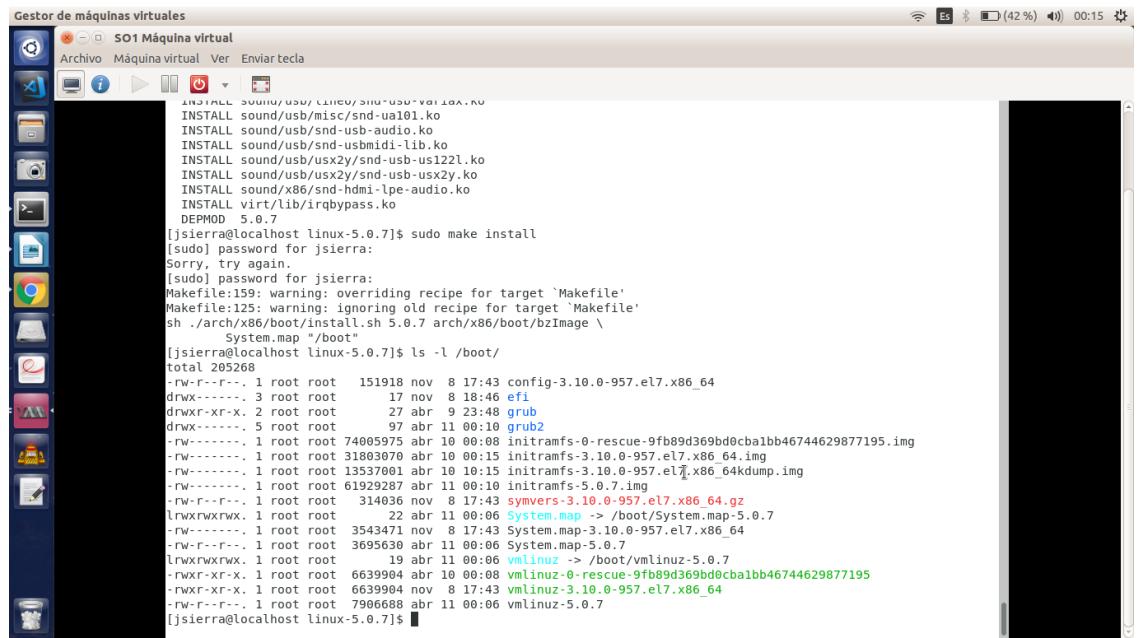
```
Archivo Máquina virtual Ver Enviar tecla
[jsierra@localhost linux-5.0.7]$ sudo make modules
[sudo] password for jsierra:
Makefile:159: warning: overriding recipe for target 'Makefile'
Makefile:125: warning: ignoring old recipe for target 'Makefile'
make[1]: *** No hay ninguna regla para construir el objetivo 'mododules'. Alto.
make: *** [sub-make] Error 2
[jsierra@localhost linux-5.0.7]$ sudo make modules
Makefile:159: warning: overriding recipe for target 'Makefile'
Makefile:125: warning: ignoring old recipe for target 'Makefile'
CALL scripts/checksyscalls.sh
DESCEND objtool
Building modules, stage 2.
MODPOST 2458 modules
[jsierra@localhost linux-5.0.7]$ sudo make modules_install
```

Luego el comando **sudo make modules_install** lo que también podría tardar bastante tiempo.



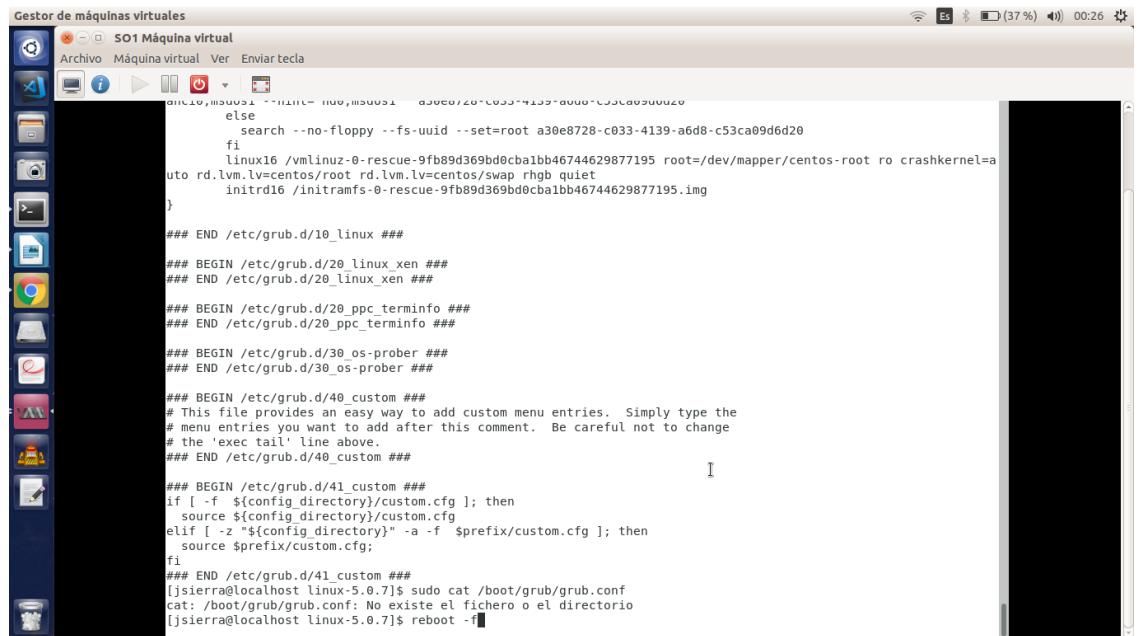
```
Archivo Máquina virtual Ver Enviar tecla
[jsierra@localhost linux-5.0.7]$ sudo make install
[...]
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-rt5645.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-rt5672.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-haswell.ko
INSTALL sound/soc/intel/common/snd-soc-acpi-intel-match.ko
INSTALL sound/soc/intel/common/snd-soc-sst-acpi.ko
INSTALL sound/soc/intel/common/snd-soc-sst-dsp.ko
INSTALL sound/soc/intel/common/snd-soc-sst-firmware.ko
INSTALL sound/soc/intel/common/snd-soc-sst-ipc.ko
INSTALL sound/soc/intel/haswell/snd-soc-sst-haswell-pcm.ko
INSTALL sound/soc/intel/skylake/snd-soc-skl-ipc.ko
INSTALL sound/soc/intel/skylake/snd-soc-skl-spp-clk.ko
INSTALL sound/soc/intel/skylake/snd-soc-skl.ko
INSTALL sound/soc/snd-soc-acpi.ko
INSTALL sound/soc/snd-soc-core.ko
INSTALL sound/soundcore.ko
INSTALL sound/synth/emu/snd-emu-synth.ko
INSTALL sound/synth/snd-util-mem.ko
INSTALL sound/usb/6fire/snd-usb-6fire.ko
INSTALL sound/usb/bcd2000/snd-bcd2000.ko
INSTALL sound/usb/caiaq/snd-usb-caiaq.ko
INSTALL sound/usb/hiface/snd-usb-hiface.ko
INSTALL sound/usb/line6/snd-usb-line6.ko
INSTALL sound/usb/line6/snd-usb-pod.ko
INSTALL sound/usb/line6/snd-usb-podhd.ko
INSTALL sound/usb/line6/snd-usb-toneport.ko
INSTALL sound/usb/line6/snd-usb-variax.ko
INSTALL sound/usb/misc/snd-snd-u101.ko
INSTALL sound/usb/usb2y/snd-usb-us122l.ko
INSTALL sound/usb/usb2y/snd-usb-usx2y.ko
INSTALL sound/x86/snd-hdmi-lpe-audio.ko
INSTALL virt/lib/irqbypass.ko
DEPMOD 5.0.7
[jsierra@localhost linux-5.0.7]$ sudo make install
```

Luego el comando **sudo make install** lo que también podría tardar bastante tiempo.



```
Gestor de máquinas virtuales
  ○ □ SO1 Máquina virtual
  Archivo Máquina virtual Ver Enviar tecla
  [jsierra@localhost linux-5.0.7]$ sudo make install
  [sudo] password for jsierra:
  Sorry, try again.
  [sudo] password for jsierra:
  Makefile:159: warning: overriding recipe for target `Makefile'
  Makefile:125: warning: ignoring old recipe for target `Makefile'
  sh ./arch/x86/boot/install.sh 5.0.7 arch/x86/boot/bzImage \
      System.map "/boot"
  [jsierra@localhost linux-5.0.7]$ ls -l /boot/
  total 205268
  -rw-r--r-- 1 root root 151918 nov  8 17:43 config-3.10.0-957.el7.x86_64
  drwx----- 3 root root    17 nov  8 18:46 efi
  drwxr-xr-x  2 root root   27 abr  9 23:48 grub
  drwx----- 5 root root   97 abr 11 00:10 grub2
  -rw----- 1 root root 74005975 abr 10 00:08 initramfs-0-rescue-9fb89d369bd0cba1bb46744629877195.img
  -rw----- 1 root root 31803076 abr 10 00:15 initramfs-3.10.0-957.el7.x86_64.img
  -rw----- 1 root root 13537001 abr 10 10:15 initramfs-3.10.0-957.el7.x86_64.kdump.img
  -rw----- 1 root root 61929287 abr 11 00:10 initramfs-5.0.7.img
  -rw-r--r-- 1 root root 31403 nov  8 17:43 symvers-3.10.0-957.el7.x86_64.gz
  lrwxrwxrwx 1 root root   22 abr 11 00:06 System.map -> /boot/System.map-5.0.7
  -rw-r--r-- 1 root root 3543471 nov  8 17:43 System.map-3.10.0-957.el7.x86_64
  -rw-r--r-- 1 root root 3695630 abr 11 00:06 System.map-5.0.7
  lrwxrwxrwx 1 root root   19 abr 11 00:06 vmlinuz -> /boot/vmlinuz-5.0.7
  -rwxr-xr-x 1 root root 6639904 abr 10 00:08 vmlinuz-0-rescue-9fb89d369bd0cba1bb46744629877195
  -rwxr-xr-x 1 root root 6639904 nov  8 17:43 vmlinuz-3.10.0-957.el7.x86_64
  -rw-r--r-- 1 root root 7906688 abr 11 00:06 vmlinuz-5.0.7
  [jsierra@localhost linux-5.0.7]$
```

Al mostrar el directorio boot debería de aparecer ahora los archivos correspondientes a nuestra versión de kernel que en este caso seria la 5.0.7



```
Gestor de máquinas virtuales
  ○ □ SO1 Máquina virtual
  Archivo Máquina virtual Ver Enviar tecla
  [jsierra@localhost ~]$ cat /boot/grub/grub.conf
  #!/bin/sh
  # This file provides an easy way to add custom menu entries. Simply type the
  # menu entries you want to add after this comment. Be careful not to change
  # the 'exec tail' line above.
  ## END /etc/grub.d/10_linux ##

  ## BEGIN /etc/grub.d/20_linux_xen ##
  ## END /etc/grub.d/20_linux_xen ##

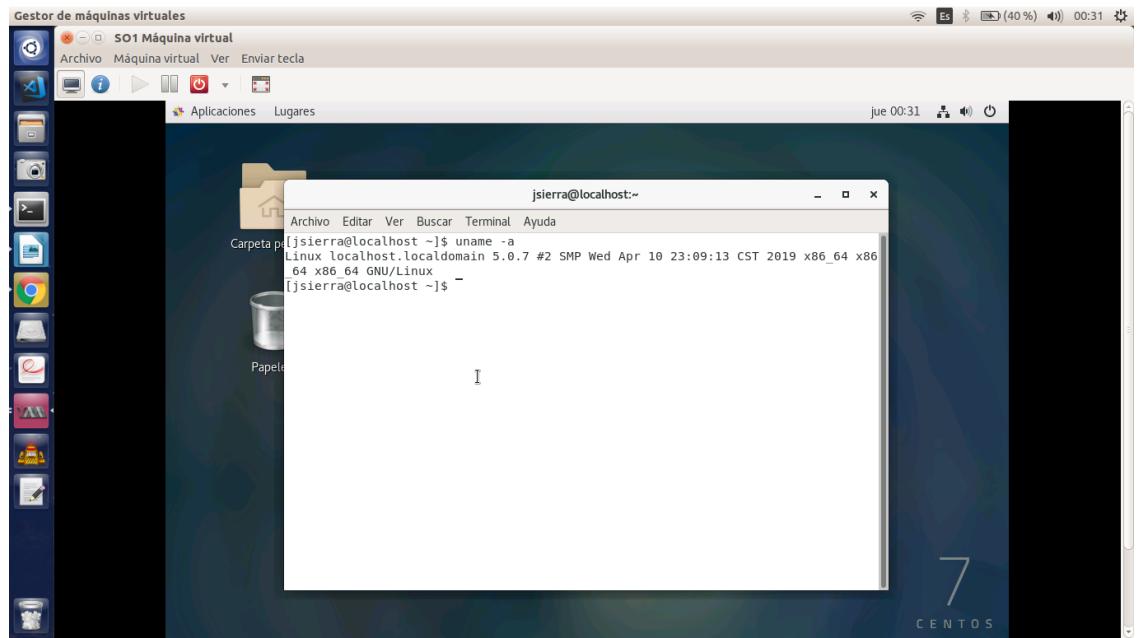
  ## BEGIN /etc/grub.d/20_ppc_terminfo ##
  ## END /etc/grub.d/20_ppc_terminfo ##

  ## BEGIN /etc/grub.d/30_os-prober ##
  ## END /etc/grub.d/30_os-prober ##

  ## BEGIN /etc/grub.d/40_custom ##
  # This file provides an easy way to add custom menu entries. Simply type the
  # menu entries you want to add after this comment. Be careful not to change
  # the 'exec tail' line above.
  ## END /etc/grub.d/40_custom ##

  ## BEGIN /etc/grub.d/41_custom ##
  if [ -f ${config_directory}/custom.cfg ]; then
    source ${config_directory}/custom.cfg
  elif [ -z "${config_directory}" -a -f $prefix/custom.cfg ]; then
    source $prefix/custom.cfg;
  fi
  ## END /etc/grub.d/41_custom ##
  [jsierra@localhost linux-5.0.7]$ sudo cat /boot/grub/grub.conf
  cat: /boot/grub/grub.conf: No existe el fichero o el directorio
  [jsierra@localhost linux-5.0.7]$ reboot -f
```

Ahora reiniciaremos la maquina con el siguiente comando **reboot -f**



Ahora volveremos a usar el comando del principio **uname -a** y podemos ver que la versión del kernel es diferente.