



**Sistemas Operativos III**  
**DAYSWARE**

**Integrantes**

ROL	Apellido	Nombre	Email	Teléfono
Coordinador	Gutierrez	Tadeo	tgperciante@gmail.com	092112099
Subcoordinador	Navarro	Sebastian	navarrosebastian2003@gmail.com	098044013
Integrante	Cruz	Mateo	idkmateocruz@gmail.com	093577941
Integrante	Sanchez	Diego	diego612sanchezz@gmail.com	097737079

**Docente: Luis Fagúndez**

**Fecha de culminación:**  
**09/09/22**

**Segunda Entrega**

# ÍNDICE

<b>Manual de instalación Fedora Server:</b>	<b>2</b>
Instalación de Docker:	<b>13</b>
Instalación de mysql en Docker:	<b>16</b>
Instalación del servidor LAMP:	<b>18</b>
Instalación de MySQL/MariaDB:	<b>25</b>
Instalación de ssh en Fedora Server (servidor)	<b>32</b>
<b>Instalación de cliente ssh en windows</b>	<b>34</b>
<b>Anexo:</b>	<b>36</b>
Estudio de los roles de usuarios del sistema:	<b>36</b>
Sistema operativo del equipo:	36
Sistema operativo del servidor:	37

## Manual de instalación Fedora Server:

Bienvenido al manual de instalación de Fedora Server, a continuación, vamos a estar explicando la instalación correcta del sistema para nuestro servidor, detallando cada parte de la instalación lo más claro y precisa posible para que pueda ser un manual claro y detallado para el usuario final o la empresa correspondiente.

### 1)Menú principal de instalación



Antes de iniciar con el proceso de instalación tenemos que definir varias opciones para que nuestro servidor sea el más eficiente a la hora de cumplir la función de servidor LAMP por ejemplo. A continuación, detallamos los requisitos mínimos para que el sistema operativo funcione correctamente:

Requisitos mínimos de Fedora Server:

CPU: 1 GHz.

RAM: 4 GB.  
Disco duro: 20 o 30 GB.  
Conexión a Internet opcional.  
Tarjeta gráfica compatible con VGA.

## 1) Configuración del root

A) Primero vamos a configurar el usuario root desde el menú de instalación de fedora, para eso bajamos y hacemos clic en la última opción de abajo en donde dice “AJUSTES DE USUARIO”:



B) Seleccionamos la opción cuenta root y nos va a aparecer la siguiente interfaz:



Al principio aparecen las dos opciones para activar el usuario root, por lo general está desactivado, pero activamos la cuenta de root en la última casilla y agregamos la contraseña administrativa que sea adecuada y segura para el manejo del servidor como superusuario, por último, damos en el botón de hecho y nos lleva al menú principal.

## 2) Particiones del sistema

En esta parte detallamos claramente cuál es la forma de particionar el disco de nuestro servidor manualmente para que podamos personalizar los tamaños de la instalación y las ubicaciones de donde va a instalarse cada parte necesaria del sistema

Para acceder a la opción de particionar, en el menú principal damos clic en la opción “destino de la instalación” que está ubicada en la categoría de Sistema:

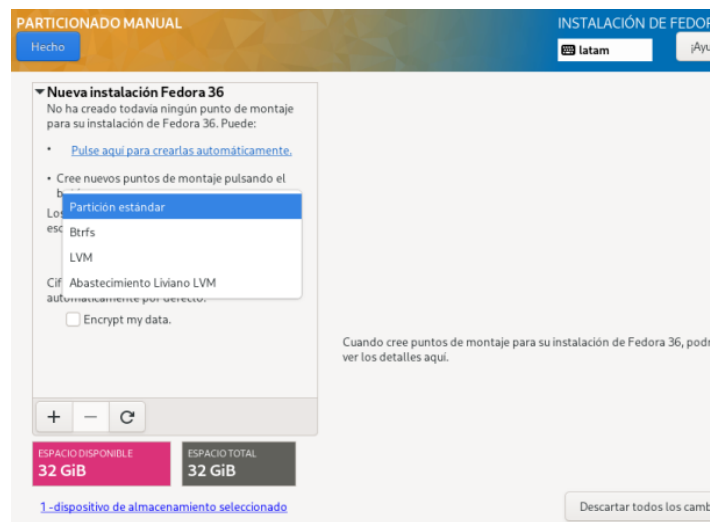


Una vez dentro de la selección del particionado nos van a aparecer múltiples opciones:

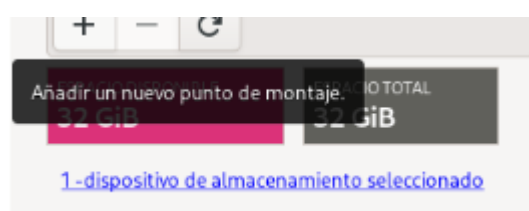


Dentro de la misma elegimos la unidad de almacenamiento en la que vamos a instalar y seleccionamos la configuración de almacenamiento personalizada. Una vez seleccionada la damos en hecho y nos aparece la tabla para crear un punto de montaje para nuestro sistema.

Una vez en la creación del punto de montaje, seleccionamos el esquema de particionado que vamos a utilizar, seleccionamos la opción de particionado Estándar:



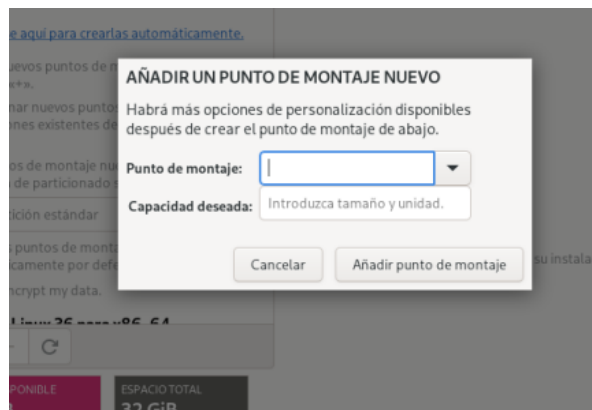
Una vez hecho esto comenzamos a agregar las siguientes particiones desde el botón de agregar arriba del espacio disponible:



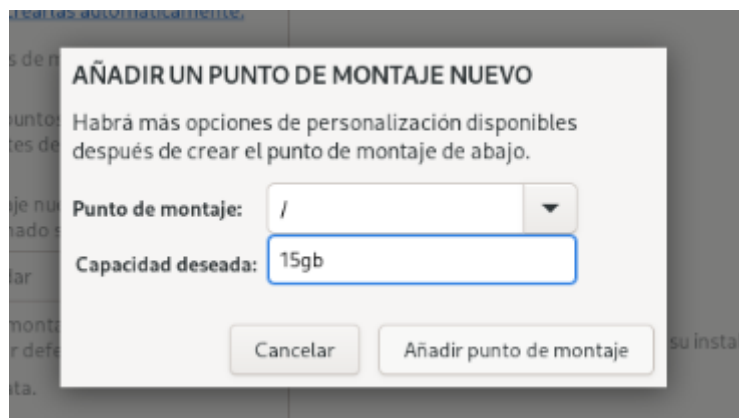
## Partición del sistema [ / ]

Para la siguiente partición asignamos el tamaño adecuado para que permita la creación de las demás particiones que faltan. En este caso yo tengo un disco de 32gb como prueba, pero para el servidor se va a utilizar mínimo una unidad de 1tb.

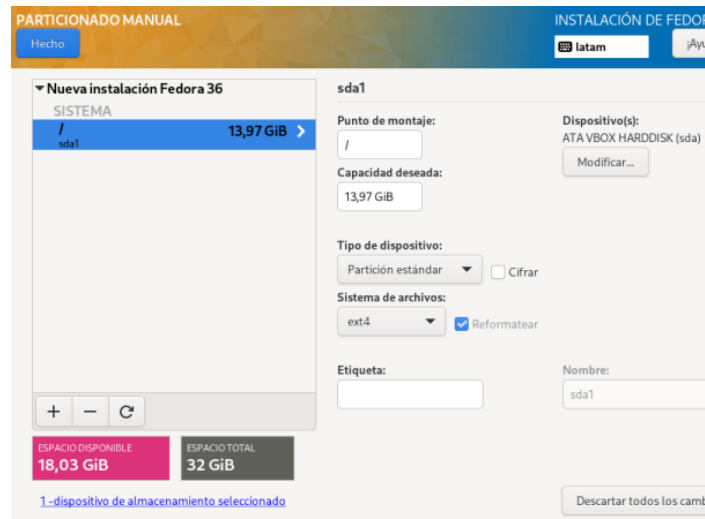
Hacemos clic en el botón para añadir un montaje y nos aparece lo siguiente:



En la sección de punto de montaje seleccionamos el / a continuación, seleccionamos también el tamaño que le vamos a asignar a dicha partición, en el caso del servidor se le va a asignar 100gb.

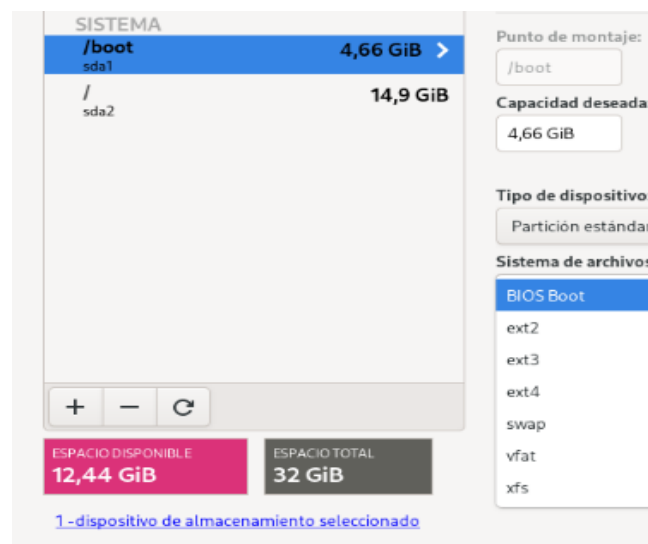


Damos en añadir y listo se nos va a crear la partición y debería de aparecer de la siguiente forma:



## Partición del sistema [ /boot ]

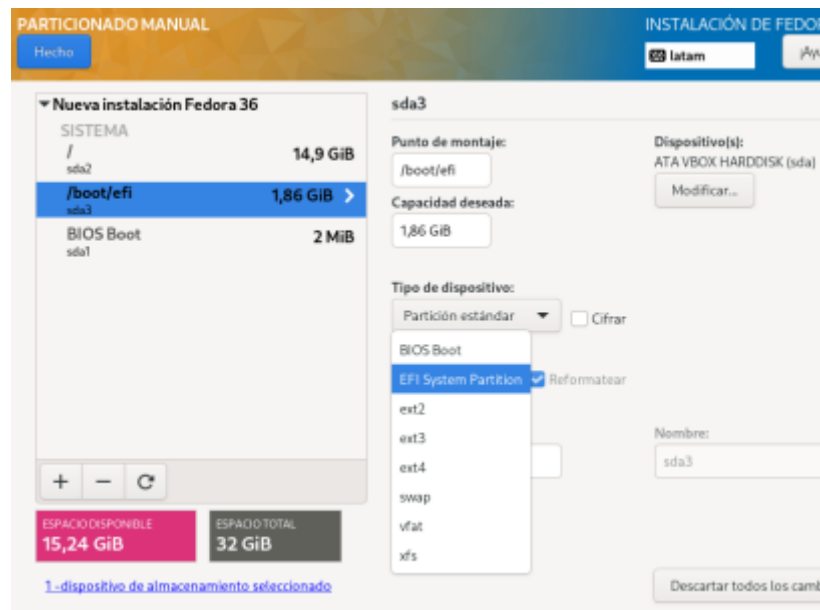
Partición del sistema que es para el arranque del servidor, asignamos el tamaño indicado para el servidor y cuando lo agregamos marcamos el sistema de archivos como BIOS Boot:



## Partición del sistema [ /boot/efi ]

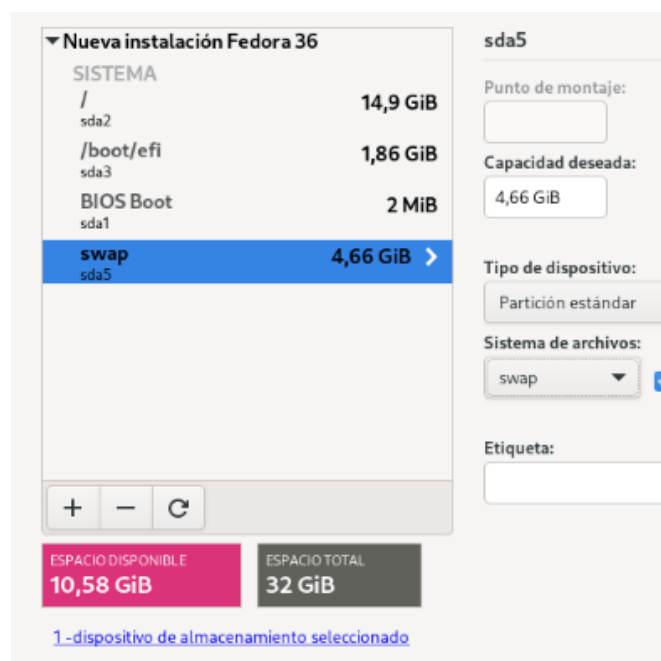
Esta partición es necesaria si no tenemos el bios tradicional y contamos con el estándar UEFI, asignamos lo mismo que la partición /boot (cuando lo creamos el tamaño de la partición bios boot se reduce automáticamente):





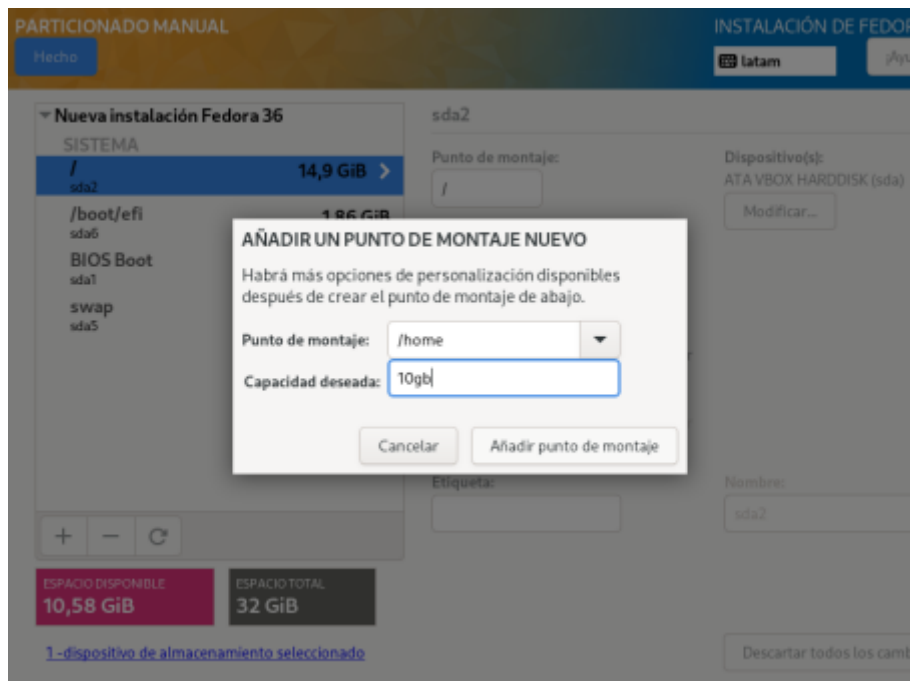
## Partición del sistema swap [ /swap ]

Esta partición es la memoria virtual del sistema, asignamos el punto de montaje /swap/ y le damos un tamaño indicado, en su sistema de archivos le damos a swap y continuamos

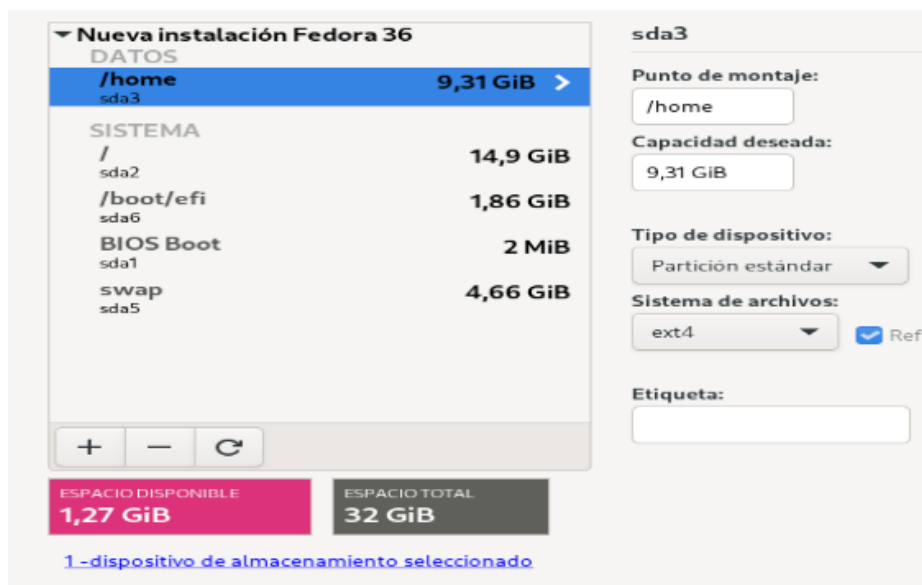


## Partición de los datos [ /home/ ]

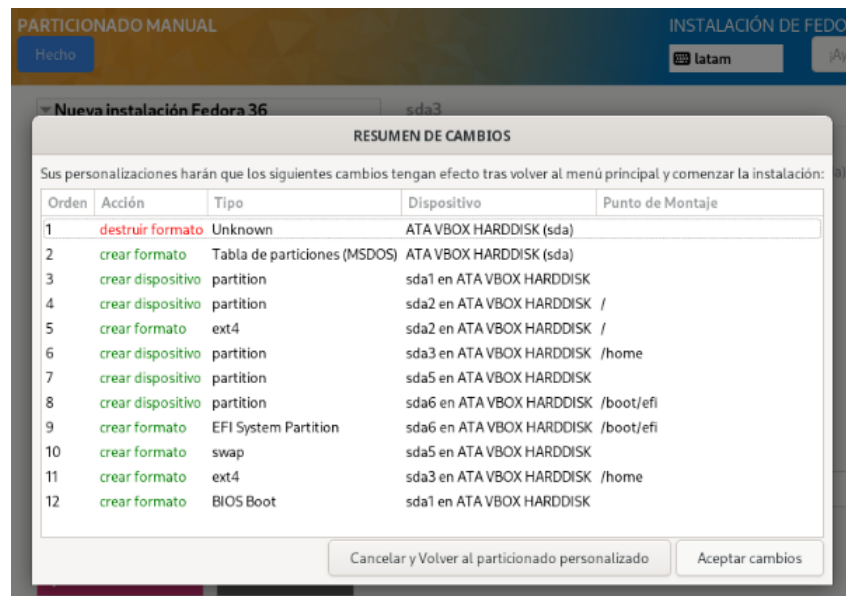
Es la partición que va a contener los datos del usuario y es la que se puede asignar mas espacio, agregamos la partición con un sistema de archivos ext4 y le asignamos el espacio indicado.



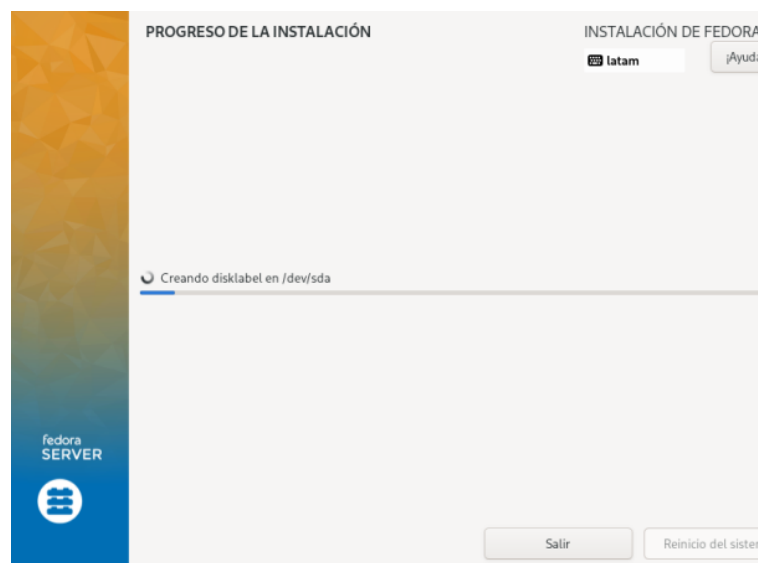
Luego de la foto añadimos el sistema de archivos que va a manejar:



Listo, ya terminamos las particiones personalizadas del sistema, le damos en hecho en la parte superior izquierda para aplicar los cambios y nos aparece el siguiente cartel:



Damos en aceptar cambios y nos vuelve al menú principal de la instalación, una vez en el menú lo que hacemos es comenzar la instalación del sistema, la duración de este proceso depende más que nada de la cantidad de memoria RAM que tenga asignada nuestro servidor.



Una vez completado reiniciamos el equipo

Una vez reiniciado, tenemos instalado el sistema en nuestro servidor, lo siguiente que podemos hacer (opcional) es instalar la interfaz gráfica

```
Fedora Linux 36 (Server Edition)
Kernel 5.17.5-300.fc36.x86_64 on an x86_64 (tty1)

Web console: https://10.0.2.15:9090/ or https://10.0.2.15:9090/

10 login: _
```

## Instalar interfaz gráfica:

### 1)dnf -y update

Este comando es para actualizar los paquetes de nuestro sistema, es importante a la hora de instalar la interfaz grafica:

```
Fedora Linux 36 (Server Edition)
Kernel 5.17.5-300.fc36.x86_64 on an x86_64 (tty1)

Web console: https://10.0.2.15:9090/ or https://10.0.2.15:9090/

10 login: root
Password:
Last login: Thu Jul  7 22:56:33 on tty1
[root@10 ~]# dnf -y update
Fedora 36 - x86_64                               2.9 MB/s | 81 MB      00:27
```

### 2)dnf -y install @GNOME-desktop

Una vez actualizados los paquetes de nuestro sistema, procedemos a instalar la interfaz gráfica, esto nos puede tomar tiempo pero al final va a quedar instalada:

```
mtc001s-4.0.40-1.fc36.x86_64      python3-gobject-base-noarch-3.42.1-1.fc36.noarch
reportd-0.7.4-8.fc36.x86_64       rpm-plugin-systemd-inhibit-4.17.0-10.fc36.x86_64

¡Listo!
[root@10 ~]# dnf -y install @GNOME-desktop
```

### 3) echo "exec /usr/bin/gnome-session" >> ~/.xinitrc

Hacemos el siguiente comando para que quede en el archivo de arranque del servidor que inicie sola la interfaz gráfica

Paso después reiniciamos el equipo con el comando reboot

```

uekkit2gtk3-2.36.4-1.fc36.x86_64
uekkit2gtk3-jac-2.36.4-1.fc36.x86_64
uekrte-audio-processing-0.3.1-0.fc36.x86_64
uireplaylib-0.4.11-1.fc36.x86_64
uireplayliber-libs-0.4.11-1.fc36.x86_64
uoff2-1.0-2-14.fc36.x86_64
upackbackend-fdo-1.12.0-2.fc36.x86_64
usb-attil-0.4.0-19.fc36.x86_64
xdg-dbus-proxy-0.1.3-1.fc36.x86_64
xdg-desktop-portal-1.12.4-1.fc36.x86_64
xdg-desktop-portal-gnome-42.1-1.fc36.x86_64
xdg-desktop-portal-gtk-1.12.0-5.fc36.x86_64
xdg-user-dirs-0.17-10.fc36.x86_64
xdg-user-dirs-gtk-0.10-22.fc36.x86_64
xen-libs-4.16.1-4.fc36.x86_64
xen-licenses-4.16.1-4.fc36.x86_64
xhost-1.0.0-3.fc36.x86_64
xhcomp-1.4.5-3.fc36.x86_64
xul-common-0.6.3-50.fc36.noarch
xulsec-1.2.33-2.fc36.x86_64
xmodmap-1.0.10-3.fc36.x86_64
xorg-x11-drv-libinput-1.2.1-1.fc36.x86_64
xorg-x11-server-Xorg-1.20.14-6.fc36.x86_64
xorg-x11-server-Wayland-22.1.2-1.fc36.x86_64
xorg-x11-server-common-1.20.14-6.fc36.x86_64
xorg-x11-xauth-1.1.1-2.fc36.x86_64
xorg-x11-xinit-1.4.0-14.fc36.x86_64
xrdp-1.2.1-3.fc36.x86_64
yaji-2.1.0-10.fc36.x86_64
yelp-2:42.1-1.fc36.x86_64
yelp-libs-2:42.1-1.fc36.x86_64
yelp-xsl-42.0-1.fc36.noarch
zenity-3.42.1-1.fc36.x86_64
zfs-fuse-0.7.2-2-21.fc36.x86_64

```

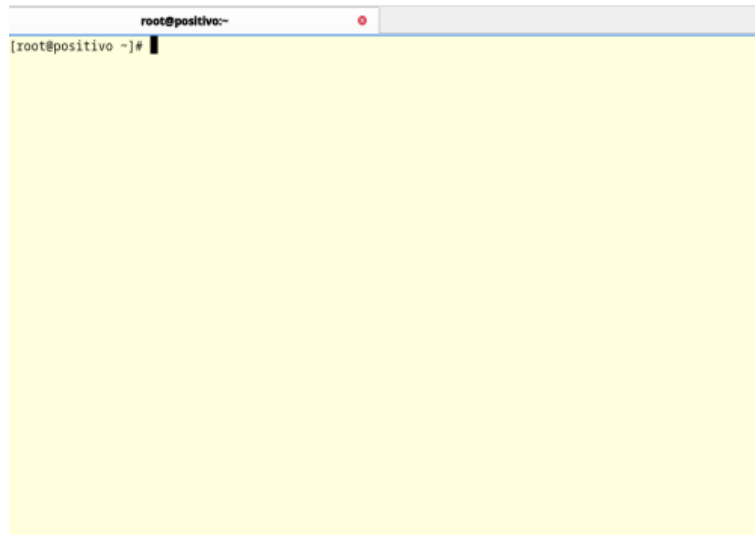
4) Por último, escribimos el comando **systemctl set-default graphical.target** para que se establezca que la interfaz gráfica inicie sola y no tener siempre que iniciarla por nosotros mismos.



# Instalación de Docker:

A continuación detallamos la instalación de docker en el sistema operativo:

1) Iniciamos como usuario root en el sistema (es necesario para instalar los archivos de docker



2) A continuación ingresamos el comando **dnf -y install docker** como superusuario para poder comenzar a descargar los archivos para poder instalar docker en nuestro sistema (si pasa cualquier tipo de error es este proceso es fundamental tener actualizado con el comando **dnf -y update** antes):



En este caso ya se encuentra instalado pero, tiene que aparecer un proceso de descarga con una continua instalación que no tarda dependiendo de la conexión a internet.

```
root@positivo:~  
[root@positivo ~]# dnf -y install docker  
Última comprobación de caducidad de metadatos hecha hace 0:48:04, el jue 16 jun 2022 13:16:53.  
El paquete moby-engine-20.10.16-1.fc36.x86_64 ya está instalado.  
Dependencias resueltas.  
Nada por hacer.  
¡Listo!  
[root@positivo ~]#
```

3) Una vez que tengamos instalado docker en nuestro sistema, verificamos si el proceso de docker está activo con el siguiente comando:

### **systemctl status docker**

```
root@positivo:~  
[root@positivo ~]# systemctl status docker  
● docker.service - Docker Application Container Engine  
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)  
   Active: inactive (dead) since Thu 2022-06-16 14:02:31 -03; 24s ago  
TriggeredBy: ● docker.socket  
   Docs: https://docs.docker.com  
  Main PID: 1108 (code=exited, status=0/SUCCESS)  
   CPU: 16.950s  
  
jun 16 13:53:47 positivo dockerd[1146]: time="2022-06-16T13:53:47.782640838-03:00" level=warning ms  
jun 16 14:02:30 positivo systemd[1]: Stopping docker.service - Docker Application Container Engine.  
jun 16 14:02:30 positivo dockerd[1108]: time="2022-06-16T14:02:30.832910501-03:00" level=info msg=">  
jun 16 14:02:30 positivo dockerd[1108]: time="2022-06-16T14:02:30.901715318-03:00" level=info msg=">  
jun 16 14:02:30 positivo dockerd[1108]: time="2022-06-16T14:02:30.902859698-03:00" level=info msg=">  
jun 16 14:02:30 positivo dockerd[1108]: time="2022-06-16T14:02:30.902992913-03:00" level=info msg=">  
jun 16 14:02:30 positivo dockerd[1108]: time="2022-06-16T14:02:30.903684048-03:00" level=info msg=">  
jun 16 14:02:31 positivo systemd[1]: docker.service: Deactivated successfully.  
jun 16 14:02:31 positivo systemd[1]: Stopped docker.service - Docker Application Container Engine.  
jun 16 14:02:31 positivo systemd[1]: docker.service: Consumed 16.950s CPU time.  
líneas 1-18/18 (END)
```

En mi caso el proceso de docker no esta activo, para activarlo, debemos escribir los siguiente en la terminal:

## systemctl start docker

```
root@positivo:~  
[root@positivo ~]# systemctl start docker  
[root@positivo ~]# systemctl status docker  
● docker.service - Docker Application Container Engine  
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)  
   Active: active (running) since Thu 2022-06-16 12:40:36 -03; 1h 19min ago  
TriggeredBy: ● docker.socket  
   Docs: https://docs.docker.com  
  Main PID: 1108 (dockerd)  
    Tasks: 25 (limit: 2118)  
  Memory: 98.1M  
   CPU: 16.455s  
   CGroup: /system.slice/docker.service  
           └─ 1108 /usr/bin/dockerd --host=fd:// --exec-opt native.cgroupdriver=systemd  
             └─ 1146 containerd --config /var/run/docker/containerd/containerd.toml  
  
jun 16 13:33:37 positivo 6bc08cadbd7f[1108]: 2022-06-16T16:33:37.822856Z 0 [Warning] >  
jun 16 13:33:37 positivo 6bc08cadbd7f[1108]: 2022-06-16T16:33:37.926935Z 0 [System] >  
jun 16 13:33:37 positivo 6bc08cadbd7f[1108]: 2022-06-16T16:33:37.927167Z 0 [System] >  
jun 16 13:53:45 positivo 6bc08cadbd7f[1108]: 2022-06-16T16:53:45.456317Z 0 [System] >  
jun 16 13:53:47 positivo 6bc08cadbd7f[1108]: 2022-06-16T16:53:47.279658Z 0 [System] >  
jun 16 13:53:47 positivo dockerd[1108]: time="2022-06-16T13:53:47.672937031-03:00" l>  
jun 16 13:53:47 positivo dockerd[1146]: time="2022-06-16T13:53:47.718290889-03:00" l>  
lines 1-20...skipping...  
● docker.service - Docker Application Container Engine  
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)  
   Active: active (running) since Thu 2022-06-16 12:40:36 -03; 1h 19min ago
```

Esto logra que el proceso de docker se inicialice pero no sirve si nosotros queremos que este proceso se inicie por si solo en el sistema, para eso necesitamos otro comando:

## systemctl enable docker

```
root@positivo:~  
[root@positivo ~]# systemctl enable docker  
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service - /usr/lib  
/systemd/system/docker.service.  
[root@positivo ~]#
```

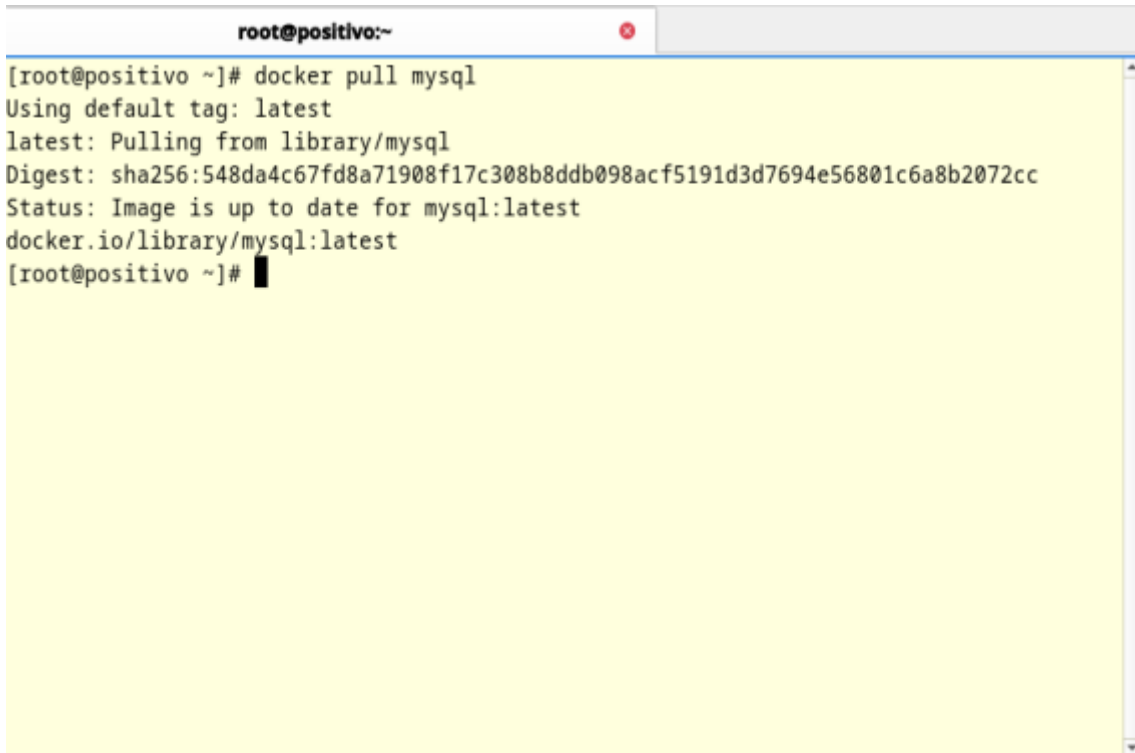
Si quieres desactivarlo escribe lo siguiente: **systemctl disable docker**



## Instalación de mysql en Docker:

Lo primero que haremos en la terminal es instalar mysql desde docker con el siguiente comando:

**docker pull mysql** (después de este punto comienza la descarga y la instalación)



```
root@positivo:~  
[root@positivo ~]# docker pull mysql  
Using default tag: latest  
latest: Pulling from library/mysql  
Digest: sha256:548da4c67fd8a71908f17c308b8ddb098acf5191d3d7694e56801c6a8b2072cc  
Status: Image is up to date for mysql:latest  
docker.io/library/mysql:latest  
[root@positivo ~]#
```

Antes de ejecutar el docker de mysql, tenemos que crear el volumen para que podamos trabajar de manera segura para que nuestros datos sean persistentes, para ello utilizamos el comando **docker volume create** “nombre” de la siguiente manera:

Para ver la lista de volúmenes escribimos el comando **docker volume list** en línea de comandos:



```
root@positivo:~  
[root@positivo ~]# docker volume list  
DRIVER      VOLUME NAME  
local       prueba  
[root@positivo ~]#
```

Ahora ya que tenemos creado nuestro volumen para no perder nuestros datos procedemos a ejecutarlo con varios modificadores, el comando que utilizamos es en siguiente:

```
docker run -d -p 33060:3306 --name "nombre_contenedor" -e MYSQL_ROOT_PASSWORD=1234 --mount src="nombre_volumen",dst=/var/lib/mysql mysql
```

```
root@positivo:~  
[root@positivo ~]# docker run -d -p 33060:3306 --name prueba_mysql -e MYSQL_ROOT_PASSWORD=1234 --mount src=prueba,dst=/var/lib/mysql mysql  
6bc08cadbd7f5a401a03e1f656d9f8e388af38c2cd7413546be0a0b5304ed7da
```

Una vez que ingresamos el comando para lanzar el contenedor y nos haya devuelto un hash, lo siguiente que haremos es acceder al contenedor que ejecutamos con el siguiente comando:

```
docker exec -it "nombre_contenedor" mysql -p
```

```
[root@positivo ~]# docker exec -it prueba_mysql mysql -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 9  
Server version: 8.0.29 MySQL Community Server - GPL  
  
Copyright (c) 2000, 2022, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> exit
```

Ahí logramos acceder al contenedor exitosamente, para salir solo escribimos el comando exit

La instalación de docker y ejecución del contenedor mysql esta completa.

## Instalación del servidor LAMP:

El acrónimo LAMP se utiliza para describir un sistema de infraestructura de internet que utiliza las siguientes herramientas:

**Linux:** el sistema operativo, en este caso usaremos la distribución Fedora 36.

**Apache:** el servidor web.

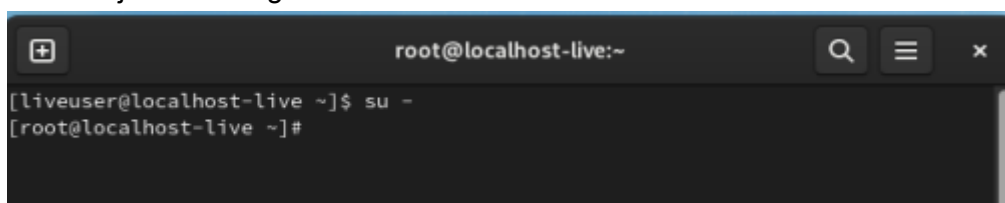
**MySQL o MariaDB:** el gestor de base de datos, utilizaremos MariaDB.

**PHP:** el lenguaje de programación.

Una vez sabiendo esto realizaremos la instalación de nuestro ambiente de desarrollo LAMP, se va a dividir en 6 pasos y estos son:

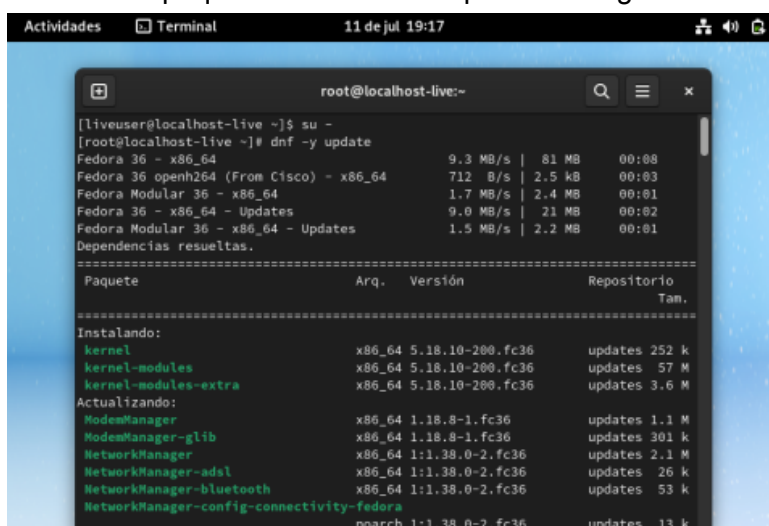
- 1.Actualización de paquetes del sistema operativo.
- 2.Instalación del sistema Apache.
- 3.Instalación del sistema PHP.
- 4.Instalación del sistema motor de base de datos MariaDB
- 5.Instalación de dependencias Apache para el lenguaje de programación PHP y el motor de base de datos MariaDB.
- 6.Activación del sistema LAMP.

1)Para la actualización de los paquetes del sistema operativo general se realiza a través del sistema DNF en el cual se hacen todas las versiones de producción y en Fedora, para esto debemos ejecutar la siguiente instrucción como usuario root utilizando el comando **su -**



```
root@localhost-live:~  
[liveuser@localhost-live ~]$ su -  
[root@localhost-live ~]#
```

Luego de acceder como usuario root copiaremos el comando **dnf -y update**, el cual realizará la actualización de paquetes del sistema operativo en general.

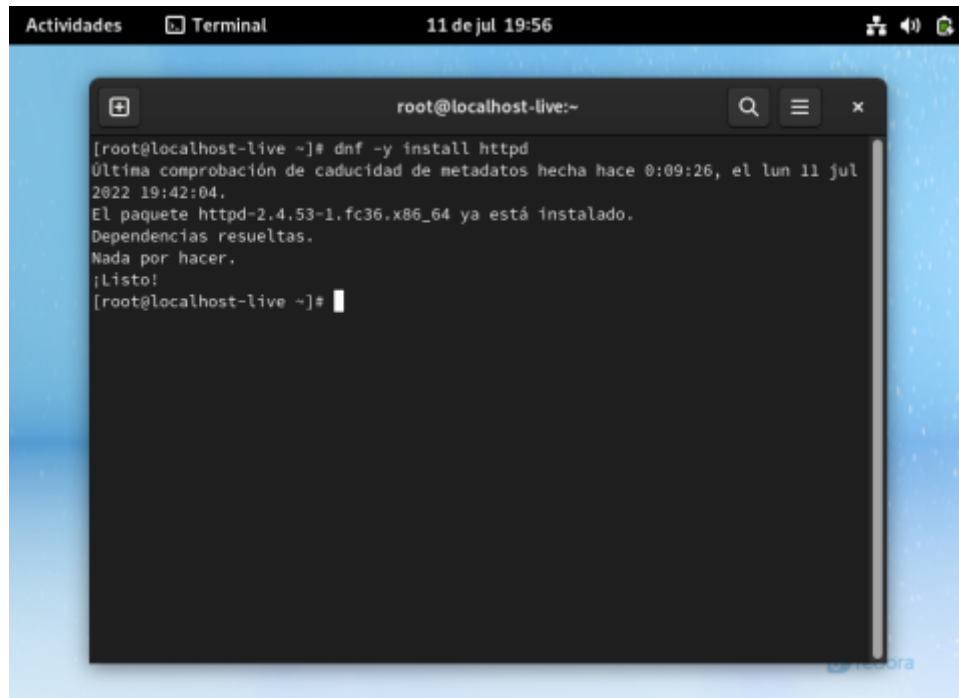


```
Actividades Terminal 11 de jul 19:17  
root@localhost-live:~  
[liveuser@localhost-live ~]$ su -  
[root@localhost-live ~]# dnf -y update  
Fedora 36 - x86_64 9.3 MB/s | 81 MB 00:08  
Fedora 36 openh264 (From Cisco) - x86_64 712 B/s | 2.5 kB 00:03  
Fedora Modular 36 - x86_64 1.7 MB/s | 2.4 MB 00:01  
Fedora 36 - x86_64 - Updates 9.0 MB/s | 21 MB 00:02  
Fedora Modular 36 - x86_64 - Updates 1.5 MB/s | 2.2 MB 00:01  
Dependencias resueltas.  
=====
```

Paquete	Arq.	Versión	Repositorio	Tam.
Instalando:				
kernel	x86_64	5.18.10-200.fc36	updates	252 k
kernel-modules	x86_64	5.18.10-200.fc36	updates	57 M
kernel-modules-extra	x86_64	5.18.10-200.fc36	updates	3.6 M
Actualizando:				
ModemManager	x86_64	1.18.8-1.fc36	updates	1.1 M
ModemManager-glib	x86_64	1.18.8-1.fc36	updates	301 k
NetworkManager	x86_64	1:1.38.0-2.fc36	updates	2.1 M
NetworkManager-adsl	x86_64	1:1.38.0-2.fc36	updates	26 k
NetworkManager-bluetooth	x86_64	1:1.38.0-2.fc36	updates	53 k
NetworkManager-config-connectivity-fedora	x86_64	1:1.38.0-2.fc36	updates	13 k
noarch	1:1.38.0-2.fc36		updates	13 k

Una vez ejecutado este comando nuestro sistema operativo quedo actualizado con las versiones estables.

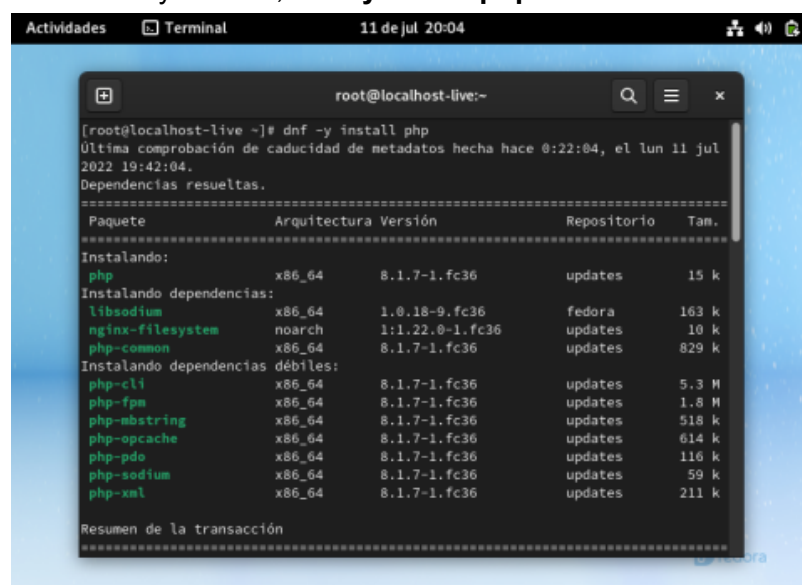
2)Ahora vamos a proceder a hacer la instalación del sistema Apache el cual se va instalar a través del siguiente comando, **dnf -y install httpd**.



```
root@localhost-live:~  
[root@localhost-live ~]# dnf -y install httpd  
Última comprobación de caducidad de metadatos hecha hace 0:09:26, el lun 11 jul 2022 19:42:04.  
El paquete httpd-2.4.53-1.fc36.x86_64 ya está instalado.  
Dependencias resueltas.  
Nada por hacer.  
¡Listo!  
[root@localhost-live ~]#
```

En la imagen se puede apreciar que en esta distribución Fedora ya se encuentra instalado el sistema Apache con sus dependencias pero, en el caso de que no lo esté solo debemos esperar unos minutos y ya quedará hecha la descarga.

3)Ahora continuaremos con la instalación del sistema PHP, de igual manera se realizará mediante un comando dnf y este es, **dnf -y install php**.



```
root@localhost-live:~  
[root@localhost-live ~]# dnf -y install php  
Última comprobación de caducidad de metadatos hecha hace 0:22:04, el lun 11 jul 2022 19:42:04.  
Dependencias resueltas.  
=====
```

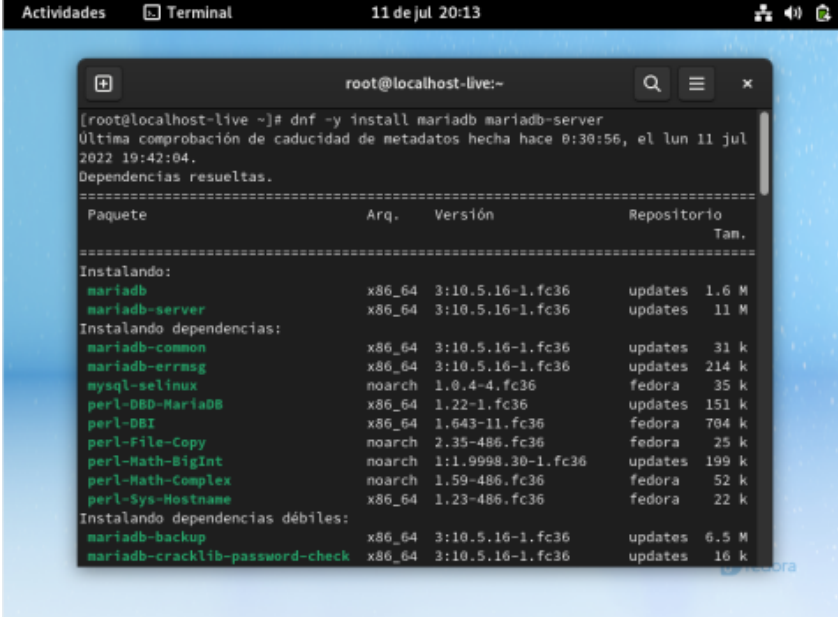
Paquete	Arquitectura	Versión	Repositorio	Tam.
Instalando:				
php	x86_64	8.1.7-1.fc36	updates	15 k
Instalando dependencias:				
libsodium	x86_64	1.0.18-9.fc36	fedora	163 k
nginxfilesystem	noarch	1:1.22.0-1.fc36	updates	10 k
php-common	x86_64	8.1.7-1.fc36	updates	829 k
Instalando dependencias débiles:				
php-cli	x86_64	8.1.7-1.fc36	updates	5.3 M
php-fpm	x86_64	8.1.7-1.fc36	updates	1.8 M
php-mbstring	x86_64	8.1.7-1.fc36	updates	518 k
php-opcache	x86_64	8.1.7-1.fc36	updates	614 k
php-pdo	x86_64	8.1.7-1.fc36	updates	116 k
php-sodium	x86_64	8.1.7-1.fc36	updates	59 k
php-xml	x86_64	8.1.7-1.fc36	updates	211 k

```
=====
```

Resumen de la transacción

```
=====
```

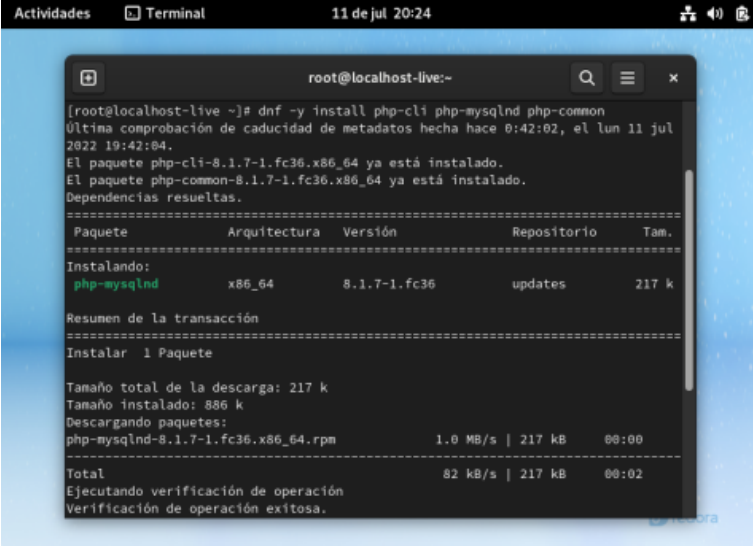
4) Luego de que se haya terminado la descarga e instalación de los paquetes continuaremos con la descarga del gestor de base de datos MariaDB, el comando que pondremos en la terminal será **dnf -y install mariadb mariadb-server**.



```
root@localhost-live:~# dnf -y install mariadb mariadb-server
Última comprobación de caducidad de metadatos hecha hace 0:30:56, el lun 11 jul 2022 19:42:04.
Dependencias resueltas.
=====
Paquete                Arq.    Versión          Repositorio      Tam.
=====
Instalando:
mariadb                x86_64  3:10.5.16-1.fc36 updates          1.6 M
mariadb-server         x86_64  3:10.5.16-1.fc36 updates          11 M
Instalando dependencias:
mariadb-common         x86_64  3:10.5.16-1.fc36 updates          31 k
mariadb-errlog         x86_64  3:10.5.16-1.fc36 updates          214 k
mysql-selinux          noarch  1.0.4-4.fc36    fedora           35 k
perl-DBD-MariaDB       x86_64  1.22-1.fc36     updates          151 k
perl-DBI               x86_64  1.643-11.fc36   fedora           704 k
perl-File-Copy         noarch  2.35-486.fc36   fedora           25 k
perl-Math-BigInt       noarch  1:1.9998.30-1.fc36 updates          199 k
perl-Math-Complex      noarch  1.59-486.fc36   fedora           52 k
perl-Sys-Hostname      x86_64  1.23-486.fc36   fedora           22 k
Instalando dependencias débiles:
mariadb-backup         x86_64  3:10.5.16-1.fc36 updates          6.5 M
mariadb-cracklib-password-check x86_64  3:10.5.16-1.fc36 updates          16 k
```

5) Ahora realizaremos la instalación de las dependencias de Apache para la integración con el lenguaje de programación PHP y el motor de base de datos MariaDB. Para ello usaremos el comando:

**dnf -y install php-cli php-mysqlnd php-common.**



```
root@localhost-live:~# dnf -y install php-cli php-mysqlnd php-common
Última comprobación de caducidad de metadatos hecha hace 0:42:02, el lun 11 jul 2022 19:42:04.
El paquete php-cli-8.1.7-1.fc36.x86_64 ya está instalado.
El paquete php-common-8.1.7-1.fc36.x86_64 ya está instalado.
Dependencias resueltas.
=====
Paquete                Arquitectura  Versión          Repositorio      Tam.
=====
Instalando:
php-mysqlnd            x86_64        8.1.7-1.fc36     updates          217 k
Resumen de la transacción
=====
Instalar 1 Paquete

Tamaño total de la descarga: 217 k
Tamaño instalado: 886 k
Descargando paquetes:
php-mysqlnd-8.1.7-1.fc36.x86_64.rpm          1.0 MB/s | 217 kB  00:00
Total                                         82 kB/s | 217 kB  00:02
Ejecutando verificación de operación
Verificación de operación exitosa.
```

6) Ya tenemos instalados los sistemas Apache, PHP y MariaDB ahora procederemos con la activación del servidor LAMP en general, para realizarlo usaremos el comando **systemctl**.

a. Primero habilitaremos el servicio Apache con el comando **systemctl enable httpd**.

b.Luego habilitaremos el motor de base de datos MariaDB con el comando **systemctl enable mariadb**.

```
[root@localhost-live ~]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@localhost-live ~]# systemctl enable mariadb
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
[root@localhost-live ~]#
```

Ya tenemos los servicios habilitados, esto lo que nos permitirá es que cuando se inicie el sistema operativo automáticamente se inicien esos servicios. Sin embargo todavía no están activados, para ello debemos inicializarlos con los siguientes comandos:

**-systemctl start httpd**  
**-systemctl start mariadb**

```
[root@localhost-live ~]# systemctl start httpd
[root@localhost-live ~]# systemctl start mariadb
[root@localhost-live ~]#
```

Una vez inicializados los servicios de Apache y MariaDB ahora debemos generar una configuración en el motor de base de datos con el comando **mysql\_secure\_installation**

```
[root@localhost-live ~]# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] Y
Enabled successfully!
Reloading privilege tables..
... Success!

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

En este proceso nos pedirá que creamos una contraseña para el usuario root y adicional a esto por defecto nos deshabilita los usuarios anónimos y los logueos remotos.

```

Remove anonymous users? [Y/n] Y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

```

Luego de ello se removerán las bases de datos de datos y se recargan los permisos y con esto ya terminamos con la configuración de MariaDB.

```

Remove test database and access to it? [Y/n] Y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] Y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
[root@localhost-live ~]#

```

A continuación vamos a reinicializar el servicio de MariaDB con el comando **systemctl restart mariadb**.

Una vez hecho esto ahora verificaremos que los servicios de Apache y MariaDB estén activos e inicializados, para ello utilizaremos nuevamente el comando **systemctl** pero ahora consultando el estado de cada uno.

**-systemctl status httpd**

```

[root@localhost-live ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
            └─ghp-fpm.conf
   Active: active (running) since Mon 2022-07-11 20:53:00 -03; 38min ago
     Docs: man:httpd.service(8)
   Main PID: 37638 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:  0 B/sec"
     Tasks: 177 (limit: 6855)
    Memory: 18.5M
       CPU: 1.522s
   CGroup: /system.slice/httpd.service
           └─37638 /usr/sbin/httpd -DFOREGROUND
             └─37638 /usr/sbin/httpd -DFOREGROUND
               └─37640 /usr/sbin/httpd -DFOREGROUND
                 └─37641 /usr/sbin/httpd -DFOREGROUND
                   └─37642 /usr/sbin/httpd -DFOREGROUND

```

Como se ve se encuentra activo y corriendo.

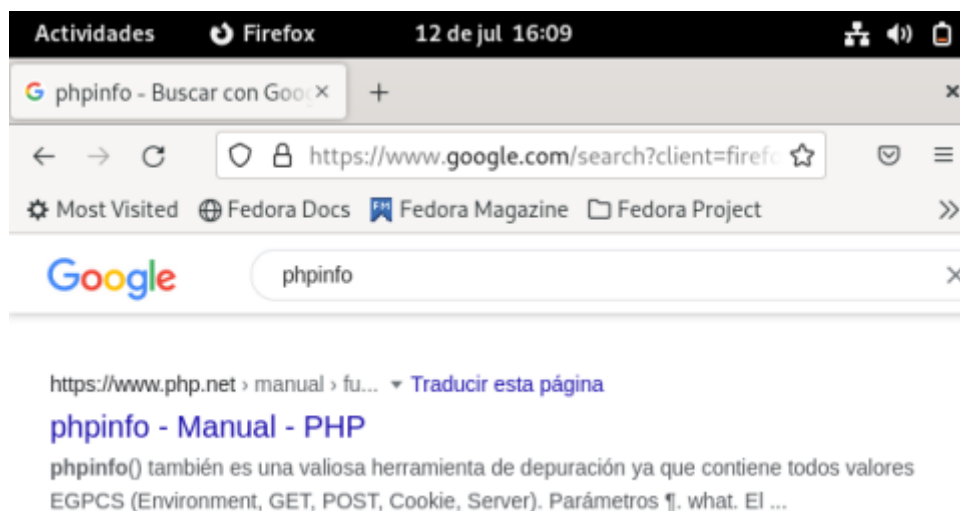
## -systemctl status mariadb

```
mariadb.service - MariaDB 10.5 database server
Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
Active: active (running) since Tue 2022-07-12 15:41:41 -03; 4min 58s ago
Docs: man:mariadb(8)
      https://mariadb.com/kb/en/library/systemd/
Process: 778 ExecStartPre=/usr/libexec/mariadb-check-socket (code=exited, status=0/SUCCESS)
Process: 818 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir mariadb.service (code=exited, status=0/SUCCESS)
Process: 959 ExecStartPost=/usr/libexec/mariadb-check-upgrade (code=exited, status=0/SUCCESS)
Main PID: 878 (mariadb)
Status: "Taking your SQL requests now..."
Tasks: 8 (limit: 6896)
Memory: 96.0M
CPU: 64ms
CGroup: /system.slice/mariadb.service
        └─ 878 /usr/libexec/mariadb --basedir=/usr
```

El sistema gestor de base de datos MariaDB también está corriendo sin problemas.

Ahora para verificar que nuestro sistema cuenta con Apache y PHP precisamos la creación de un archivo que se llama **phpinfo**, esto lo podemos consultar en el manual de la página de php. Para buscar el manual hacemos lo siguiente:

-Primero abrimos nuestro navegador y buscaremos “phpinfo” e ingresamos en donde dice manual.



-Una vez que estemos en la página bajaremos hasta donde nos salga el siguiente texto:



Esto lo vamos a copiar en el archivo que se encuentra en la ruta donde está nuestro sistema Apache. Debemos poner en la terminal lo siguiente: **vim /var/www/html/phpinfo.php**



The image shows a terminal window and a web browser. The terminal window displays the vim editor creating the file `/var/www/html/phpinfo.php`. The web browser shows the output of the `phpinfo()` function, displaying PHP version 8.1.7 and various system information.

**Ejemplo #1 Ejemplo de phpinfo()**

```
<?php

// Muestra toda la información, por defecto INFO_ALL
phpinfo();

// Muestra solamente la información de los módulos.
// phpinfo(8) hace exactamente lo mismo.
phpinfo(INFO_MODULES);

?>
```

PHP Version 8.1.7

System	Linux localhost-live 5.18.10-200.fc36.x86_64 #1 SMP PREEMPT_DYNAMIC Thu Jul 7 17:21:38 UTC 2022 x86_64
Build Date	Jun 7 2022 18:21:38
Build System	Fedora release 36 (Thirty Six)
Build Provider	Fedora Project
Compiler	gcc (GCC) 12.1.1 20220507 (Red Hat 12.1.1-1)
Architecture	x86_64
Server API	PHP/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/10-opcache.ini, /etc/php.d/20-bc2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-erf.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-ibstring.ini, /etc/php.d/20-mbstring.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-simplexml.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sodium.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-xml.ini, /etc/php.d/20-xmlwriter.ini, /etc/php.d/20-xsl.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini, /etc/php.d/30-xmlreader.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902.NTS

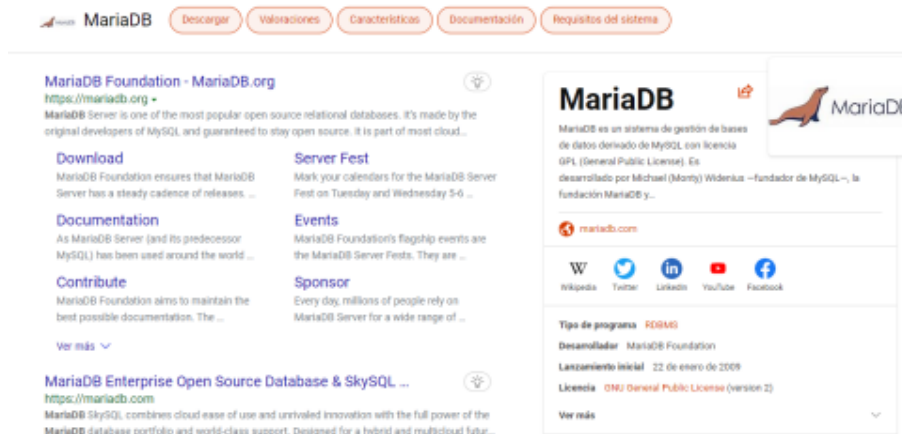
Una vez que lo hayamos copiado en el editor de texto vim se creará el archivo, ahora realizaremos una consulta a través de nuestro navegador web con el nombre del archivo.

Y con esto ya tenemos todo nuestro sistema LAMP instalado y configurado.

# Instalación de MySQL/MariaDB:

En esta instancia vamos a realizar la instalación del motor de bases de datos MariaDB, lo vamos a realizar en el sistema operativo Windows 10 pro ya que va a ser nuestro sistema que usaremos para los equipos.

1. Primero buscaremos en nuestro navegador web “mariadb” y vamos a ingresar en el primer enlace que aparece que es “MariaDB Foundation – MariaDB.org”.



2. Luego de que estemos en la página le vamos a dar clic en donde dice “download”.



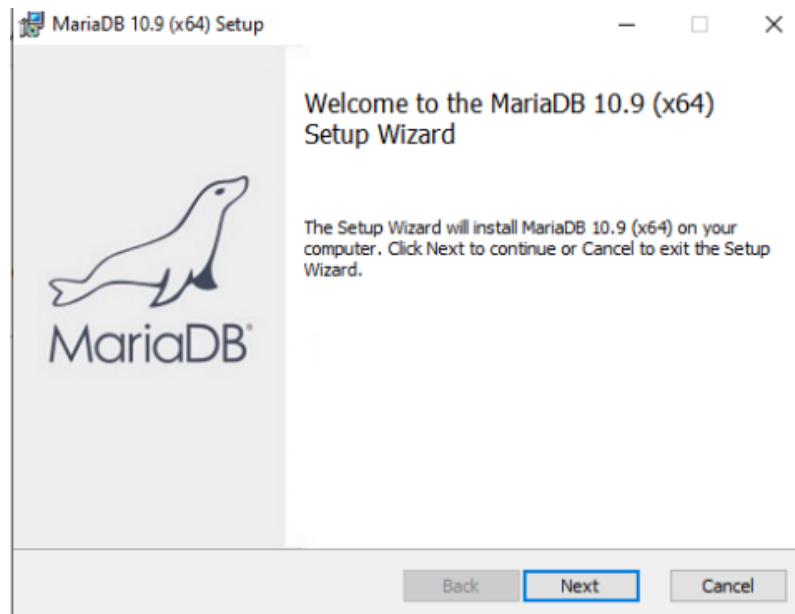
3. Ahora nos mandara a un enlace en donde sale lo siguiente:

Primero nos aparece la versión del servidor, nosotros escogeremos la más estable que es la 10.9.1 RC, abajo nos sale el sistema operativo que utilizaremos en este caso será Windows, luego sale la arquitectura que usaremos que es la de 64 bits y por último, nos sale el tipo de empaquetado de cómo queremos instalarlo, en este lo dejaremos así para que al darle doble clic se descargue de una vez.

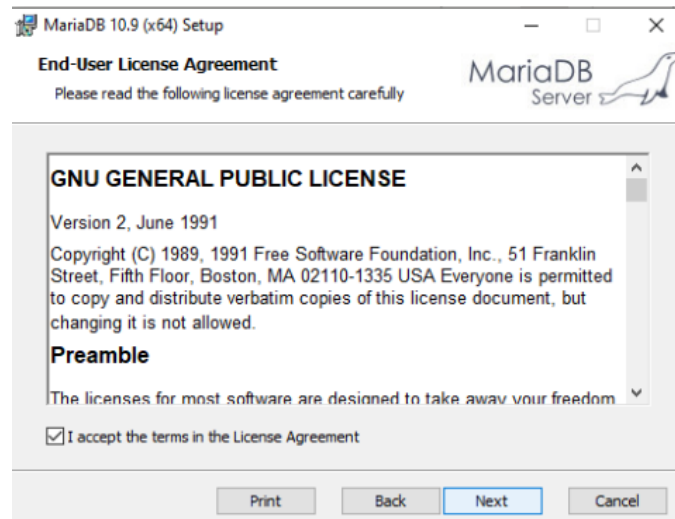
Sabiendo esto ahora le daremos a “Download” para que se descargue en nuestro equipo.



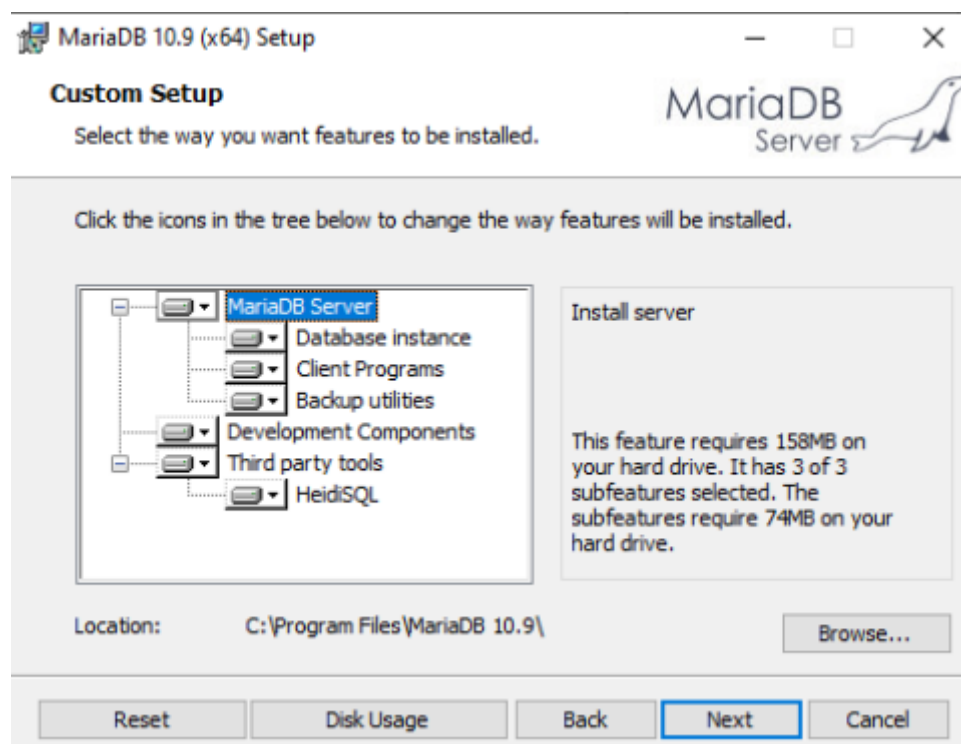
Una vez que se haya descargado le vamos a dar clic en el archivo para empezar la instalación.



4. Le daremos en “next”.



5. Vamos a aceptar los términos y le daremos nuevamente en “next”.



6. Aquí se muestra que nos va a instalar una herramienta que se llama “HeidiSQL”, esto nos va a permitir conectarnos y ver todos nuestros objetos de nuestra base de datos de forma gráfica. Sabiendo esto ahora continuaremos dándole a “next”.

**User settings**

Default instance properties  
MariaDB 10.9 (x64) database configuration

☒ **Modify password for database user 'root'**

New root password:  Enter new root password

Confirm:  Retype the password

☐ **Enable access from remote machines for 'root' user**

☒ **Use UTF8 as default server's character set**

**Data directory**

7. Aquí nos pedirá que creamos una contraseña, abajo nos pregunta si queremos habilitar el acceso de otras computadoras con el usuario root, esto en el caso de que se instale en un servidor y se acceda de forma remota, lo vamos a dejar deshabilitado por cuestiones de seguridad. Y por último, usaremos los caracteres UTF8 por defecto y ahora le damos de nuevo a "next".

**Database settings**

Default instance properties  
MariaDB 10.9 (x64) database configuration

☒ **Install as service**

Service Name:

☒ **Enable networking**

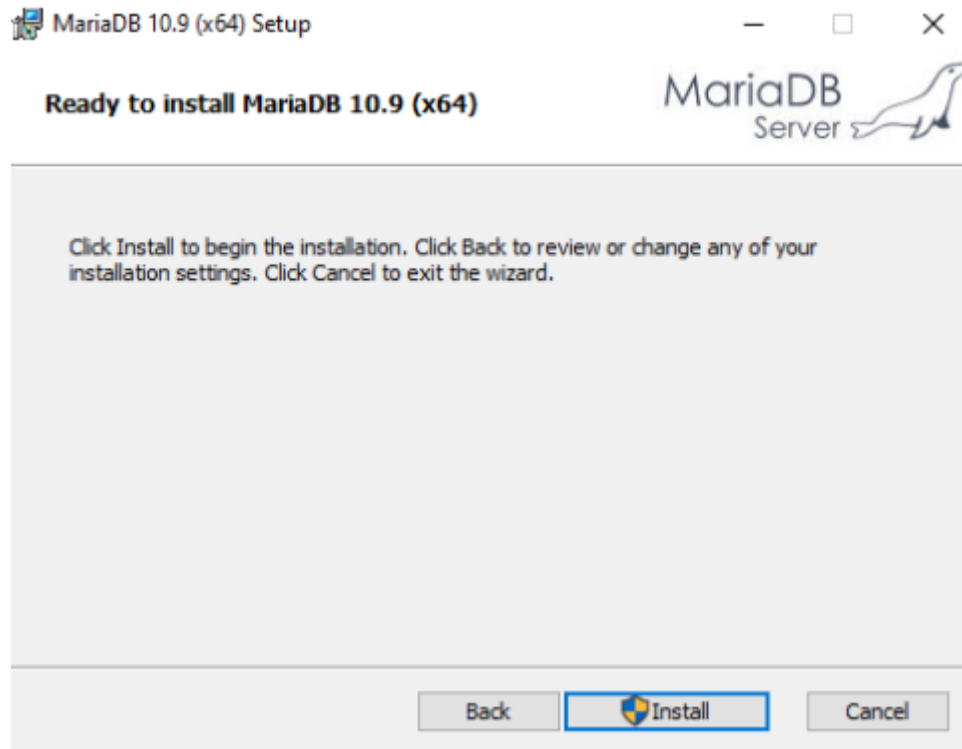
TCP port:

**InnoDB engine settings**

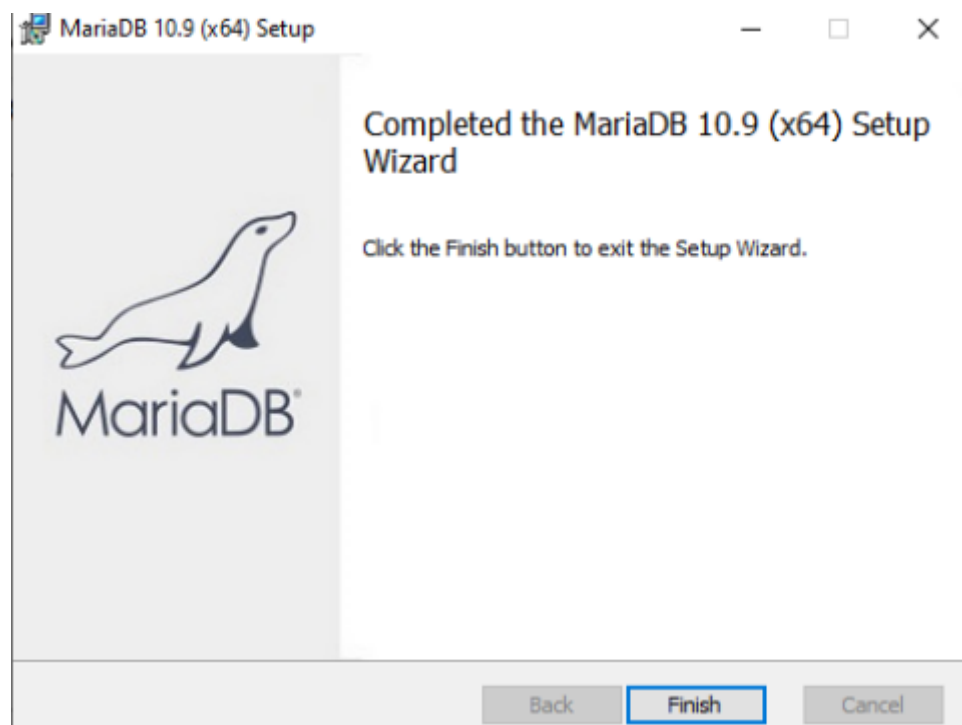
Buffer pool size:  MB

Page size:  KB

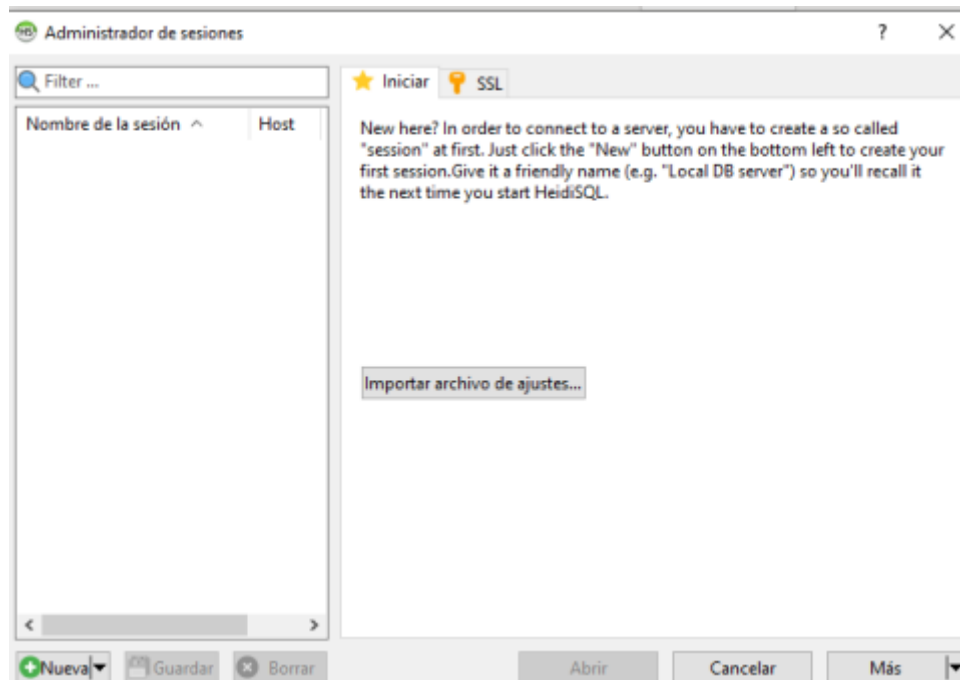
8. Arriba sale el nombre que va a tener cada vez que iniciemos el programa. Abajo nos sale el número del puerto que vamos a utilizar, por defecto siempre sale el puerto 3306 pero lo tenemos que cambiar, en este caso colocamos el puerto 3310. Terminamos de configurar y le damos a “next”.



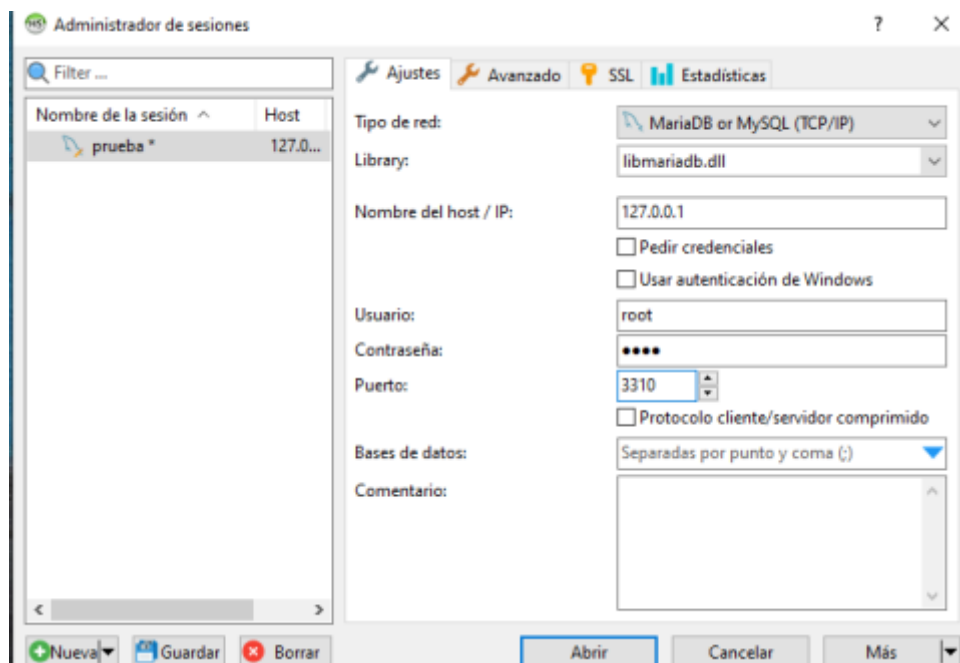
9. Le daremos a instalar y esperamos unos segundos.



10. Luego de que se haya instalado le damos en “Finish” y ya quedará listo todo. Ahora para comprobar de que se haya descargado e instalado el programa correctamente abriremos el HeidiSQL.

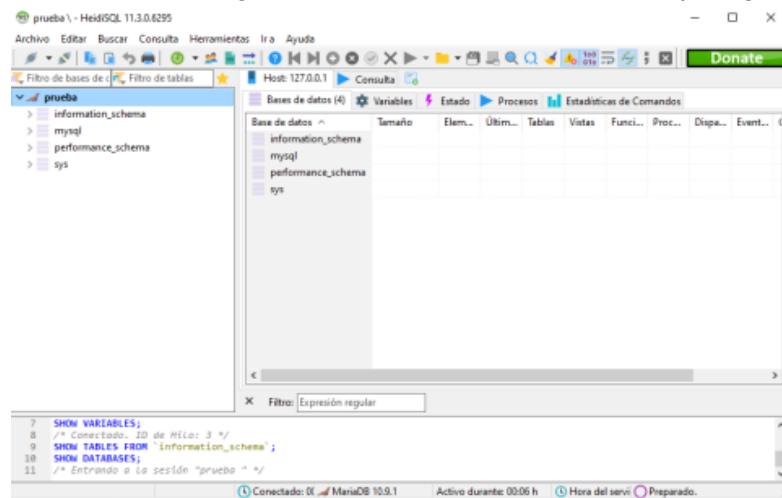


11. Le vamos a dar clic en “Nueva” para crear una conexión.



12. Aquí nos pedirá que pongamos un nombre a la conexión, en nuestro caso le pusimos “prueba”, en la parte derecha nos sale que tipo de red queremos usar, la librería, la IP de

nuestro host, el usuario que es el root con su contraseña y el puerto donde lo vamos a crear que es el 3310. Después de configurar todo le damos en “Guardar” y luego en “Abrir”.



13. Y listo, se realizó correctamente la instalación y ya se encuentra corriendo el MariaDB.



## Instalación de ssh en Fedora Server (servidor)

- A continuación dejamos los pasos para poder instalar y utilizar el protocolo ssh en linux, en este caso usamos Fedora 36 de acuerdo al sistema operativo utilizado por el servidor.
- Como primer paso vamos a instalar openssh en linux, generalmente viene instalado en fedora server pero podemos instalarlo con el comando **dnf -y install openssh** como usuario root

```
[root@192 ~]# dnf -y install openssh
Última comprobación de caducidad de metadatos hecha hace 1:05:55, el mar 06 set
2022 20:59:17.
El paquete openssh-8.8p1-1.fc36.1.x86_64 ya está instalado.
Dependencias resueltas.
Nada por hacer.
¡Listo!
[root@192 ~]#
```

- Lo siguiente será verificar que el proceso de ssh esté activo, para ver el estado del servicio utilizamos el comando **systemctl status sshd** :

```
[root@192 ~]# systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: ena
   Active: active (running) since Tue 2022-09-06 21:43:09 -03; 1h 4min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 2786 (sshd)
     Tasks: 1 (limit: 2320)
    Memory: 3.6M
       CPU: 133ms
    CGroup: /system.slice/sshd.service
            └─ 2786 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

set 06 21:43:09 192.168.1.3 systemd[1]: Starting sshd.service - OpenSSH server daemon:
set 06 21:43:09 192.168.1.3 sshd[2786]: Server listening on 0.0.0.0 port 22.
set 06 21:43:09 192.168.1.3 sshd[2786]: Server listening on :: port 22.
set 06 21:43:09 192.168.1.3 systemd[1]: Started sshd.service - OpenSSH server daemon:
set 06 21:43:47 192.168.1.3 sshd[2791]: Accepted password for user from 192.168.1.3:
set 06 21:43:47 192.168.1.3 sshd[2791]: pam_unix(sshd:session): session opened for
lines 1-18/18 (END)
```

- En mi caso el servicio está activo (por defecto en fedora server) pero ahora vamos a editarlo para que el servicio inicie por sí solo con el comando **systemctl enable sshd**:

```
[root@192 ~]# systemctl enable sshd
[root@192 ~]#
```

- Ahora tenemos que permitir el puerto 22 al servicio, para eso vamos a editar un archivo ubicado en **/etc/ssh/sshd\_config** y lo editamos con vim o vi dependiendo de que tengamos instalado:
- Una vez editando el archivo vamos a editar la línea de texto que dice **#port 22** y le quitamos el numeral delante, esto hace que deje de ser un comentario y que sea un comando para usar el puerto

```
# $OpenBSD: sshd_config,v 1.104 2021/07/02 05:11:21 dtucker Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

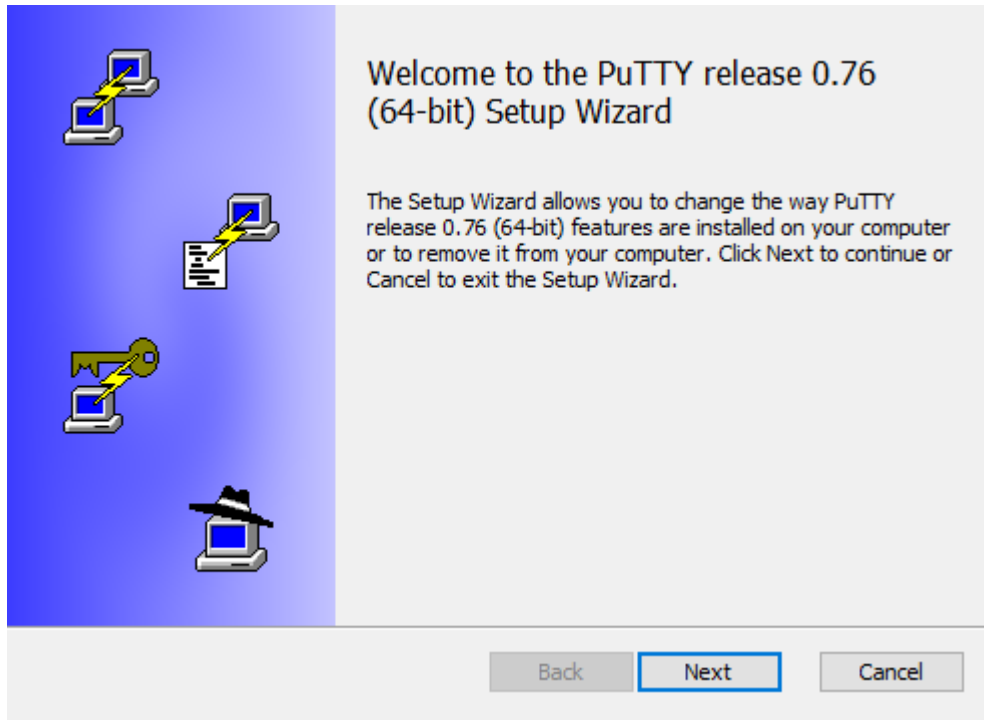
# To modify the system-wide sshd configuration, create a *.conf file under
# /etc/ssh/sshd_config.d/ which will be automatically included below
Include /etc/ssh/sshd_config.d/*.conf

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
-- INSERTAR --
```

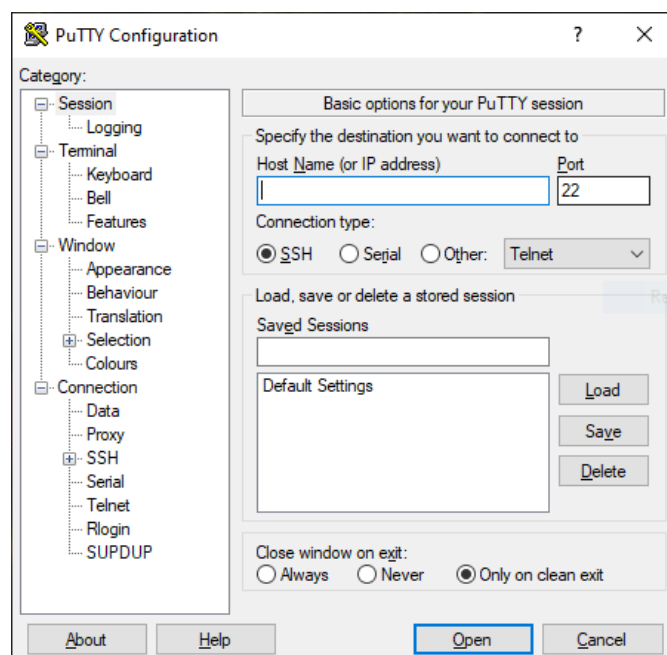
En esta parte ya tendremos el servicio ssh listo para usar.

## Instalación de cliente ssh en windows

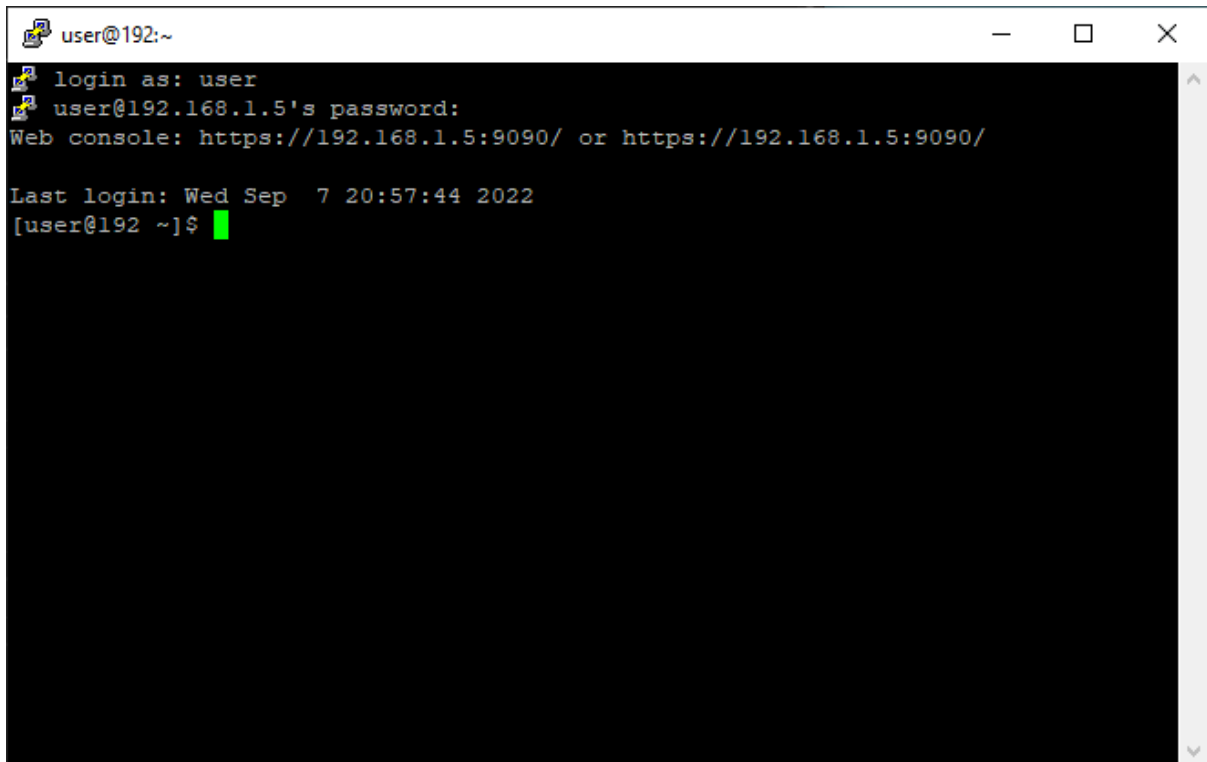
- En este apartado utilizaremos como ejemplo al cliente ssh putty para windows



- La instalación de putty se hace como cualquier otro instalador de windows, tenemos el parámetro para elegir donde instalar y listo, el programa requiere al menos 50 mb de espacio en disco duro.
- Una vez abierto el programa vemos la siguiente interfaz:



- En la parte donde dice host name (or ip address) ingresamos la ip asignada en el sistema operativo del servidor (en este caso fedora server), después de asignarla le damos al botón de open y ya accedemos a la terminal de linux si todo está correcto, al iniciar la terminal pide información de usuario del sistema linux por defecto usuario y contraseña:



```
user@192:~  
login as: user  
user@192.168.1.5's password:  
Web console: https://192.168.1.5:9090/ or https://192.168.1.5:9090/  
  
Last login: Wed Sep  7 20:57:44 2022  
[user@192 ~]$
```

Listo, ya podremos usar el protocolo ssh

- **Anexo:**

- **Estudio de los roles de usuarios del sistema:**

Usuario servidor y respaldos:

- **Informático:** El informático se encarga de darle mantenimiento y vigilancia al servidor, tiene acceso a los controles y monitoreo de hardware del servidor desde el sistema operativo.
- **administrador:** Es el que se encarga de los respaldos que hay en el sistema y cómo se efectúan, tiene acceso al log de eventos de respaldos.
- **Usuario root:** el usuario que administra y controla todo el sistema completo, es el rango más alto en lo que a permisos se refiere porque contiene todos los permisos del sistema

Usuario base de datos:

- **Informático:** En la base de datos el informático tiene permisos de ALTER y SELECT en algunos casos.
- **Administrador:** En la base de datos el administrador tiene los permisos de SELECT, INSERT, UPDATE y DELETE
- **Usuario root:** Contiene todos los permisos sobre la base de datos hecha

**Usuarios del sistema:** (son usuarios básicos y de la base de datos que componen al sistema y no contienen permisos de mayor grado)

- Cuerpo directivo
- Repartidor
- Cliente

- **Relevamiento y justificación de los sistemas operativos:**

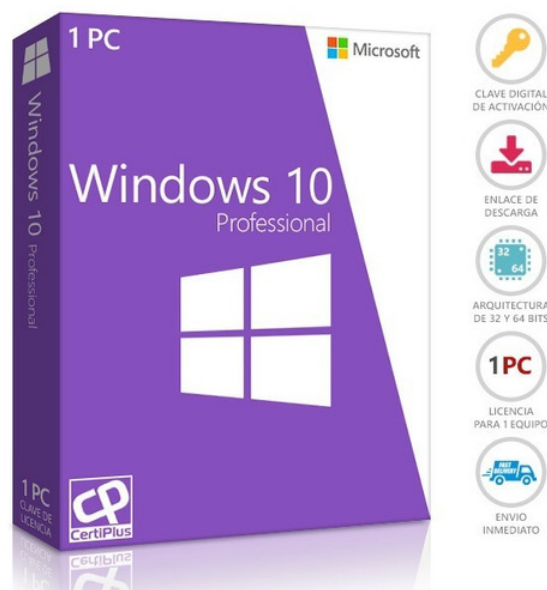
**Sistema operativo del equipo:**

Nuestro sistema operativo que elegimos para los equipos fue el Windows 10 pro, básicamente este es uno de los sistemas operativos para una empresa ya que nos

ofrece herramientas sofisticadas de conectividad y privacidad como unión de dominio, Internet Explorer modo empresarial, Bitlocker, administración de políticas de grupo, etc. Una de las ventajas que tiene el Windows 10 pro es que organiza actualizaciones a través de la nube, esto nos va a permitir actualizar varias computadoras portátiles y computadoras que estén en un dominio, todo esto se hace desde una computadora central y es facilita mucho el trabajo y ahorra tiempo.

### **Requisitos mínimos para usar Windows 10 pro:**

- El procesador debe ser de 2 GHz y tiene que ser compatible con NX, PAE y SSE2.
- Memoria RAM de 1 GB (32 bits) o de 2 GB (64 bits).
- Almacenamiento de 16 GB (32 bits) o más espacio como 20 GB (64 bits).
- Tarjeta gráfica requerida, tiene que ser compatible con Microsoft DirectX 9 o posterior, pero con un controlador WDDM.
- Tener una cuenta Microsoft y acceso a Internet al momento de la instalación.
- La pantalla debe tener una resolución de 800x600 o superior.



### **Sistema operativo del servidor:**

Fedora Server versión 36

Elegimos este sistema operativo para nuestro servidor porque propone una base adaptable, práctica y con acceso inmediato a servicios adecuados para empresas y organizaciones.

Es una distribución basada en Red Hat, su sistema de paquetes es rpm, destaca por ser confiable y tener la última versión lanzada de sus paquetes o aplicaciones.

Fedora Server es una distribución de Linux basada en software de código abierto y gratuito. La misión del Proyecto Fedora es liderar el avance del software y del contenido Libre y de código abierto como una comunidad colaborativa. Los tres elementos de esta misión están claros:

- El Proyecto Fedora se esfuerza siempre por liderar, no por seguir.
- El Proyecto Fedora, consecuentemente, busca crear, mejorar y difundir el código y el contenido Free/Libre.
- El Proyecto Fedora tiene éxito gracias a la acción compartida de muchas personas en nuestra comunidad.

Fedora proporciona software que se ajusta a una amplia variedad de aplicaciones. Los requisitos de almacenamiento, memoria y procesamiento varían dependiendo de la utilización. Por ejemplo, un servidor de base de datos de alto tráfico requiere muchas más memoria y almacenamiento que un sobremesa de negocio, que en cambio tiene unos requerimientos más altos que una máquina virtual de propósito sencillo.

### **Requisitos mínimos Fedora Server:**

Mínimo recomendado para una instalación predeterminada. Sus requerimientos pueden diferir y la mayoría de las aplicaciones se beneficiarán si se superan los recursos mínimos.

- Procesador de 1 GHz o más rápido.
- 4GB de memoria del sistema.
- 20 o 30 GB de almacenamiento de disco no asignado.

