

Sumario

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

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:

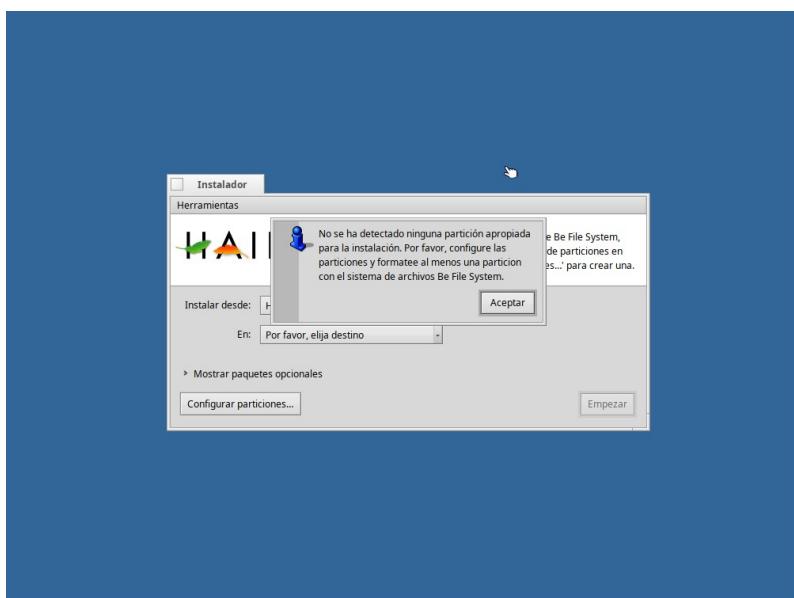
```
dam1@Carlos:~/Descargas$ sha256sum haiku-r1beta5-x86_64-anyboot.iso
22ae312a38e98083718b6984186e753d15806bd6ea44542144fdcef42c4dcb69 haiku-r1beta5-
```

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.

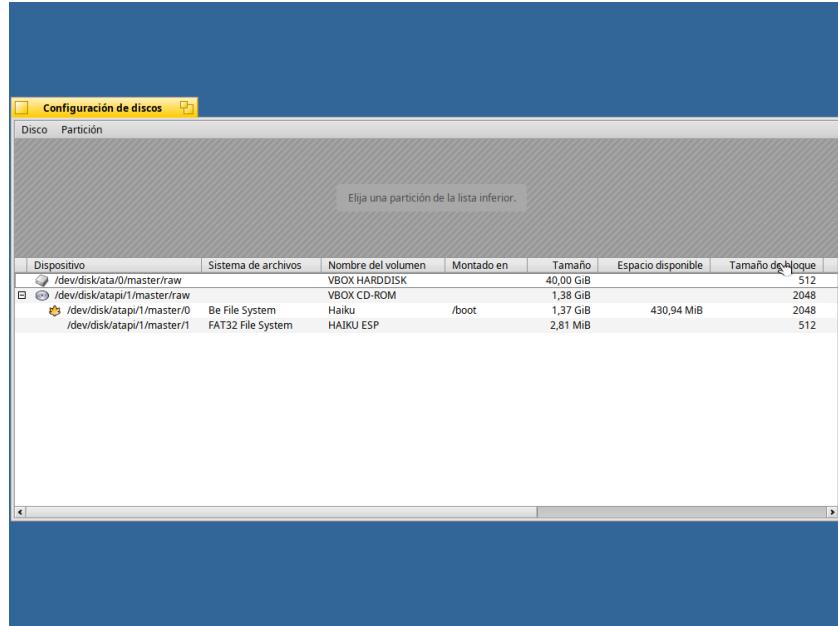
```
22ae312a38e98083718b6984186e753d15806bd6ea44542144fdcef42c4dcb69 haiku-r1beta5-x86_64-anyboot.iso
```

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)

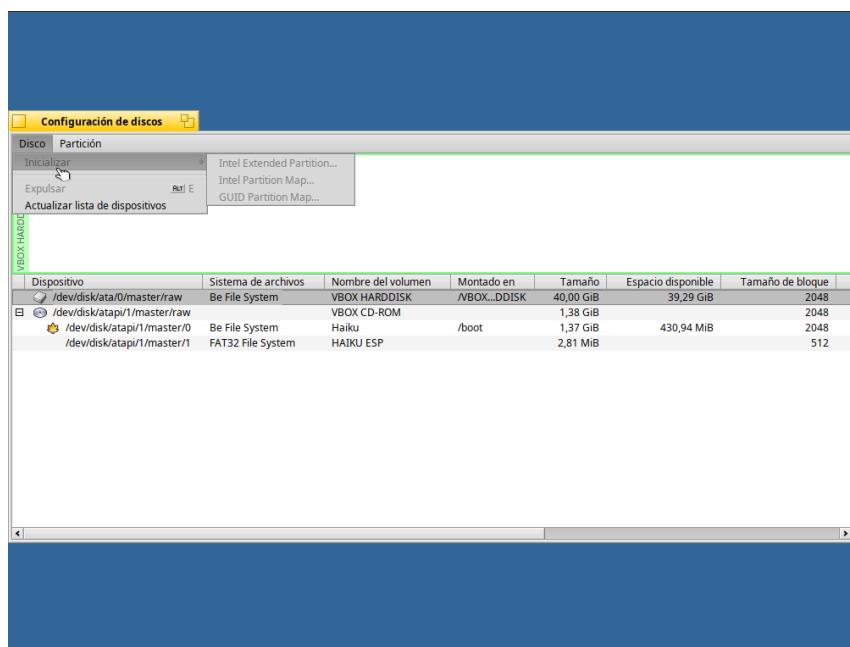


Aquí comenzaron los problemas.

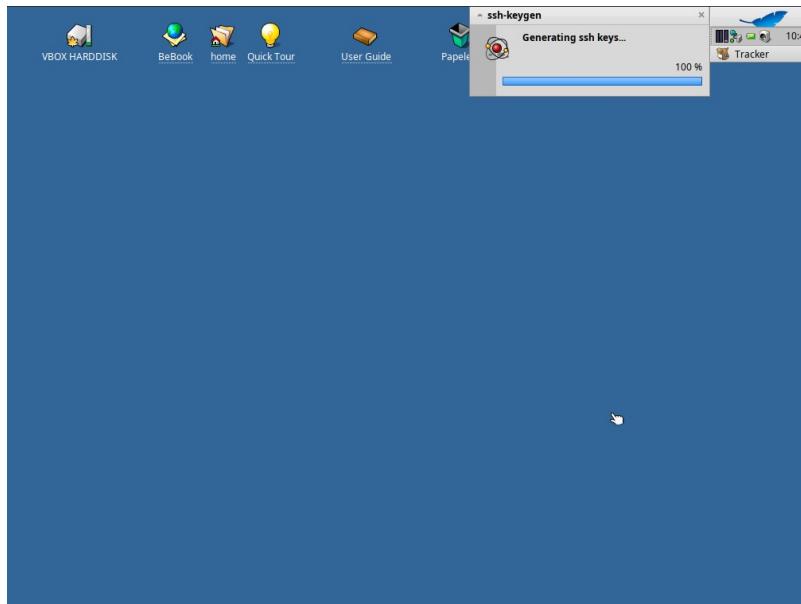
El sistema operativo Haiku utiliza un diferente sistema de archivos que no es común (Be file System o BFS), algunos discos duros no están configurados o formateados con ese sistema. Por lo que para solucionarlo y poder instalar la ISO hay que realizar dos pasos:



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.



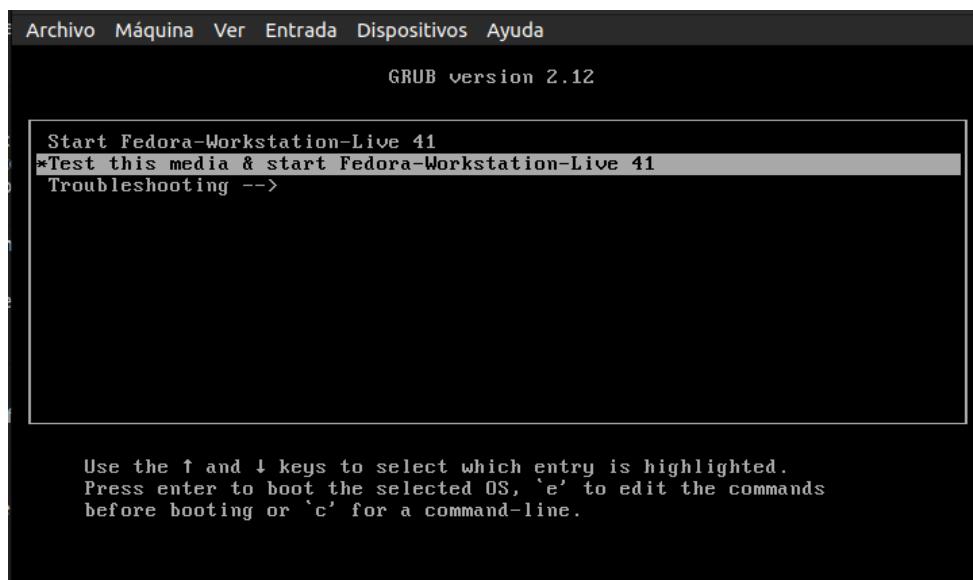
Al continuar, finalmente entramos al entorno gráfico de Haiku y nos permite seguir con la práctica.

Fedora:

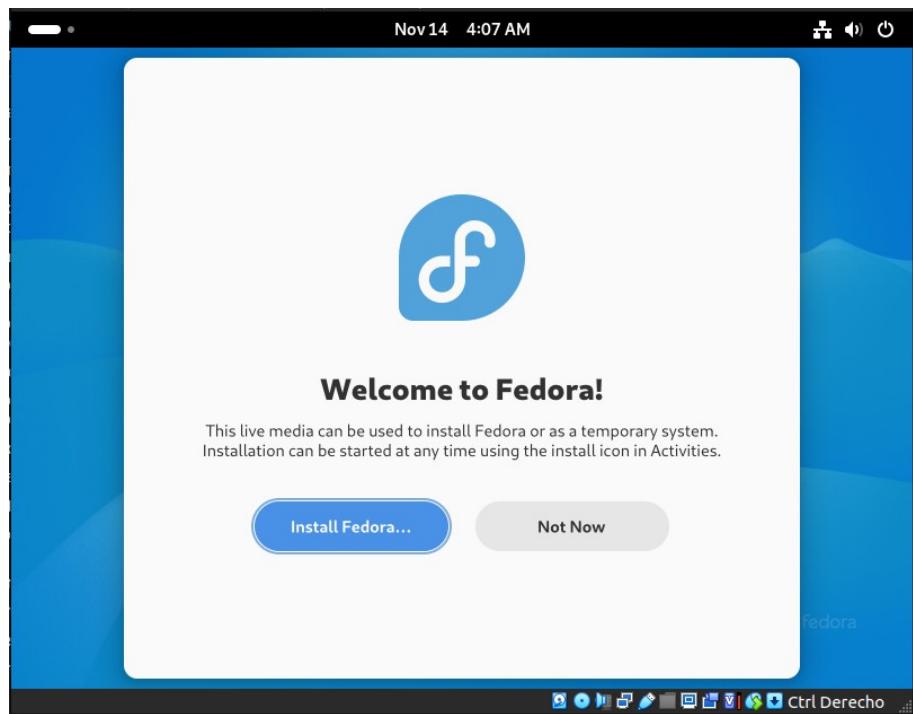
```
dam1@Carlos:~/Descargas$ sha256sum Fedora-Workstation-Live-x86_64-41-1.4.iso
a2dd3caf3224b8f3a640d9e31b1016d2a4e98a6d7cb435a1e2030235976d6da2  Fedora-Worksta
tion-Live-x86_64-41-1.4.iso

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

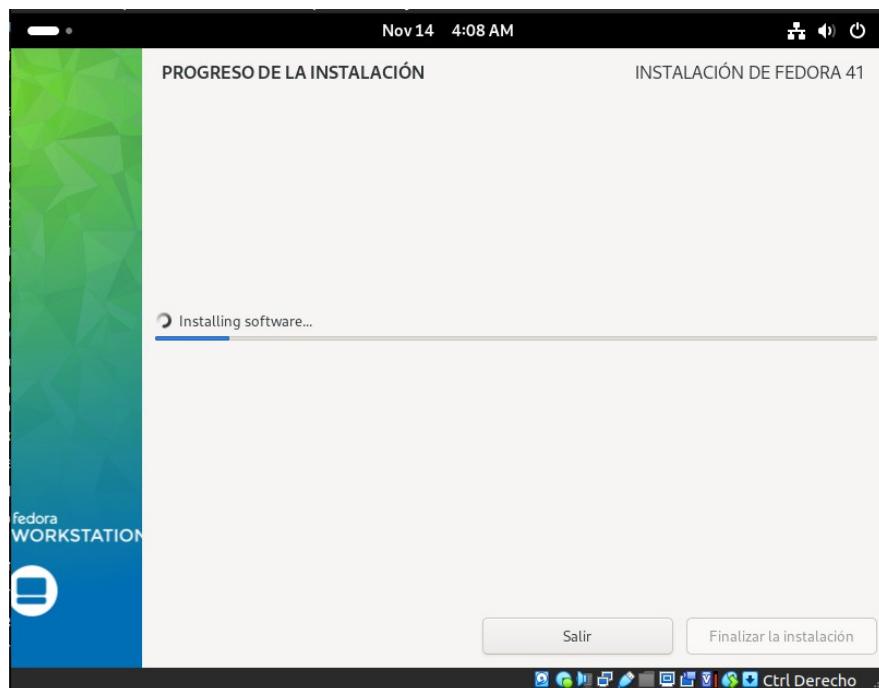
Como siempre, realizamos el checksum antes de comenzar la instalación para verificar si el archivo es fiable, de la página oficial.



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.

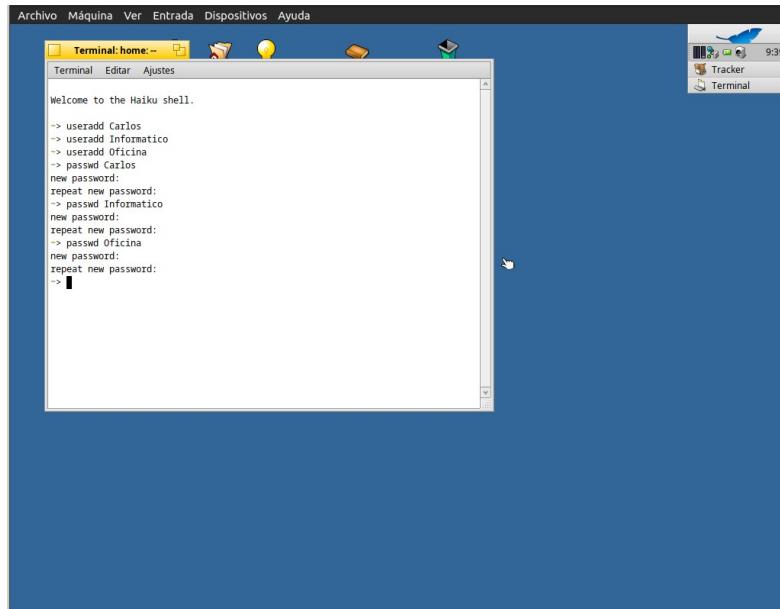


Se deben realizar una serie de configuraciones previas pero simples y luego, comienza con la instalación.

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:

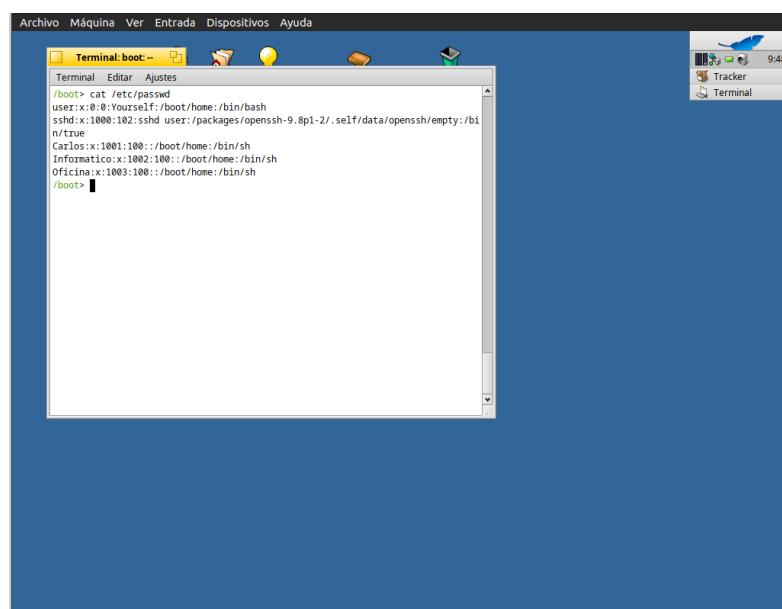


Utilizando comandos derivados de Unix, se pueden crear una serie de usuarios que a primera vista lo parecen, pero tiene un poco de trampa:

useradd “Nombre usuario”

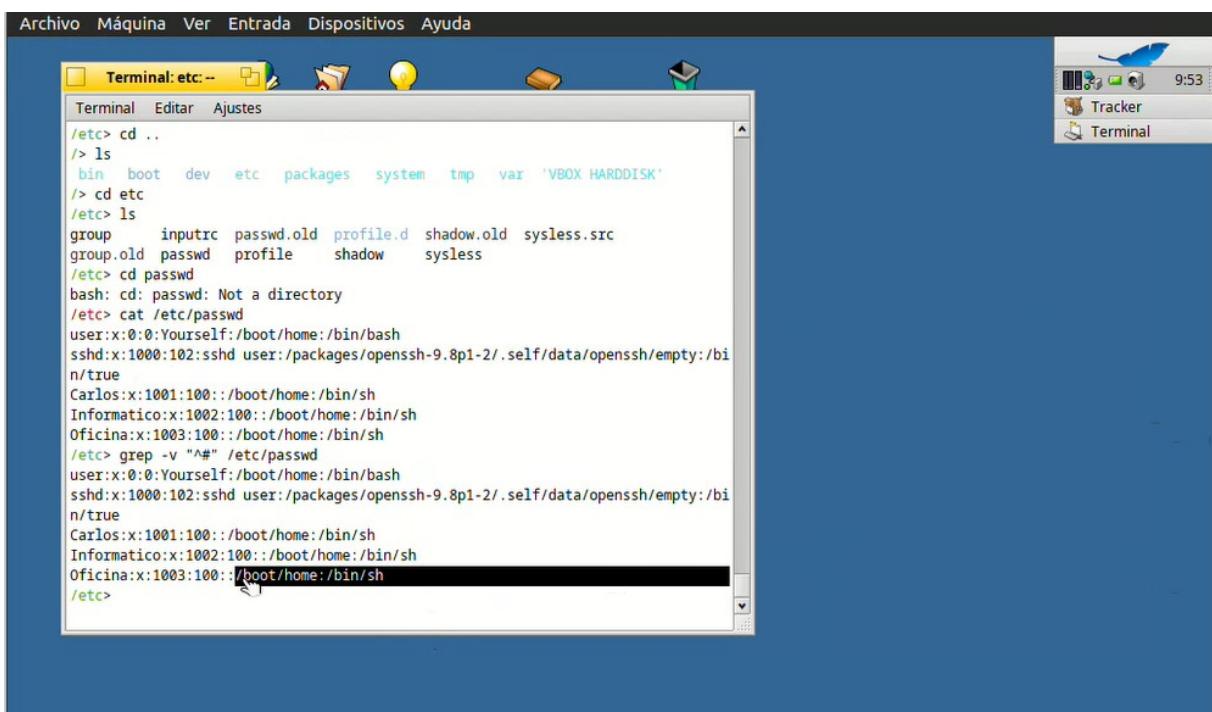
Sin embargo, se le pueden asignar contraseña aunque no sean por así decirlo cuentas independientes:

passwd “Nombre usuario”

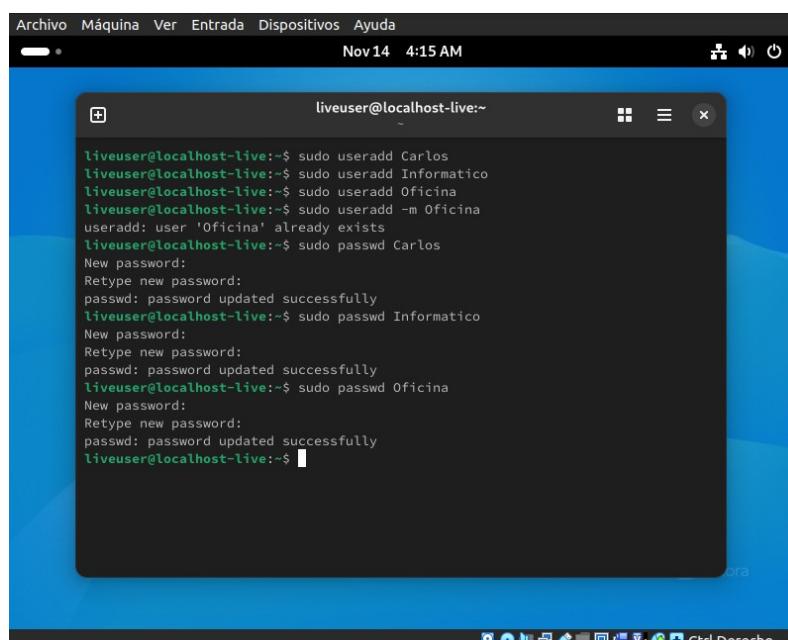


Explicación (Porque se puede utilizar comandos de creación de usuarios si el SO es monousuario):

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.



Fedora:



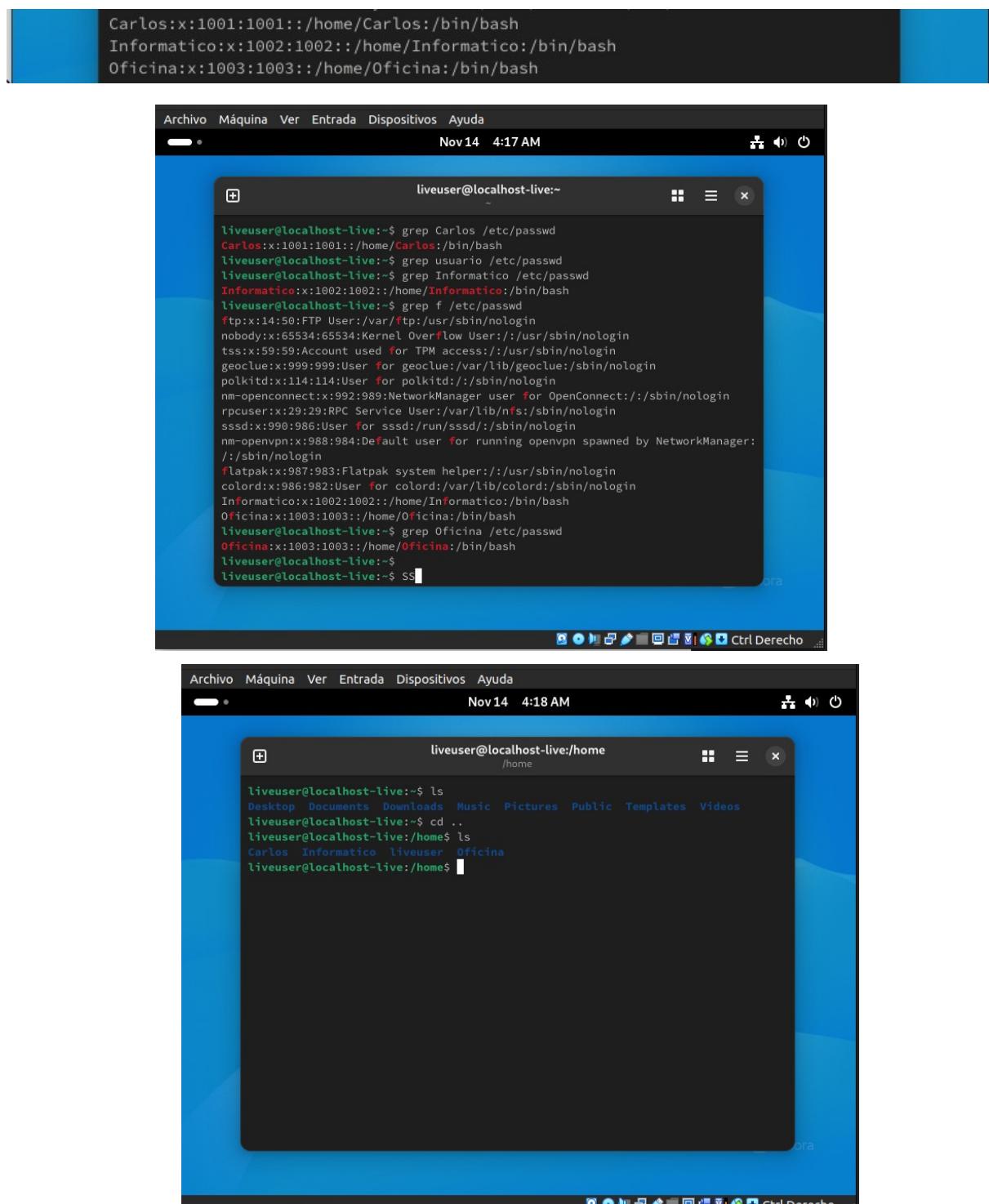
Mediante una serie de comandos, es posible crear usuarios independientes y asignarles contraseña: (el comando sudo permite tener privilegios al utilizar un comando)

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.



The image consists of three vertically stacked screenshots from a Linux desktop environment. The top screenshot shows a terminal window with the command 'cat /etc/passwd' output, listing three users: Carlos, Informatico, and Oficina. The middle screenshot shows a terminal window with the user 'liveuser' running several grep commands to search for these user names in the /etc/passwd file. The bottom screenshot shows a file browser window displaying the contents of the '/home' directory, where three home folders named 'Carlos', 'Informatico', and 'Oficina' are visible.

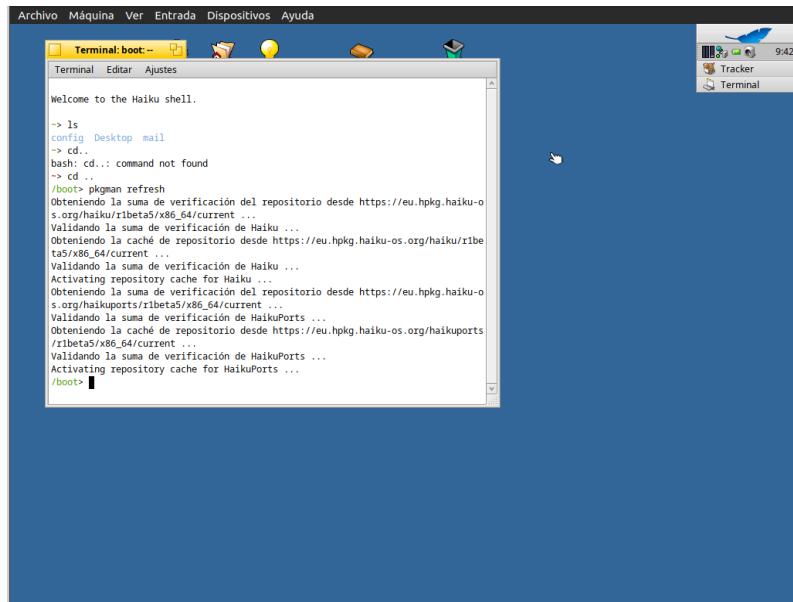
```
Carlos:x:1001:1001::/home/Carlos:/bin/bash
Informatico:x:1002:1002::/home/Informatico:/bin/bash
Oficina:x:1003:1003::/home/Oficina:/bin/bash

liveuser@localhost-live:~$ grep Carlos /etc/passwd
Carlos:x:1001:1001::/home/Carlos:/bin/bash
liveuser@localhost-live:~$ grep usuario /etc/passwd
liveuser@localhost-live:~$ grep Informatico /etc/passwd
Informatico:x:1002:1002::/home/Informatico:/bin/bash
liveuser@localhost-live:~$ grep f /etc/passwd
ftp:x:14:50:FTP User:/var/ftp:/usr/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/usr/sbin/nologin
tss:x:59:59:Account used for TPM access:/:/usr/sbin/nologin
geoclue:x:999:999:User for geoclue:/var/lib/geoclue:/sbin/nologin
polkitd:x:114:114:User for polkitd:/sbin/nologin
nm-openconnect:x:992:989:NetworkManager user for OpenConnect:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
sssd:x:990:986:User for sssd:/run/sssd:/sbin/nologin
nm-openvpn:x:988:984:Default user for running openvpn spawned by NetworkManager:/sbin/nologin
flatpak:x:987:983:Flatpak system helper:/usr/sbin/nologin
colorl:x:986:982:User for colorl:/var/lib/colorl:/sbin/nologin
Informatico:x:1002:1002::/home/Informatico:/bin/bash
Oficina:x:1003:1003::/home/Oficina:/bin/bash
liveuser@localhost-live:~$ grep Oficina /etc/passwd
Oficina:x:1003:1003::/home/Oficina:/bin/bash
liveuser@localhost-live:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
liveuser@localhost-live:~$ cd ..
liveuser@localhost-live:/home$ ls
Carlos Informatico liveuser Oficina
liveuser@localhost-live:/home$
```

Apartado 3. Instalación y configuración de una aplicación

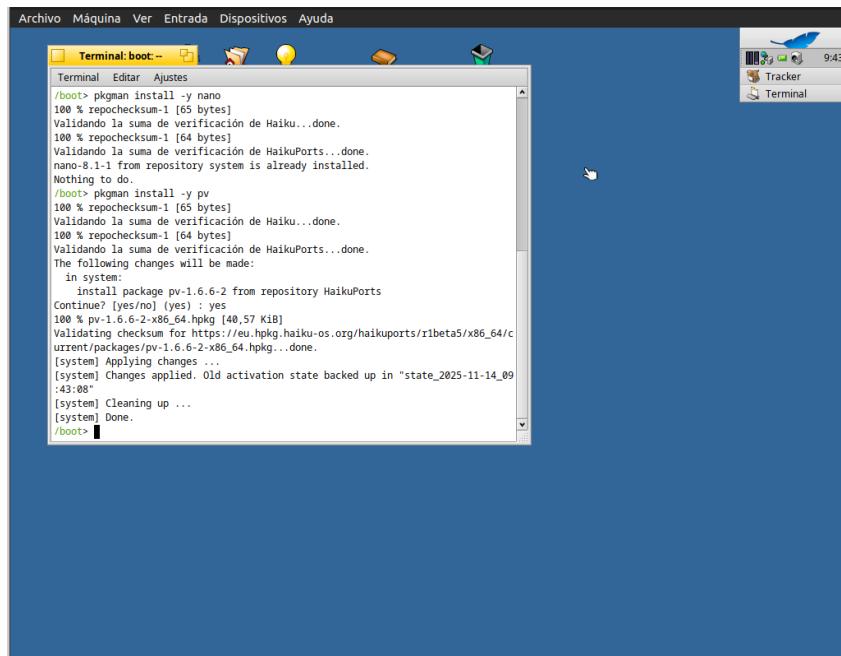
Instalación de cualquier paquete mediante terminal. Detallar instalación, comprobación y desinstalación de un paquete.

Haiku:



Primeramente, realizamos una actualización de repositorios mediante el comando:

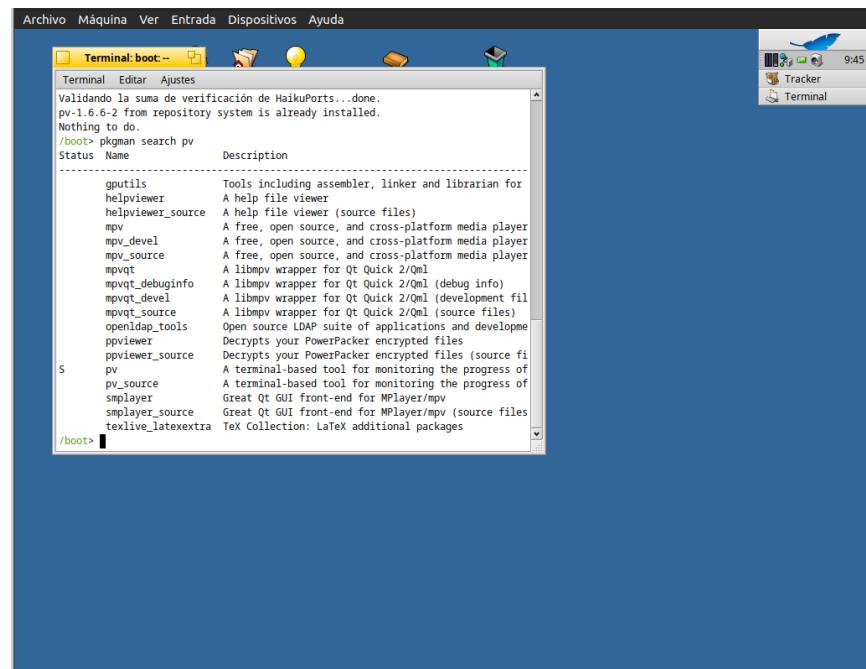
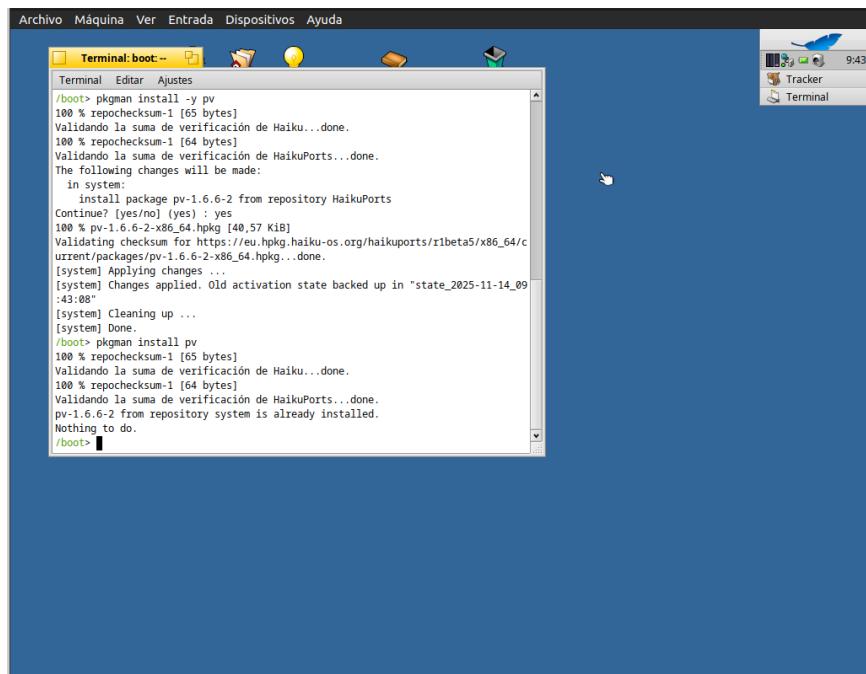
pkgman refresh



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”



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)

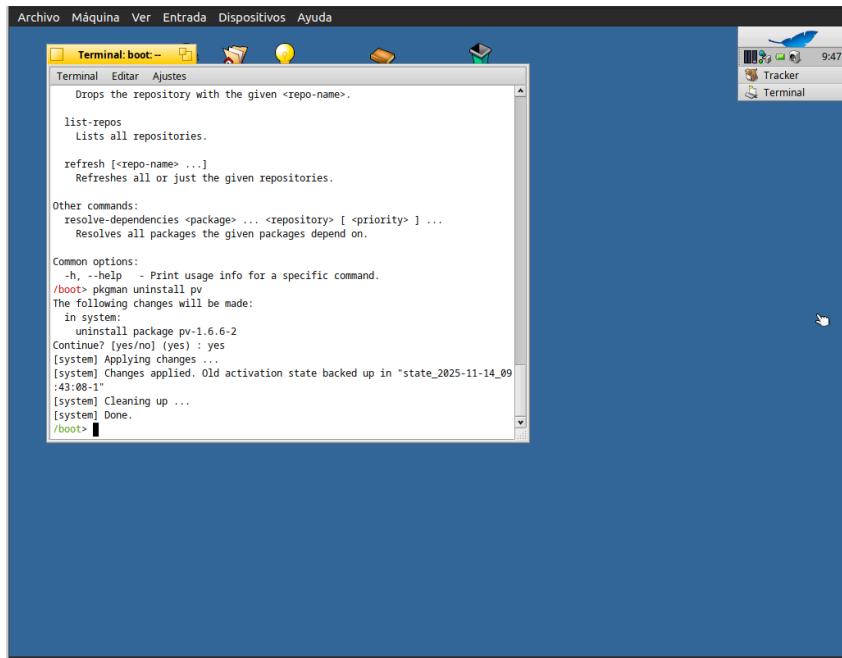
```
Archivo Máquina Ver Entrada Dispositivos Ayuda
Terminal: boot:-- Terminal Editar Ajustes
S openldap_tools Open source LDAP suite of applications and development tools
ppviewer Decrypts your PowerPacker encrypted files
ppviewer_source Decrypts your PowerPacker encrypted files (source files)
pv A terminal-based tool for monitoring the progress of data through a pipeline
pv_source A terminal-based tool for monitoring the progress of data through a pipeline
smplayer Great QT GUI front-end for MPlayer/mpv
smplayer_source Great QT GUI front-end for MPlayer/mpv (source files)
texlive_latextra TeX Collection: LaTeX additional packages
/boots pkman list | grep pv
/boots pkman info pv
pv: A terminal-based tool for monitoring the progress of data through a pipeline
pv - Pipe Viewer - is a terminal-based tool for monitoring the progress of data through a pipeline. It can be inserted into any normal pipeline between two processes to give a visual indication of how quickly data is passing through, how long it has taken, how near to completion it is, and an estimate of how long it will be until completion. Additional support is available for multiple instances working in tandem, to give a visual indicator of relative throughput in a complex pipeline.
/boots> which pv
/bin/pv
/boots> ls -l /system/bin/pv
-r-xr-xr-x 1 user root 70136 ago 6 2018 /system/bin/pv
/boots>
```

Otros comandos que permiten verificar la existencia de un paquete son:

ls -l /system/bin/"Nombre del paquete" (Permite ver el archivo en binario)

"Nombre del paquete" –version (En algunos paquetes, permite ver su versión)

```
Archivo Máquina Ver Entrada Dispositivos Ayuda
Terminal: boot:-- Terminal Editar Ajustes
pv - Pipe Viewer - is a terminal-based tool for monitoring the progress of data through a pipeline. It can be inserted into any normal pipeline between two processes to give a visual indication of how quickly data is passing through, how long it has taken, how near to completion it is, and an estimate of how long it will be until completion. Additional support is available for multiple instances working in tandem, to give a visual indicator of relative throughput in a complex pipeline.
/boots> which pv
/bin/pv
/boots> ls -l /system/bin/pv
-r-xr-xr-x 1 user root 70136 ago 6 2018 /system/bin/pv
/boots> pv --version
pv 1.6.6 - Copyright 2015 Andrew Wood <andrew.wood@ivarch.com>
Web site: http://www.ivarch.com/programs/pv.shtml
This program is free software, and is being distributed under the terms of the Artistic License 2.0.
This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
/boots>
```



```
Archivo Máquina Ver Entrada Dispositivos Ayuda
Terminal: boot -- Terminal Editar Ajustes
Drops the repository with the given <repo-name>.

list-repos
Lists all repositories.

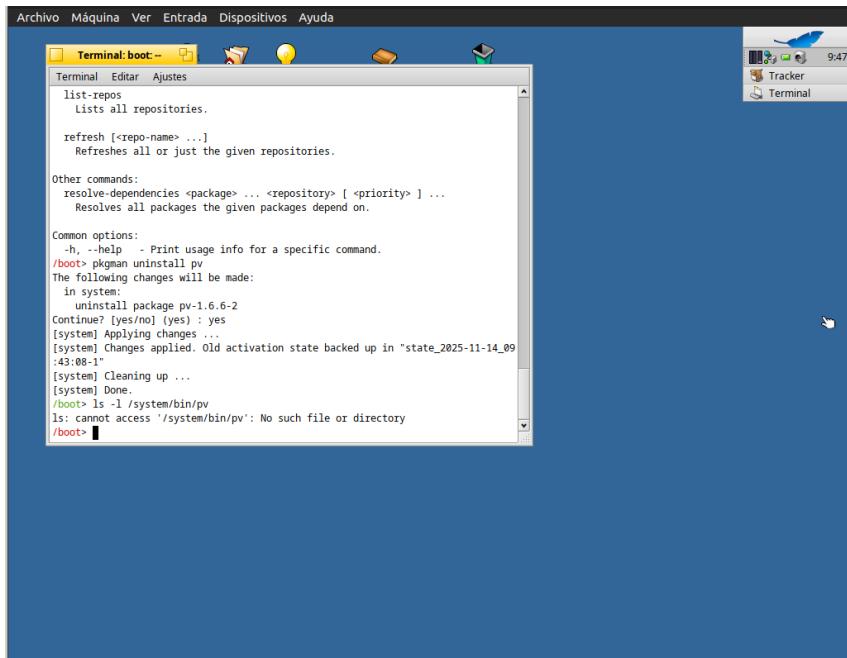
refresh [<repo-name> ...]
Refreshes all or just the given repositories.

Other commands:
resolve-dependencies <package> ... <repository> [ <priority> ] ...
Resolves all packages the given packages depend on.

Common options:
-h, --help - Print usage info for a specific command.
/boot> pkgman uninstall pv
The following changes will be made:
in system:
uninstall package pv-1.6.6-2
Continue? [yes/no] (yes) : yes
[system] Applying changes ...
[system] Changes applied. Old activation state backed up in "state_2025-11-14_09:43:08-1"
[system] Cleaning up ...
[system] Done.
[boot]>
```

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

pkgman uninstall "Nombre paquete"



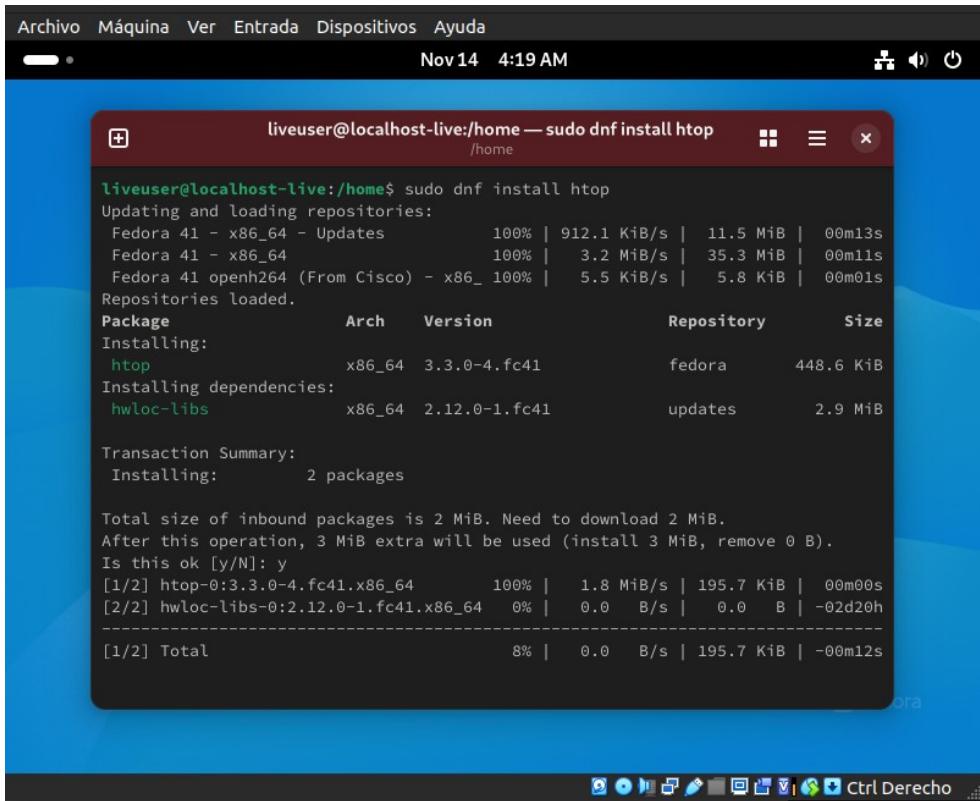
```
Archivo Máquina Ver Entrada Dispositivos Ayuda
Terminal: boot -- Terminal Editar Ajustes
list-repos
Lists all repositories.

refresh [<repo-name> ...]
Refreshes all or just the given repositories.

Other commands:
resolve-dependencies <package> ... <repository> [ <priority> ] ...
Resolves all packages the given packages depend on.

Common options:
-h, --help - Print usage info for a specific command.
/boot> pkgman uninstall pv
The following changes will be made:
in system:
uninstall package pv-1.6.6-2
Continue? [yes/no] (yes) : yes
[system] Applying changes ...
[system] Changes applied. Old activation state backed up in "state_2025-11-14_09:43:08-1"
[system] Cleaning up ...
[system] Done.
[boot]> ls -l /system/bin/pv
ls: cannot access '/system/bin/pv': No such file or directory
[boot]>
```

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_64 100% | 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

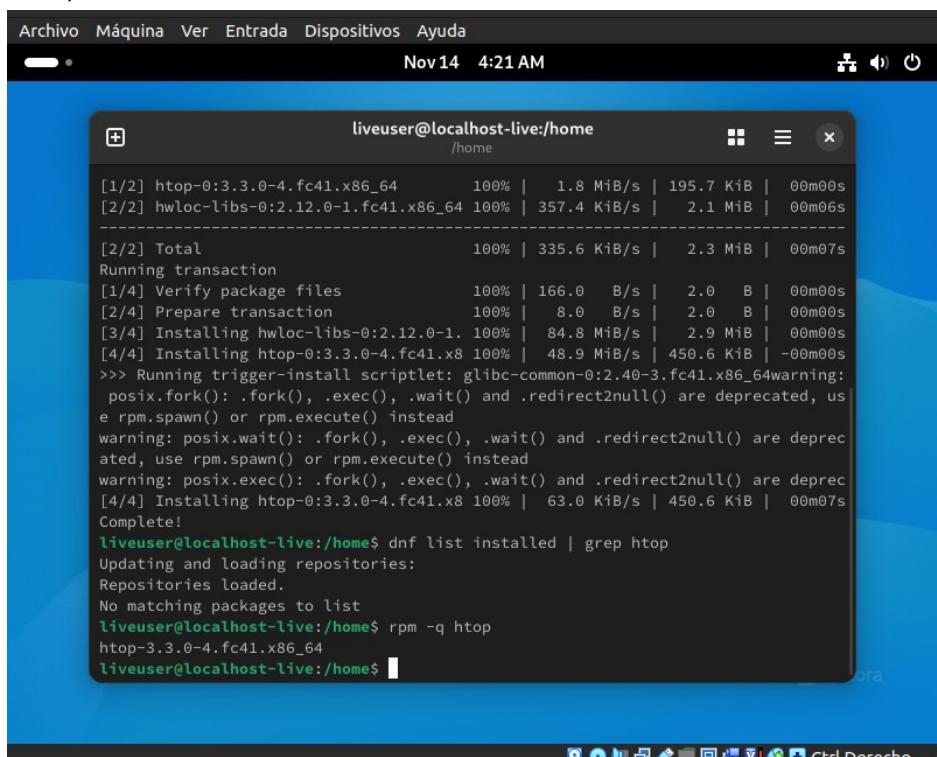
Total size of inbound packages is 2 MiB. Need to download 2 MiB.
After this operation, 3 MiB extra will be used (install 3 MiB, remove 0 B).
Is this ok [y/N]: y
[1/2] htop-0:3.3.0-4.fc41.x86_64      100% | 1.8 MiB/s | 195.7 KiB | 00m00s
[2/2] hwloc-libs-0:2.12.0-1.fc41.x86_64  0% | 0.0 B/s | 0.0 B |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

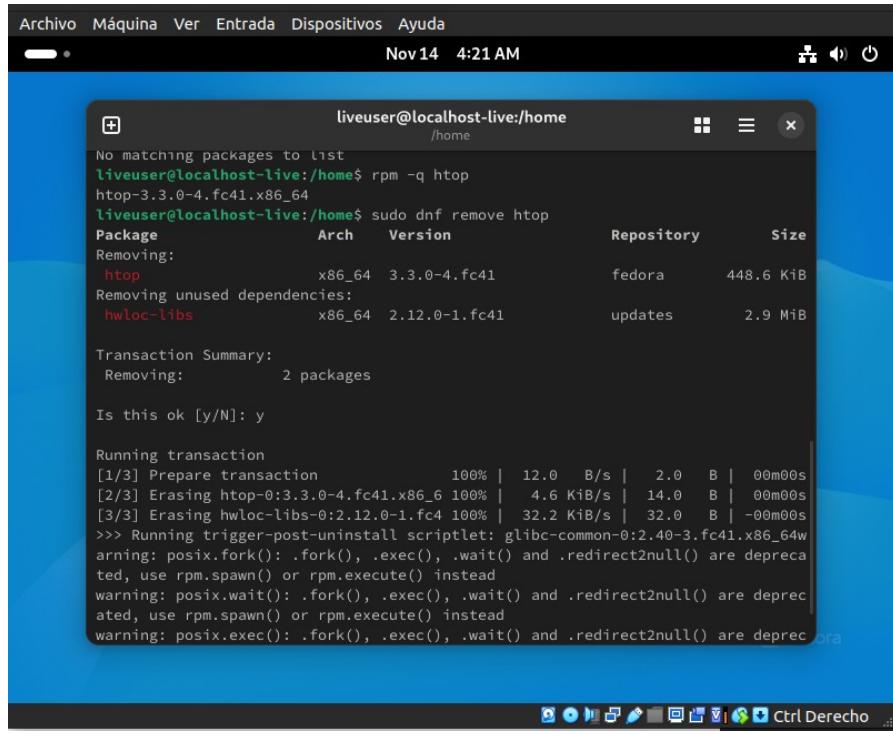
sudo dnf install “Nombre del paquete” (utilizamos dnf por el gestor de paquetes oficial, puesto que anteriormente se utilizaba otro gestor (yum))

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
liveuser@localhost-live:/home$ dnf list installed | grep htop
[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. 100% | 84.8 MiB/s | 2.9 MiB | 00m00s
[4/4] Installing htop-0:3.3.0-4.fc41.x8 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, use
  rpm.spawn() or rpm.execute() instead
warning: posix.wait(): .fork(), .exec(), .wait() and .redirect2null() are depre-
ated, use rpm.spawn() or rpm.execute() instead
warning: posix.exec(): .fork(), .exec(), .wait() and .redirect2null() are depre-
[4/4] Installing htop-0:3.3.0-4.fc41.x8 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$
```



The screenshot shows a terminal window on the Haiku desktop environment. The window title is "liveuser@localhost-live:/home". The terminal output shows the user running the command "sudo dnf remove htop". The process lists the package "htop" for removal, along with its dependency "hwloc-libs". The transaction summary indicates 2 packages are being removed. The user confirms with "Is this ok [y/N]:" followed by "y". The transaction then begins, showing progress for preparing the transaction, erasing the package files, and running trigger scripts. A warning message about deprecated POSIX functions like fork() and exec() is displayed during the scriptlet execution.

```
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_6 100% | 4.6 KiB/s | 14.0 B | 00m00s
[3/3] Erasing hwloc-libs-0:2.12.0-1.fc4 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
```

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.