

# Sumario

<b>UT3: Instalación de sistemas operativos.....</b>	<b>2</b>
<b>Apartado 1. Instalación de SO.....</b>	<b>2</b>
Haiku:.....	2
Fedora:.....	4
<b>Apartado 2. Crear usuarios.....</b>	<b>6</b>
Haiku:.....	6
Fedora:.....	7
<b>Apartado 3. Instalación y configuración de una aplicación.....</b>	<b>8</b>
Haiku:.....	8
Federa:.....	13
<b>Reflexión.....</b>	<b>14</b>

# UT3: Instalación de sistemas operativos

En esta práctica vamos a trabajar la instalación de diferentes sistemas operativos, usando VirtualBox. Después de la instalación del sistema operativo necesitaremos extraer información del sistema utilizando comandos y herramientas gráficas.

## Apartado 1. Instalación de SO

Los sistemas operativos que vamos a instalar son:

- Haiku OS R1 BETA
- Fedora 41

Para la instalación deberemos descargar la ISO de la página oficial de los sistemas operativos e instalarlas en VirtualBox. Tras descargar la ISO debes documentar cómo podemos hacer la comprobación de una correcta descarga del archivo utilizando CHECKSUM.

**Importante:** Las versiones tienen que ser las indicadas.

La máquina virtual debe tener conexión a internet y con el sistema operativo de nuestro ordenador.

### Haiku:

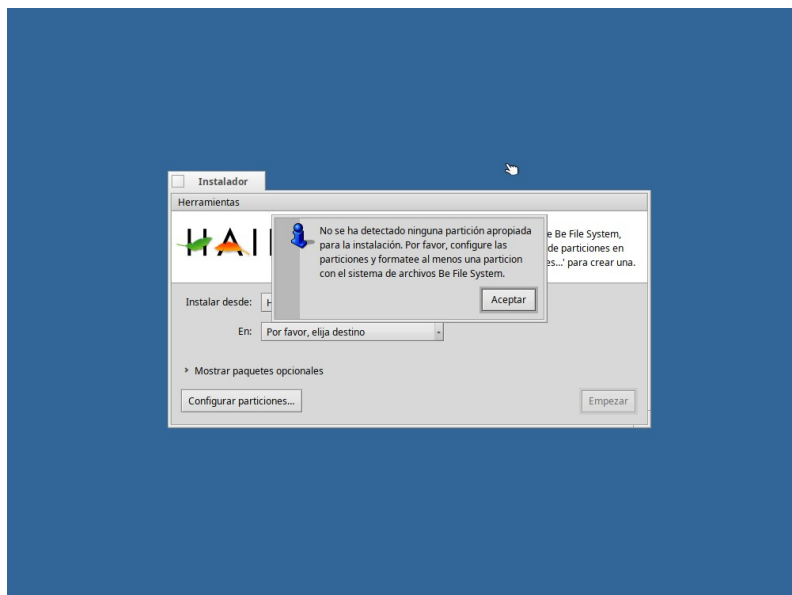
```
dam1@Carlos:~/Descargas$ sha256sum haiku-r1beta5-x86_64-anyboot.iso
22ae312a38e98083718b6984186e753d15806bd6ea44542144fdcef42c4dcb69  haiku-r1beta5-
```

#### Checksum Verification

To make sure that you have got the right file and that it has not been corrupted during file transfer, it is always wise to ensure the integrity of your download. To that end, you can use the SHA256 checksums shown below to verify that what you have is the correct file.

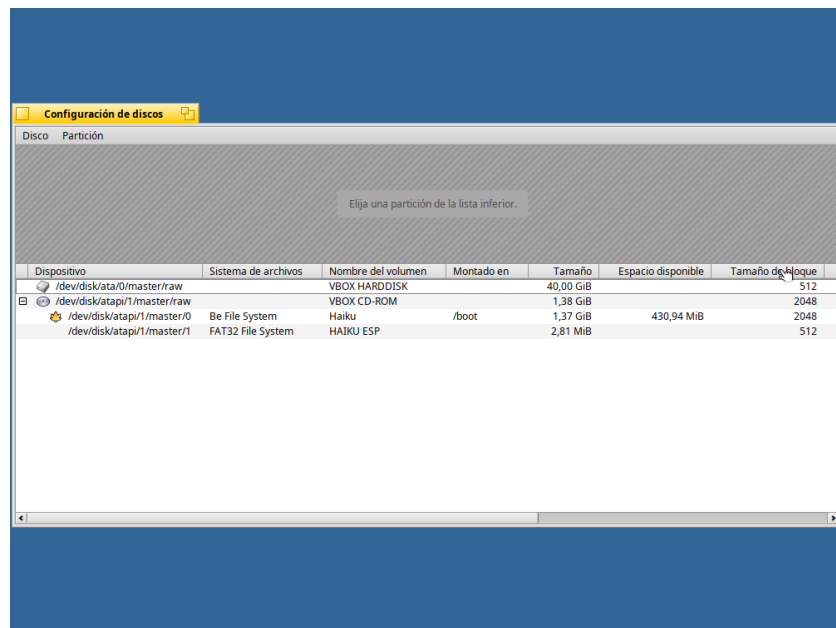
```
22ae312a38e98083718b6984186e753d15806bd6ea44542144fdcef42c4dcb69  haiku-r1beta5-x86_64-anyboot.iso
```

Realicé un checksum (**sha256sum “ISO”**) para verificar si era el archivo correcto (como se puede observar, si lo era). Luego comencé con la instalación (Utilice la misma configuración en los dos sistemas operativos: 2gb Ram, 2 Núcleos del procesador, 40gb Almacenamiento)

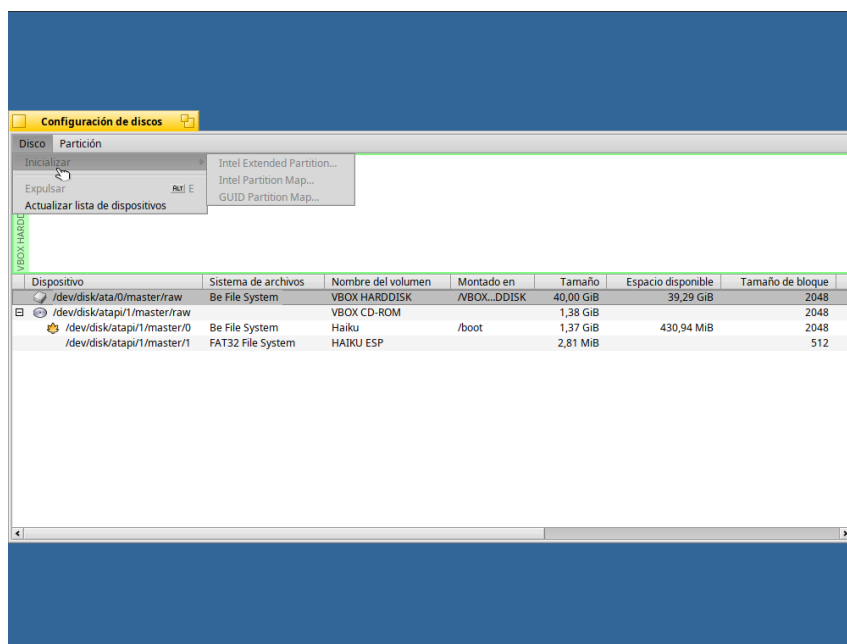


Aquí comenzaron los problemas.

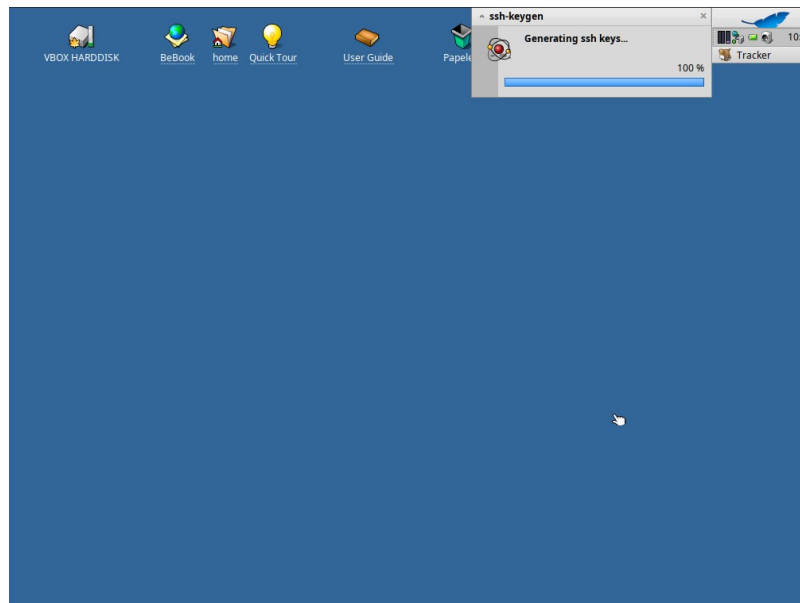
El sistema operativo Haiku utiliza un diferente sistema de archivos que no es común (Be File System o BFS), algunos discos duros no están configurados o formateados con ese sistema. Por lo que para solucionarlo y poder instalar la ISO hay que realizar dos pasos:



1. Formatear el disco e inicializar un nuevo mapa de particiones nuevo.



2. Una vez hecho, se crea una partición con el sistema de archivos BFS, y al realizarlo, se permite la instalación y se continua con el proceso.



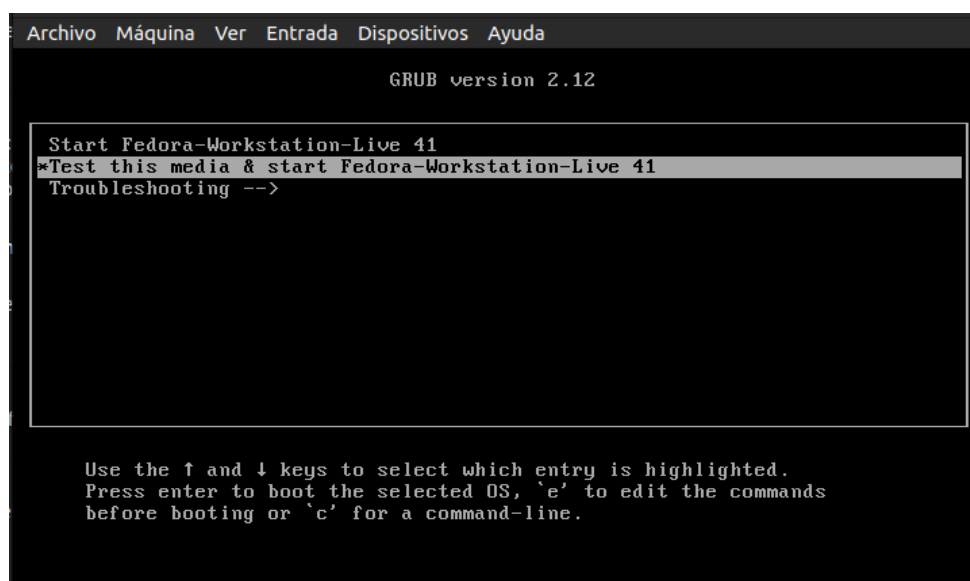
Al continuar, finalmente entramos al entorno gráfico de Haiku y nos permite seguir con la práctica.

## Fedora:

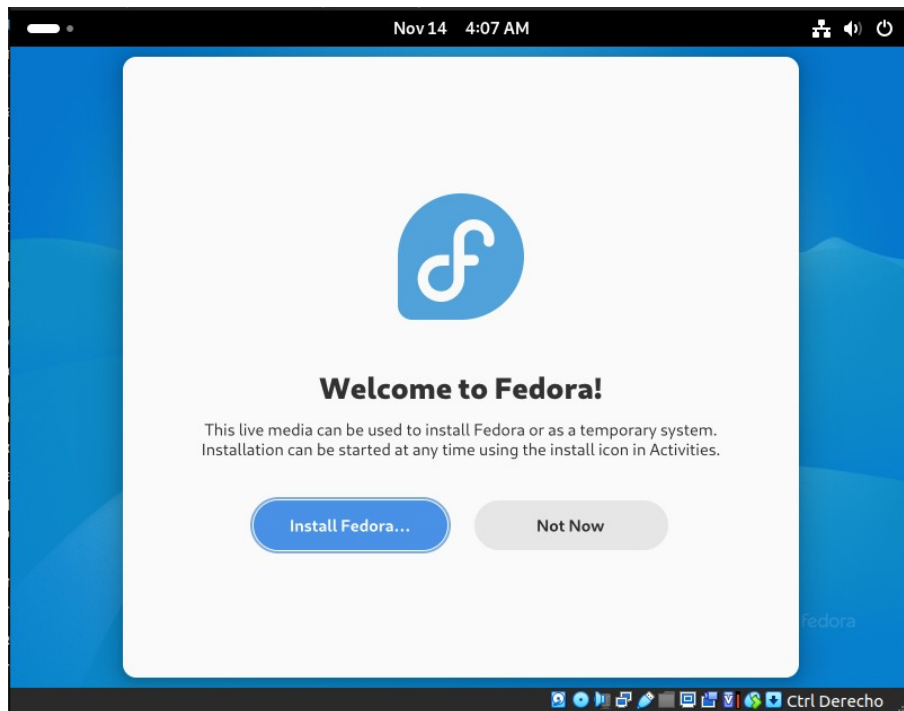
```
dam1@Carlos:~/Descargas$ sha256sum Fedora-Workstation-Live-x86_64-41-1.4.iso
a2dd3caf3224b8f3a640d9e31b1016d2a4e98a6d7cb435a1e2030235976d6da2  Fedora-Worksta
tion-Live-x86_64-41-1.4.iso
```

```
# Fedora-Workstation-Live-x86_64-41-1.4.iso: 2458187776 bytes
SHA256 (Fedora-Workstation-Live-x86_64-41-1.4.iso) = a2dd3caf3224b8f3a640d9e31b1016d2a4e98a6d7cb435a1e2030235976d6da2
```

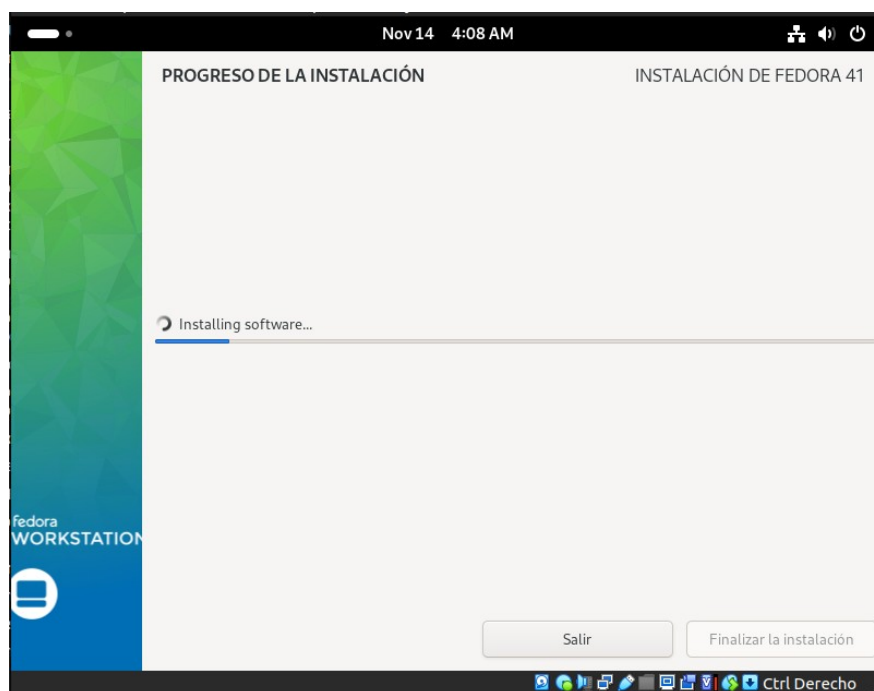
Como siempre, realizamos el checksum antes de comenzar la instalación para verificar si el archivo es fiable, de la página oficial.



Al iniciar el equipo por primera vez, se ve una interfaz oscura.



La instalación de Fedora es uno de los más fáciles, ya que tiene una interfaz muy intuitiva y visual.

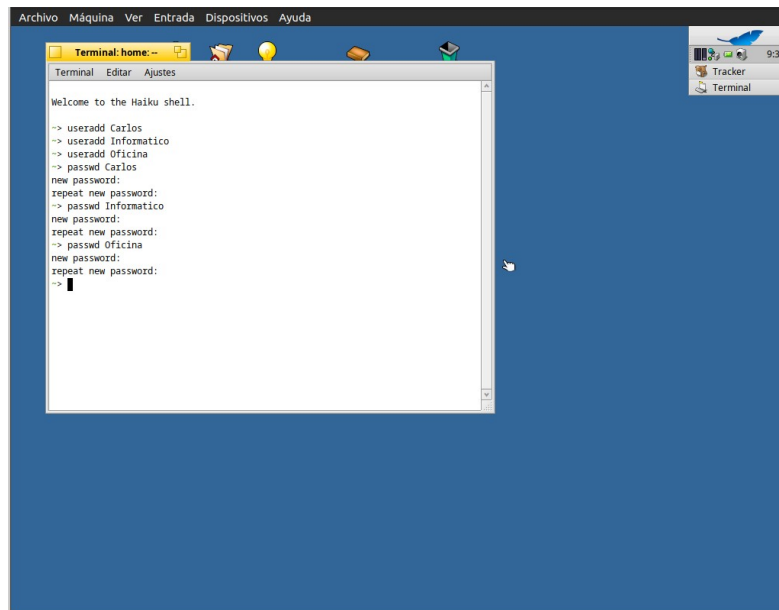


Se deben realizar una serie de configuraciones previas pero simples y luego, comienza con la instalación.

## Apartado 2. Crear usuarios

En los dos sistemas operativos vamos a crear 3 usuarios diferentes, con sus directorios propios. Los 3 usuarios serán: informatico, oficina y (“Vuestro nombre”). Para crear estos usuarios utilizaremos comandos por terminal.

### Haiku:

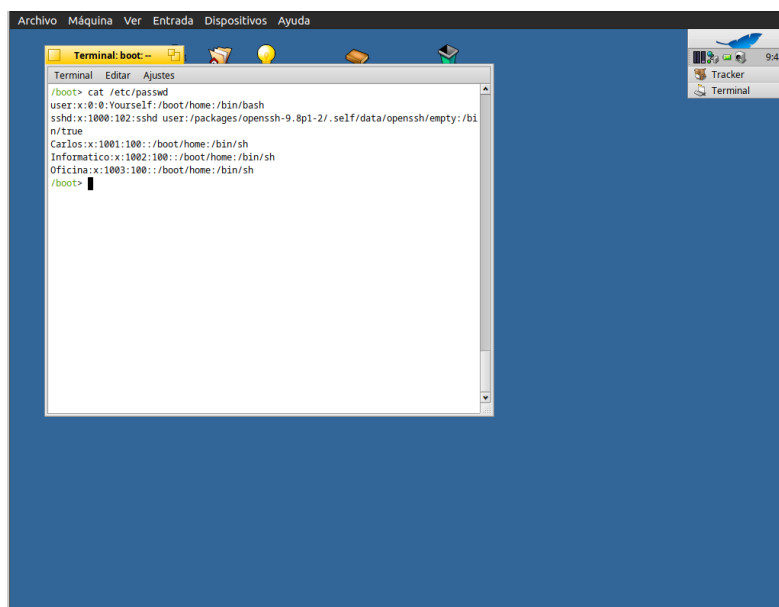


Utilizando comandos derivados de Unix, se pueden crear una serie de usuarios que a primera vista lo parecen, pero tiene un poco de trampa:

### useradd “Nombre usuario”

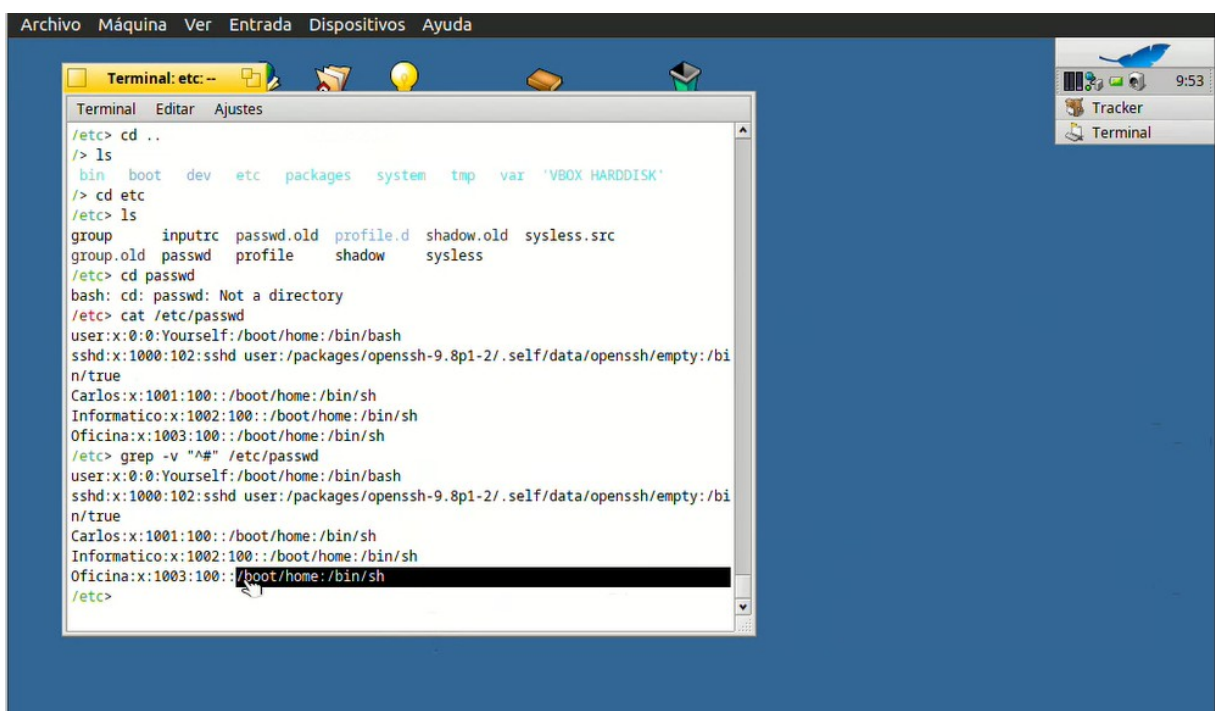
Sin embargo, se le pueden asignar contraseña aunque no sean por así decirlo cuentas independientes:

### passwd “Nombre usuario”



## Explicación (Porque se puede utilizar comandos de creación de usuarios si el SO es monousuario):

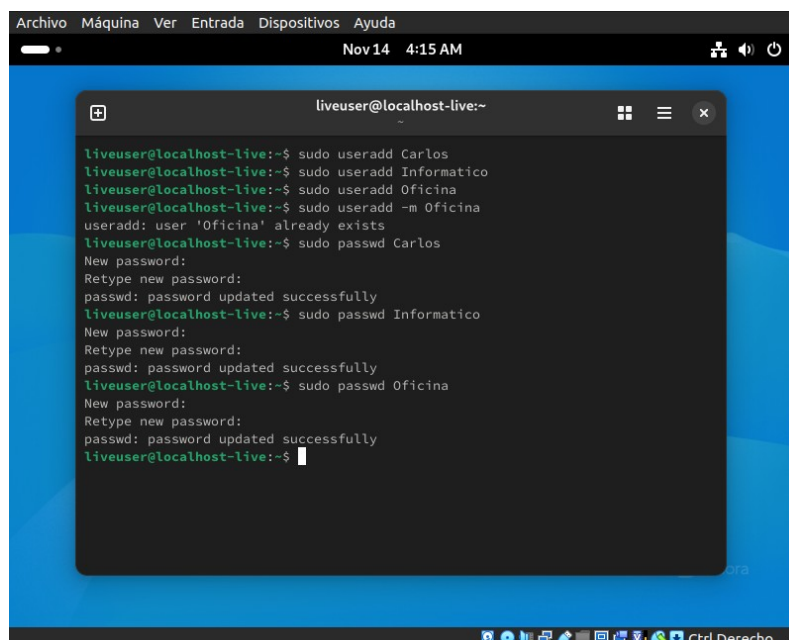
Haiku es un sistema operativo monousuario, por lo que no permite crear ni utilizar varios usuarios reales, sino únicamente trabajar con uno solo. Los usuarios creados mediante el comando **"useradd"** son simbólicos, pues no generan sesiones independientes ni modifican el comportamiento del sistema. Aunque Haiku incluye utilidades clásicas de Unix para resultar familiar, no implementa seguridad multiusuario: todos los procesos se ejecutan con los mismos privilegios, existe un único entorno de escritorio y un único usuario real. El comando **"useradd"** únicamente añade entradas en **/etc/passwd** y **/etc/group**, pero no crea cuentas activas, solo registra texto plano sin afectar a permisos ni ejecución. Esto puede comprobarse usando **cat /etc/passwd**. Haiku incorpora comandos de creación de usuarios solo por usar un sistema de comandos POSIX-like (BSD/Linux), pero, al ser un SO monousuario, la creación de usuarios reales es imposible; solo puede simularse mediante carpetas en **/home** o usando **"useradd"** sin efecto funcional.



The screenshot shows a Haiku desktop environment with a blue background. A terminal window titled "Terminal: etc--" is open, displaying the following commands and output:

```
/etc> cd ..
/> ls
bin boot dev etc packages system tmp var 'VBOX HARDISK'
/> cd etc
/etc> ls
group inputrc passwd.old profile.d shadow.old sysless.src
group.old passwd profile shadow sysless
/etc> cd passwd
bash: cd: passwd: Not a directory
/etc> cat /etc/passwd
user:x:0:0:Yourself:/boot/home:/bin/bash
sshd:x:1000:102:sshd user:/packages/openssh-9.8p1-2/.self/data/openssh/empty:/bin/true
Carlos:x:1001:100:/boot/home:/bin/sh
Informatico:x:1002:100:/boot/home:/bin/sh
Oficina:x:1003:100:/boot/home:/bin/sh
/etc> grep -v "^#" /etc/passwd
user:x:0:0:Yourself:/boot/home:/bin/bash
sshd:x:1000:102:sshd user:/packages/openssh-9.8p1-2/.self/data/openssh/empty:/bin/true
Carlos:x:1001:100:/boot/home:/bin/sh
Informatico:x:1002:100:/boot/home:/bin/sh
Oficina:x:1003:100:/boot/home:/bin/sh
/etc>
```

## Fedora:



The screenshot shows a Fedora live environment desktop with a blue background. A terminal window titled "liveuser@localhost-live:~" is open, displaying the following commands and output:

```
liveuser@localhost-live:~$ sudo useradd Carlos
liveuser@localhost-live:~$ sudo useradd Informatico
liveuser@localhost-live:~$ sudo useradd Oficina
liveuser@localhost-live:~$ sudo useradd -m Oficina
useradd: user 'Oficina' already exists
liveuser@localhost-live:~$ sudo passwd Carlos
New password:
Retype new password:
passwd: password updated successfully
liveuser@localhost-live:~$ sudo passwd Informatico
New password:
Retype new password:
passwd: password updated successfully
liveuser@localhost-live:~$ sudo passwd Oficina
New password:
Retype new password:
passwd: password updated successfully
liveuser@localhost-live:~$
```

Mediante una serie de comandos, es posible crear usuarios independientes y asignarles contraseña: (el comando sudo permite tener privilegios al utilizar un comando)

**sudo useradd “Nombre usuario”**

**sudo passwd “Nombre usuario”**

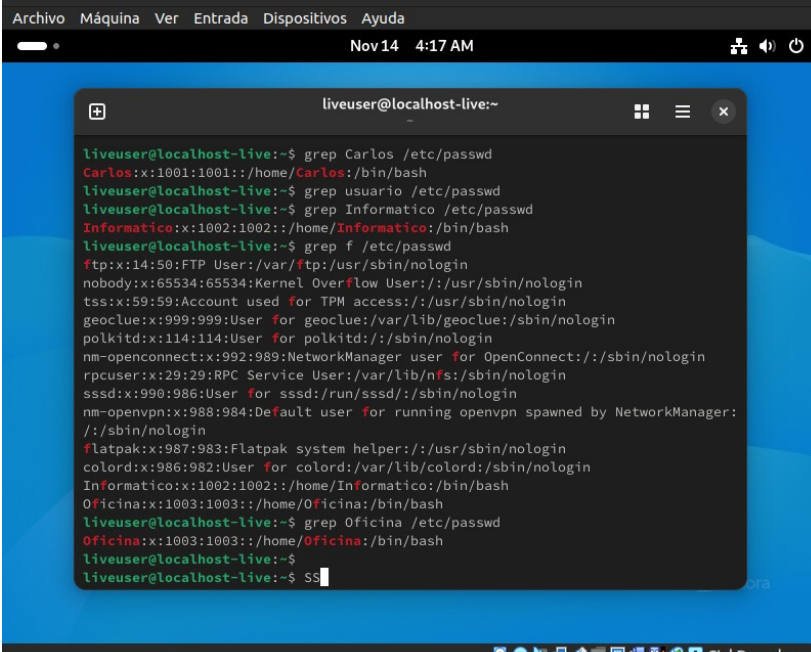
Para comprobar que si existen estos usuarios se pueden utilizar diversos comandos:

**cat /etc/passwd** (ver todos los usuarios)

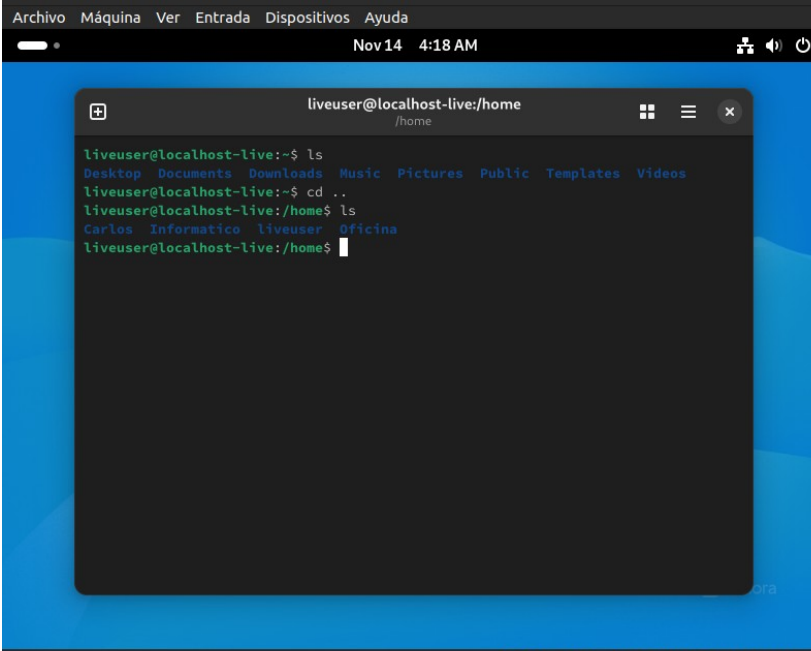
**grep usuario /etc/passwd** (filtrar un solo usuario)

O simplemente buscar en home si existe una carpeta con el nombre de usuario.

```
Carlos:x:1001:1001::/home/Carlos:/bin/bash
Informatico:x:1002:1002::/home/Informatico:/bin/bash
Oficina:x:1003:1003::/home/Oficina:/bin/bash
```



A terminal window titled 'liveuser@localhost-live:~' is shown. It displays the output of several 'grep' commands. The first command is 'grep Carlos /etc/passwd', which returns 'Carlos:x:1001:1001::/home/Carlos:/bin/bash'. The second is 'grep usuario /etc/passwd', which returns 'liveuser@localhost-live:~\$ grep usuario /etc/passwd'. The third is 'grep Informatico /etc/passwd', which returns 'Informatico:x:1002:1002::/home/Informatico:/bin/bash'. The fourth is 'grep f /etc/passwd', which returns a list of system users including ftp, nobody, tss, geoclue, polkitd, nm-openconnect, rpcuser, sssd, nm-openvpn, flatpak, colord, Informatico, and Oficina. The fifth is 'grep Oficina /etc/passwd', which returns 'Oficina:x:1003:1003::/home/Oficina:/bin/bash'. The terminal ends with 'liveuser@localhost-live:~\$ S\$'.



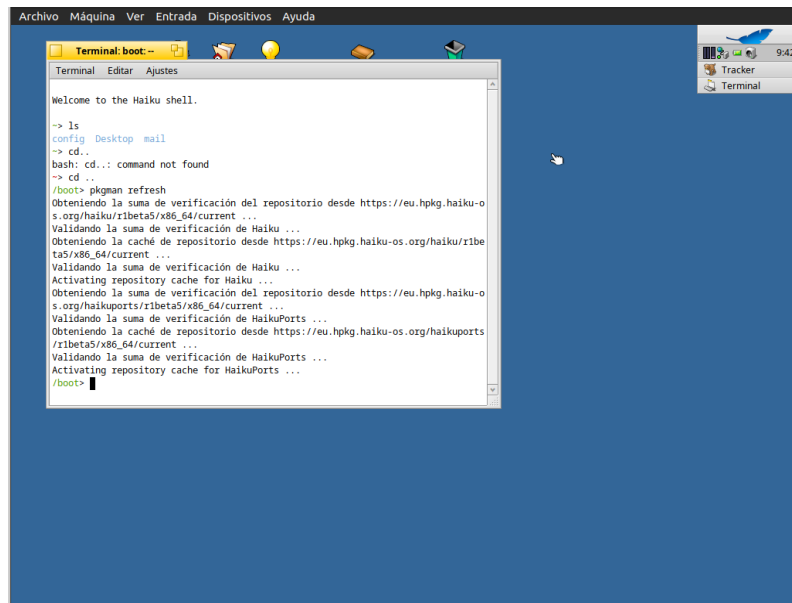
A terminal window titled 'liveuser@localhost-live:/home' is shown. It displays the output of 'ls' and 'cd' commands. The first 'ls' command shows the contents of the /home directory: 'Desktop Documents Downloads Music Pictures Public Templates Videos'. The 'cd ..' command changes the directory to '/home'. The second 'ls' command shows the contents of the /home directory: 'Carlos Informatico liveuser Oficina'. The terminal ends with 'liveuser@localhost-live:/home\$'.



## Apartado 3. Instalación y configuración de una aplicación

Instalación de cualquier paquete mediante terminal. Detallar instalación, comprobación y desinstalación de un paquete.

### Haiku:

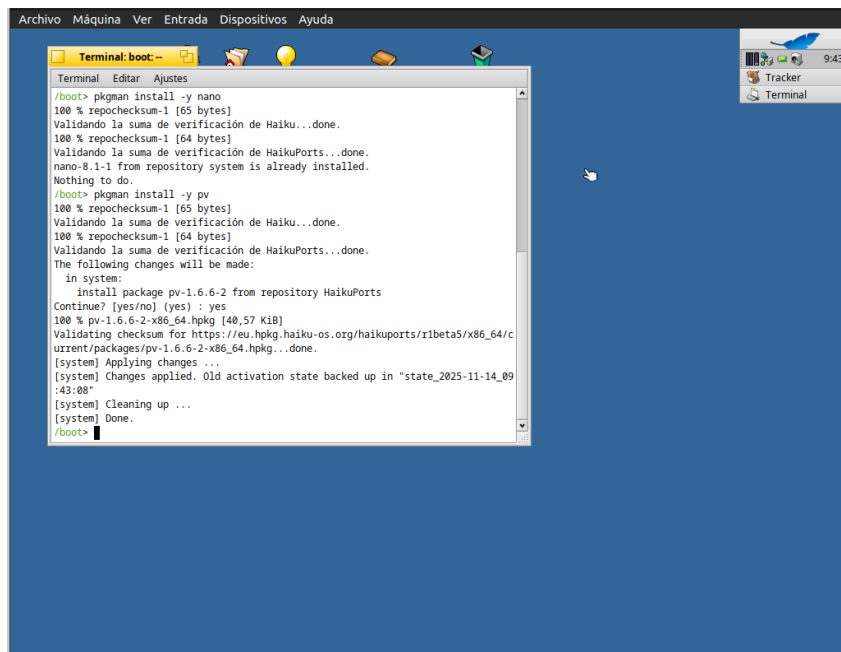


```
Terminal: boot: ~
Welcome to the Haiku shell.

-> ls
config Desktop mail
-> cd ..
bash: cd: command not found
-> cd ..
/boot> pkgman refresh
Obteniendo la suma de verificación del repositorio desde https://eu.hpkg.haiku-os.org/haiku/ribeta5/x86_64/current ...
Validando la suma de verificación de Haiku ...
Obteniendo la suma de verificación de HaikuPorts ...
Validando la suma de verificación de HaikuPorts ...
Obteniendo la suma de verificación de HaikuPorts ...
Validando la suma de verificación de HaikuPorts ...
Activating repository cache for HaikuPorts ...
/boot>
```

Primeramente, realizamos una actualización de repositorios mediante el comando:

### pkgman refresh

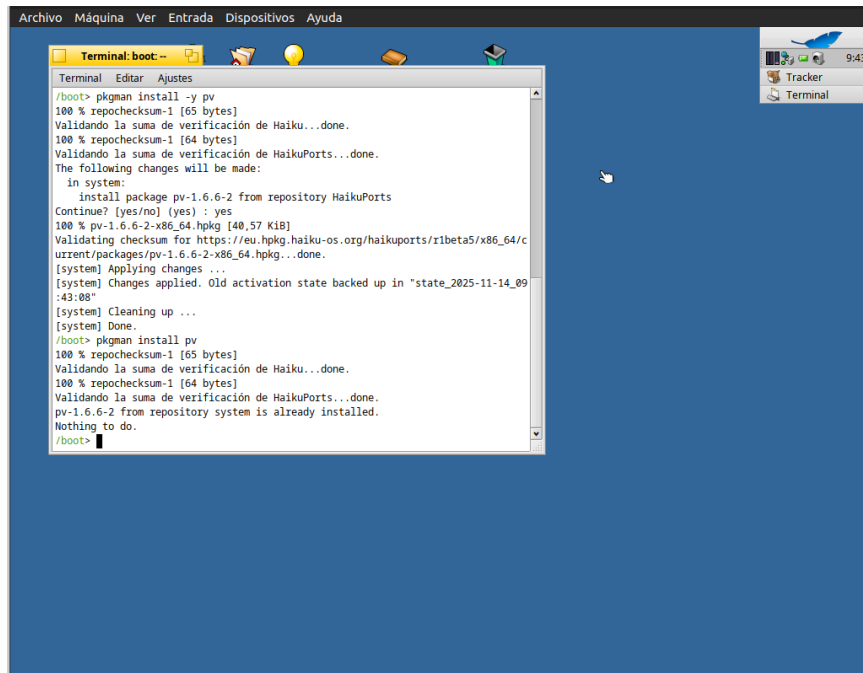


```
Terminal: boot: ~
/boot> pkgman install -y nano
100 % repochecksum-1 [65 bytes]
Validando la suma de verificación de Haiku...done.
100 % repochecksum-1 [64 bytes]
Validando la suma de verificación de HaikuPorts...done.
nano-8.1-1 from repository system is already installed.
Nothing to do.
/boot> pkgman install -y pv
100 % repochecksum-1 [65 bytes]
Validando la suma de verificación de Haiku...done.
100 % repochecksum-1 [64 bytes]
Validando la suma de verificación de HaikuPorts...done.
The following changes will be made:
in system:
  install package pv-1.6.6-2 from repository HaikuPorts
Continue? [yes/no] (yes) : yes
100 % pv-1.6.6-2-x86_64.hpkg [48,57 KiB]
Validating checksum for https://eu.hpkg.haiku-os.org/haikuports/ribeta5/x86_64/current/packages/pv-1.6.6-2-x86_64.hpkg...done.
[system] Applying changes ...
[system] Changes applied. Old activation state backed up in "state_2025-11-14_09:43:08"
[system] Cleaning up ...
[system] Done.
/boot>
```

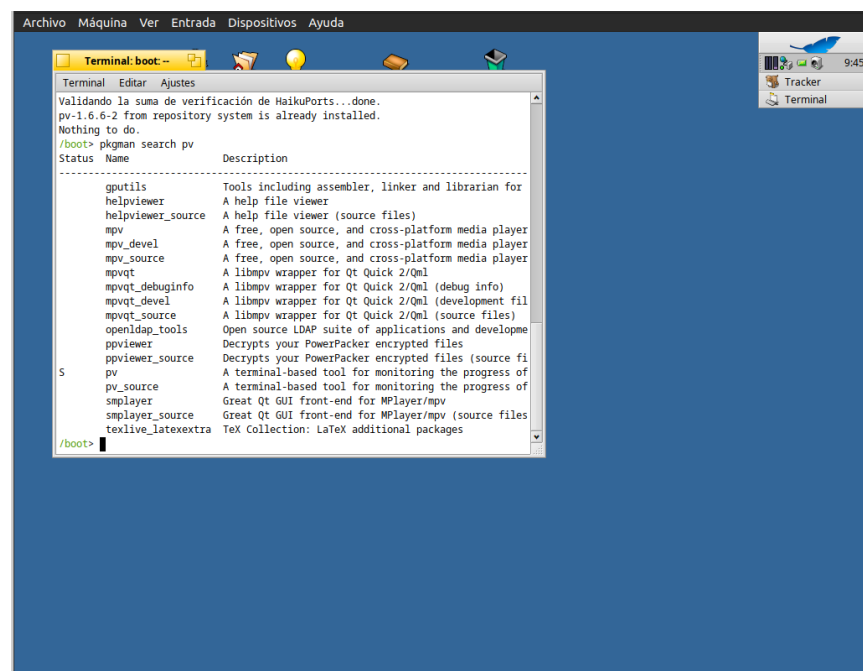
Seguidamente, comenzamos con la instalación de un archivo **pv**, para ello existen dos maneras:

**pkgman install -y “Nombre paquete”** (Instala el paquete en segundo plano, sin preguntar ni ser interactivo)

**pkgman install “Nombre paquete”**



```
Archivo Máquina Ver Entrada Dispositivos Ayuda
Terminal: boot --
Terminal Editar Ajustes
/boot> pkgman install -y pv
100 % repochecksum-1 [65 bytes]
Validando la suma de verificación de Haiku...done.
100 % repochecksum-1 [64 bytes]
Validando la suma de verificación de HaikuPorts...done.
The following changes will be made:
  in system:
    install package pv-1.6.6-2 from repository HaikuPorts
Continue? [yes/no] (yes) : yes
100 % pv-1.6.6-2-x86_64.hpkg [40.57 KiB]
Validating checksum for https://eu.hpkg.haiku-os.org/haikuports/r1beta5/x86_64/current/packages/pv-1.6.6-2-x86_64.hpkg...done.
[system] Applying changes ...
[system] Changes applied. Old activation state backed up in "state_2025-11-14_09:43:08"
[system] Cleaning up ...
[system] Done.
/boot> pkgman install pv
100 % repochecksum-1 [65 bytes]
Validando la suma de verificación de Haiku...done.
100 % repochecksum-1 [64 bytes]
Validando la suma de verificación de HaikuPorts...done.
pv-1.6.6-2 from repository system is already installed.
Nothing to do.
/boot>
```



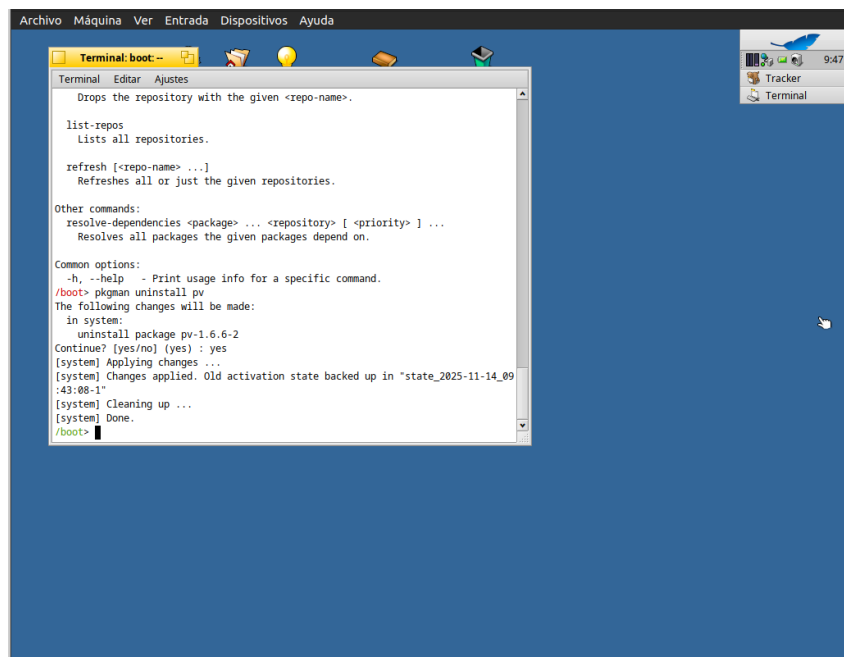
```
Archivo Máquina Ver Entrada Dispositivos Ayuda
Terminal: boot --
Terminal Editar Ajustes
Validando la suma de verificación de HaikuPorts...done.
pv-1.6.6-2 from repository system is already installed.
Nothing to do.
/boot> pkgman search pv
Status Name Description
-----
gputils Tools including assembler, linker and librarian for
helpviewer A help file viewer
helpviewer_source A help file viewer (source files)
mpv A free, open source, and cross-platform media player
mpv_devel A free, open source, and cross-platform media player
mpv_source A free, open source, and cross-platform media player
mpvqt A libmpv wrapper for Qt Quick 2/Qml
mpvqt_debuginfo A libmpv wrapper for Qt Quick 2/Qml (debug info)
mpvqt_devel A libmpv wrapper for Qt Quick 2/Qml (development fil
mpvqt_source A libmpv wrapper for Qt Quick 2/Qml (source files)
openidap_tools Open source LDAP suite of applications and developme
ppviewer Decrypts your PowerPacker encrypted files
ppviewer_source Decrypts your PowerPacker encrypted files (source fi
pv A terminal-based tool for monitoring the progress of
pv_source A terminal-based tool for monitoring the progress of
smplayer Great Qt GUI front-end for MPlayer/mpv
smplayer_source Great Qt GUI front-end for MPlayer/mpv (source files
texlive_latexextra TeX Collection: LaTeX additional packages
/boot>
```

Luego, verificamos que el paquete esta instalado:

**pkgman search “Nombre del paquete”** (También se puede utilizar con **info** → **search** para ver más información a detalle).

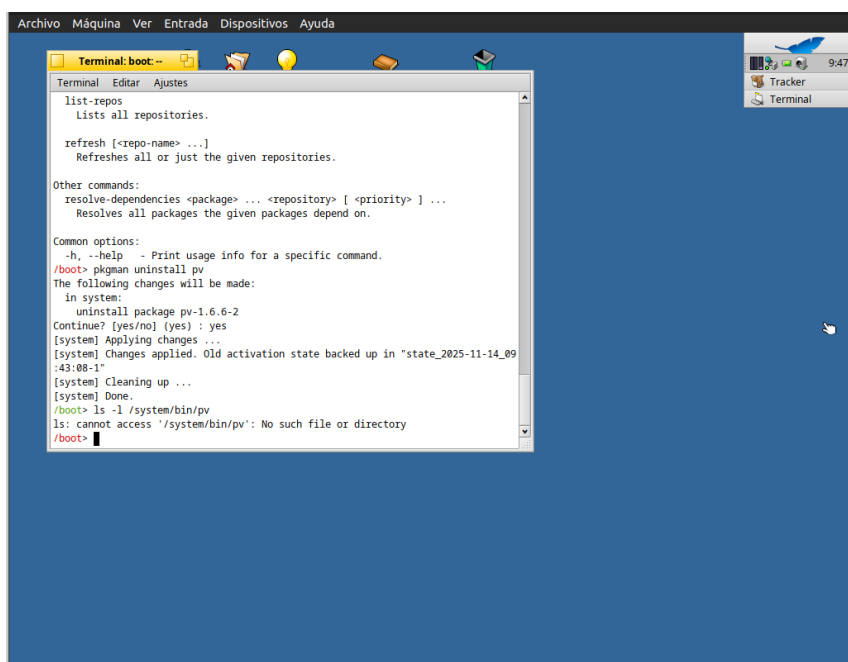
**pkgman list | grep “Nombre del paquete”** (Ver el listado de paquetes instalados)



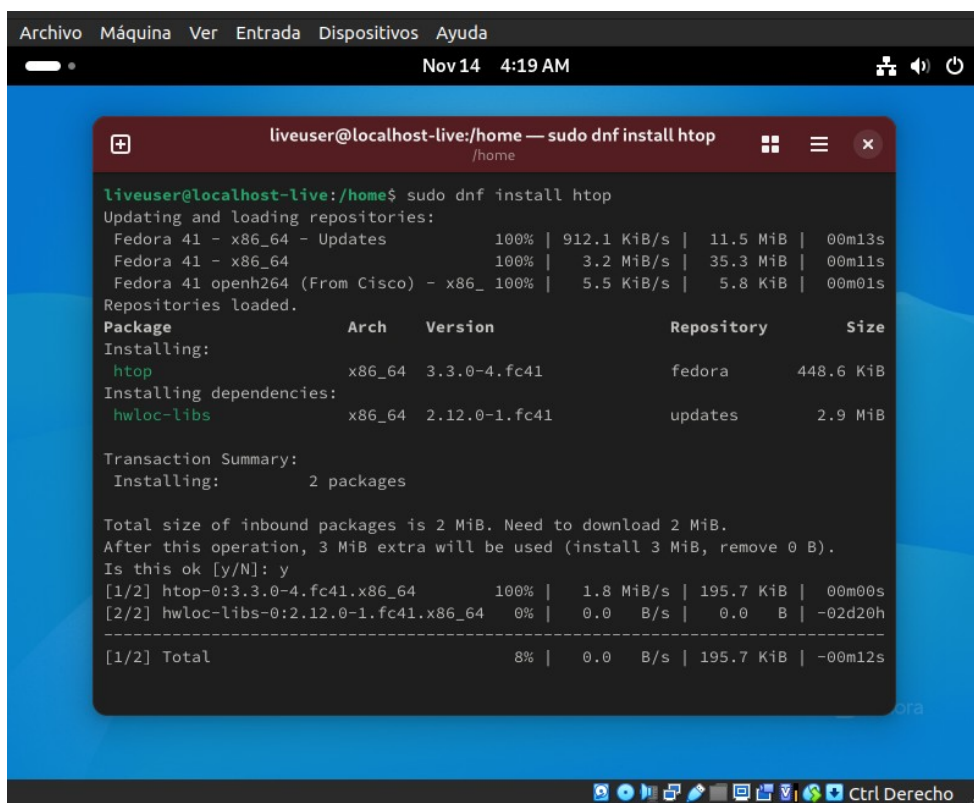


Finalmente, desinstalamos el paquete y verificamos que ya no existe con los mismo comandos dentro de nuestro equipo:

**pkgman uninstall “Nombre paquete”**



## Federa:



```
liveuser@localhost-live:/home — sudo dnf install htop
liveuser@localhost-live:/home$ sudo dnf install htop
Updating and loading repositories:
Fedora 41 - x86_64 - Updates      100% | 912.1 KiB/s | 11.5 MiB | 00m13s
Fedora 41 - x86_64                100% | 3.2 MiB/s | 35.3 MiB | 00m11s
Fedora 41 openh264 (From Cisco) - x86_ 100% | 5.5 KiB/s | 5.8 KiB | 00m01s
Repositories loaded.
Package Arch Version Repository Size
Installing:
htop x86_64 3.3.0-4.fc41 fedora 448.6 KiB
Installing dependencies:
hwloc-libs x86_64 2.12.0-1.fc41 updates 2.9 MiB

Transaction Summary:
Installing: 2 packages

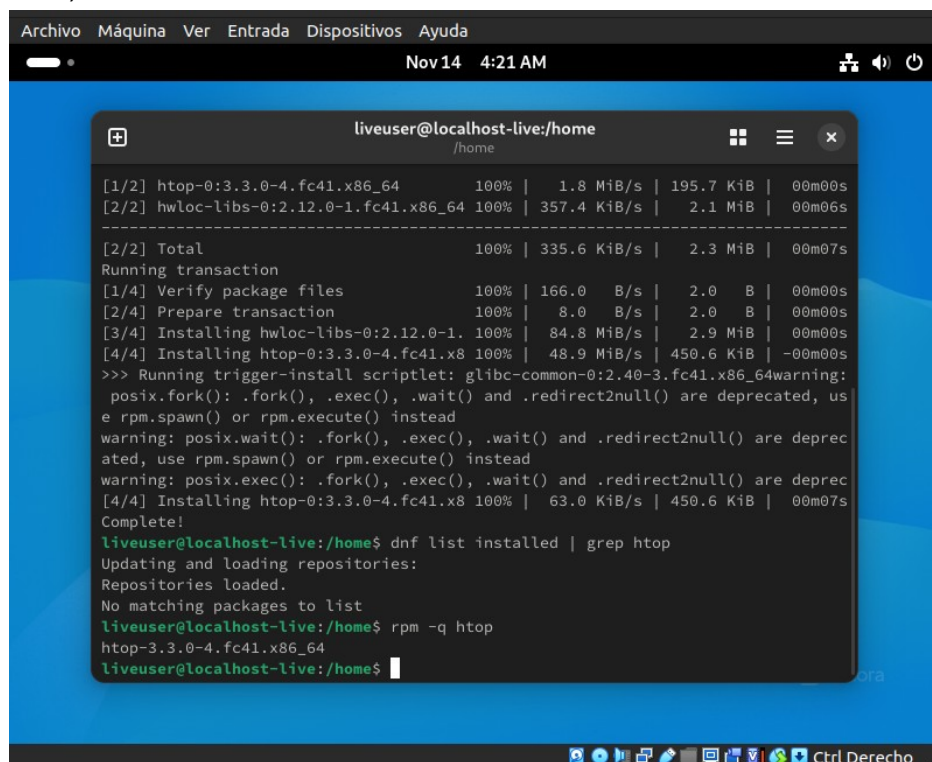
Total size of inbound packages is 2 MiB. Need to download 2 MiB.
After this operation, 3 MiB extra will be used (install 3 MiB, remove 0 B).
Is this ok [y/N]: y
[1/2] htop-0:3.3.0-4.fc41.x86_64 100% | 1.8 MiB/s | 195.7 KiB | 00m00s
[2/2] hwloc-libs-0:2.12.0-1.fc41.x86_64 0% | 0.0 B/s | 0.0 B | -02d20h
-----
[1/2] Total 8% | 0.0 B/s | 195.7 KiB | -00m12s
```

Realizamos una instalación del paquete **htop** con el siguiente comando y verificamos que si el paquete a sido instalado:

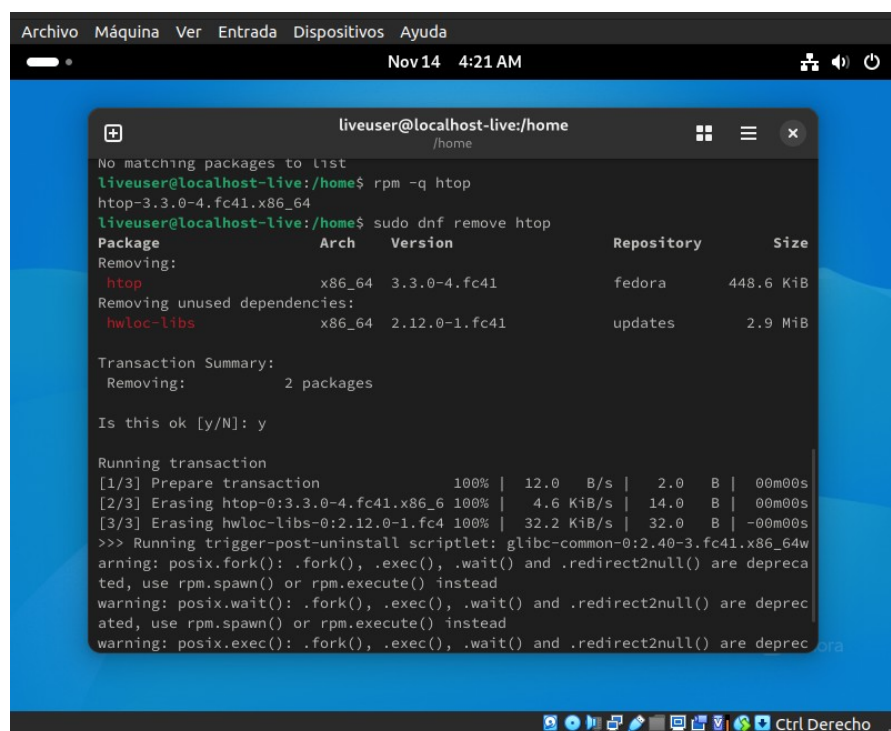
**sudo dnf install “Nombre del paquete”** (utilizamos dnf por el gestor de paquetes oficial, puesto que anteriormente se utilizaba otro gestor (yum) )

**dnf list installed | grep “Nombre del paquete”** (permite mostrar la lista de todos los paquetes instalados y lo filtramos con **grep**)

**rpm -q “Nombre del paquete”** (Busca directamente en la base de datos RPM si el paquete esta instalado)



```
liveuser@localhost-live:/home
[1/2] htop-0:3.3.0-4.fc41.x86_64 100% | 1.8 MiB/s | 195.7 KiB | 00m00s
[2/2] hwloc-libs-0:2.12.0-1.fc41.x86_64 100% | 357.4 KiB/s | 2.1 MiB | 00m06s
-----
[2/2] Total 100% | 335.6 KiB/s | 2.3 MiB | 00m07s
Running transaction
[1/4] Verify package files 100% | 166.0 B/s | 2.0 B | 00m00s
[2/4] Prepare transaction 100% | 8.0 B/s | 2.0 B | 00m00s
[3/4] Installing hwloc-libs-0:2.12.0-1.fc41.x86_64 100% | 84.8 MiB/s | 2.9 MiB | 00m00s
[4/4] Installing htop-0:3.3.0-4.fc41.x86_64 100% | 48.9 MiB/s | 450.6 KiB | -00m00s
>>> Running trigger-install scriptlet: glibc-common-0:2.40-3.fc41.x86_64warning:
posix.fork(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, us
e rpm.spawn() or rpm.execute() instead
warning: posix.wait(): .fork(), .exec(), .wait() and .redirect2null() are deprec
ated, use rpm.spawn() or rpm.execute() instead
warning: posix.exec(): .fork(), .exec(), .wait() and .redirect2null() are deprec
[4/4] Installing htop-0:3.3.0-4.fc41.x86_64 100% | 63.0 KiB/s | 450.6 KiB | 00m07s
Complete!
liveuser@localhost-live:/home$ dnf list installed | grep htop
Updating and loading repositories:
Repositories loaded.
No matching packages to list
liveuser@localhost-live:/home$ rpm -q htop
htop-3.3.0-4.fc41.x86_64
liveuser@localhost-live:/home$
```



```
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Nov14  4:21 AM

liveuser@localhost-live:/home
/home
No matching packages to list
liveuser@localhost-live:/home$ rpm -q htop
htop-3.3.0-4.fc41.x86_64
liveuser@localhost-live:/home$ sudo dnf remove htop
Package Arch Version Repository Size
Removing:
htop x86_64 3.3.0-4.fc41 fedora 448.6 KiB
Removing unused dependencies:
hwloc-libs x86_64 2.12.0-1.fc41 updates 2.9 MiB

Transaction Summary:
Removing: 2 packages

Is this ok [y/N]: y

Running transaction
[1/3] Prepare transaction 100% | 12.0 B/s | 2.0 B | 00m00s
[2/3] Erasing htop-0:3.3.0-4.fc41.x86_64 100% | 4.6 KiB/s | 14.0 B | 00m00s
[3/3] Erasing hwloc-libs-0:2.12.0-1.fc41 100% | 32.2 KiB/s | 32.0 B | 00m00s
>>> Running trigger-post-uninstall scriptlet: glibc-common-0:2.40-3.fc41.x86_64
warning: posix.fork(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
warning: posix.wait(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
warning: posix.exec(): .fork(), .exec(), .wait() and .redirect2null() are deprecated, use rpm.spawn() or rpm.execute() instead
```

Finalmente, eliminamos el paquete con **sudo dnf remove “Nombre del paquete”**.

## Reflexión:

En esta práctica, trabajar con Haiku y Fedora me permitió seguir ampliando mis conocimientos sobre distintos sistemas operativos, cada uno con sus particularidades y formas de gestión. En el caso de Haiku, aunque su interfaz gráfica facilita gran parte del proceso, encontré ciertos comandos y herramientas que difieren bastante de los entornos más comunes, por lo que fue necesario investigar algunas instrucciones básicas para realizar correctamente las tareas solicitadas. Aun así, resultó interesante descubrir un sistema menos conocido y comprender su estructura interna, lo cual considero útil para desarrollar una visión más amplia del ecosistema de sistemas operativos.

Por otro lado, Fedora presentó menos complicaciones gracias a su mayor soporte y documentación. Su entorno gráfico y la simplicidad del sistema facilitaron gran parte de la práctica. En retrospectiva, trabajar con ambos sistemas fue una experiencia enriquecedora que contribuyó significativamente a mi aprendizaje, permitiéndome enfrentar nuevas situaciones y resolver problemas de manera más autónoma. Aunque Haiku puede no ser el sistema más adecuado para ciertas actividades debido a sus limitaciones, considero que la práctica fue desafiante y muy retroalimentaria, ampliando mis habilidades para manejar diversos entornos.