

The logo of Ibero University, featuring the word "IBERO" in a large, white, serif font on a red rectangular background.

# IBERO

Ciudad de México • Tijuana ®

## ARQUITECTURA DE INFORMACIÓN EN WEB Y LABORATORIO.

### PRÁCTICA 1

AVILES ELIZALDE DIANA LAURA  
ESPITIA VILLAMIL GABRIELA VANESSA  
GARAY HERNANDEZ MAURICIO DE  
GARCÍA RAMOS BERNARDO

PRIMAVERA 2021

# ÍNDICE

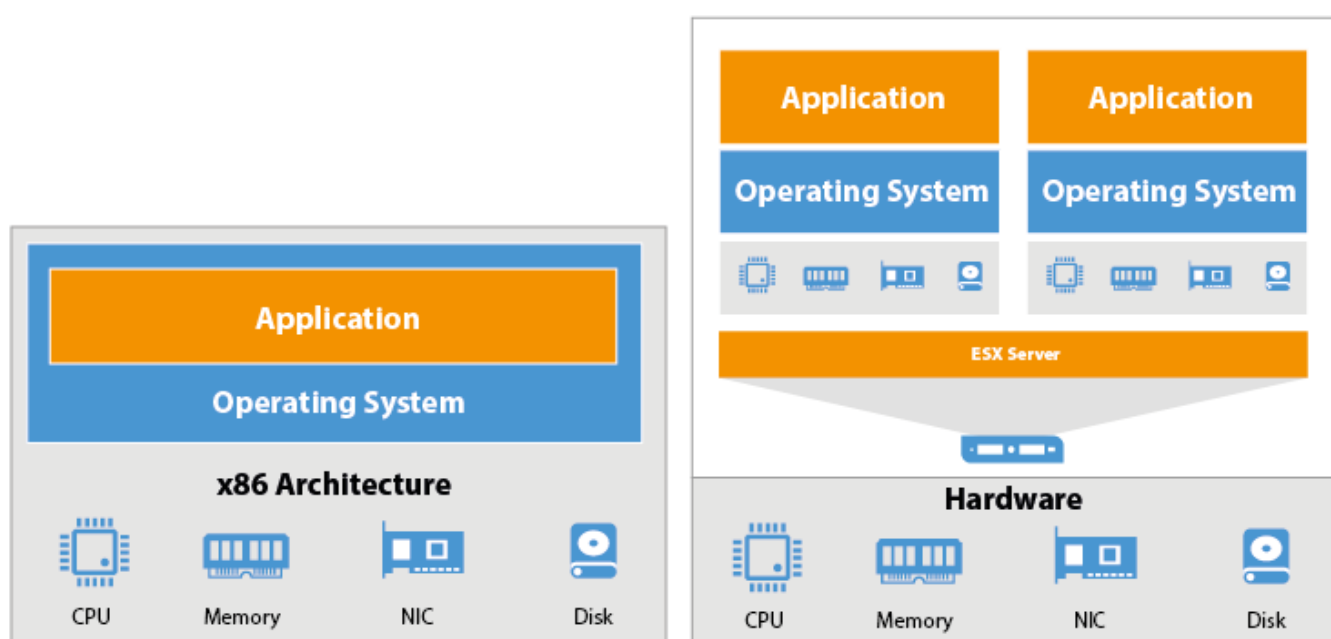
Introducción .....	3
Contenido .....	4
Objetivo de la práctica .....	4
Ventajas y desventajas .....	4
CentOS 8 .....	4
Nginx .....	4
MariaDB .....	5
PHP .....	5
HTTPS .....	5
HTTP v2 .....	6
Comparación de alternativas .....	7
NGINX vs APACHE .....	7
MARIADB vs MYSQL .....	8
Pasos .....	9
Descargar CentOS 8 .....	9
Crear una máquina virtual .....	11
Instalar y configurar CentOS 8 .....	20
Instalar y configurar Nginx .....	26
Instalar y configurar MariaDB .....	29
Instalar y configurar PHP .....	32
Configurar conexión HTTPS .....	35
Configurar conexión HTTP v2 .....	38
Conclusiones .....	40
Referencias .....	41
Anexos .....	42
Administrar el proceso de Nginx .....	42
Solución Error Mariadb .....	42

## Introducción

Un servidor virtual es la simulación de un servidor físico en un entorno virtual, permitiendo ejecutar varios servidores virtuales en una máquina física. Tienen características propias como **redundancia y ejecución en paralelo, aislamiento y aplicaciones variadas**. La versatilidad que ofrecen estos servidores virtuales frente a los tradicionales físicos los convierte en una solución tecnológica ideal.

Para esta práctica se instaló un servidor utilizando una máquina virtual así como configurar el servidor con **Nginx, MariaDB, PHP, conexión HTTPS** y el protocolo de **HTTP versión 2**. Así como las ventajas y desventajas de utilizarlos y que otras alternativas hay.

A continuación se muestra la documentación del proceso de configuración para su consulta a futuro.



*Ilustración 1 - Servidores físicos vs. Máquinas virtuales*

# Contenido

## Objetivo de la práctica

Implementar un servidor web utilizando una máquina virtual para la instalación de **CentOs**, **Nginx**, **MariaDB**, **PHP**, **HTTPS**, **HTTP v2**.

## Ventajas y desventajas

### CentOS 8.

Creemos que CentOS es un sistema operativo el cual vale la pena instalar por varias razones que mencionaremos a continuación. La primera de estas es que tiene una interfaz muy fácil de usar y sencilla, su propósito es el desarrollo y no trata de agregar funciones que no servirían para nada. Otra ventaja que tiene es su facilidad para instalar entornos de desarrollo, al igual que muchos sistemas operativos basados en Linux. Igual es relativamente fácil instalar una máquina virtual de CentOS y encontrar el archivo de imagen de disco. Por último, el hecho de que sea desarrollado y soportado activamente por Red Hat le da bastante compatibilidad con muchas tecnologías.

Sin embargo, igual pensamos que tiene un par de desventajas. La primera es que no es tan conocido como Windows o MacOS, lo cual claramente significa que no hay tanto software y conocimiento general sobre el sistema operativo; esto también puede traer como consecuencia que sea más difícil encontrar recursos de apoyo específicos a este sistema operativo (porque la base de usuario es menor). De igual forma, CentOS tiene un caso de uso muy específico para desarrollar, lo cual limita un poco sus posibilidades; si se busca un sistema operativo para a la vez hacer otro tipo de tareas (oficina, juegos, entre otros), es posible que otra alternativa sea la mejor opción.

### Nginx

Es un software multiplataforma que se puede usar en la gran mayoría de los sistemas operativos, consume menos recursos que otros servicios, proporciona un alto rendimiento soportan mayor carga y respondiendo mejor que otros. Puede usarse como balanceador de carga entre varios servidores, permitiéndonos así una mayor facilidad a la hora de escalar nuestros servidores. Es compatible con una gran variedad de CMS y aplicaciones actuales como pueden ser: Wordpress, Drupal, Prestashop, y muchas más.

El proyecto Nginx tiene detrás a la empresa Nginx Inc. y también cuenta con el apoyo de una gran comunidad contribuyendo a mejorar y resolver dudas. También podemos recurrir a soporte profesional. El único inconveniente es que no se integra con PHP de forma nativa. es necesario usar FastCGI para integrarlo según la página [aruta.com](http://aruta.com)

## **MariaDB**

Las ventajas que encontramos en MariaDB son bastantes. La primera de estas es el gran soporte a varios motores distintos (SphinxSE, TokuDB, Aria, entre otros), lo cual le da bastante versatilidad. Además, como es código abierto tiene una gran cantidad de soporte y desarrollo continuo. Por último, entre los modelos relacionales, MariaDB es una plataforma que ha sido altamente optimizada y tiene un desempeño bastante alto conforme se aumenta la cantidad de datos.

Las desventajas serían que, al igual que otras bases de datos relacionales, no es la mejor para manejar aplicaciones con cantidades de datos masivos (como Facebook, por ejemplo); en estos casos sería mejor evaluar el uso de otros modelos diferentes. La migración de MySQL a MariaDB puede llegar a ser tedioso en versiones más nuevas de este último que no tienen compatibilidad.

## **PHP**

Para fines del curso, pensamos que la instalación de PHP y el trabajo con este lenguaje de programación puede tener una gran cantidad de ventajas. La primera es que PHP es el más adaptado y utilizado en páginas web, según un estudio de W3Tech en 2020, 78.9% de las páginas web lo utilizan; es muy probable que si alguien trabaja en programación web, tenga que saber PHP (Jackson, 2021). Otra ventaja que tiene (a nuestro criterio) es que es un lenguaje relativamente sencillo y fácil de entender en comparación con varias alternativas. Por último, por su gran adaptación, PHP tiene muchísimos recursos que encontrar en internet para facilitar la programación y ahorrar mucho tiempo en cualquier proyecto.

En cuanto a desventajas, es que PHP es un lenguaje relativamente antiguo y poco actualizado (aunque siga teniendo soporte) en comparación a lenguajes más modernos que presentaremos como alternativas. Además, puede ser difícil de mantener programas muy grandes por dos razones: la primera siendo que no es tan optimizado como otros lenguajes, y la segunda porque no es un lenguaje altamente modular.

## **HTTPS**

El objetivo del protocolo es que la página web codifique la sesión con certificado digital, esto hace que tu como usuario intercambies información de manera segura, sin que la información sea interceptada, el protocolo también es llamado como certificado SSL.

Las desventajas que tiene es que hay errores 404, que es que no se redirecciona bien las antiguas urls, y para eso la migración no es fácil.

## HTTP v2

Protocolo binario que conserva las mismas bases de HTTP 1.x por lo que significa que no hay muchos cambios entre las funcionalidades, este busca resolver defectos que tiene la comunicación a través de la capa de transporte. Escanear la siguiente figura para ver el ejemplo.

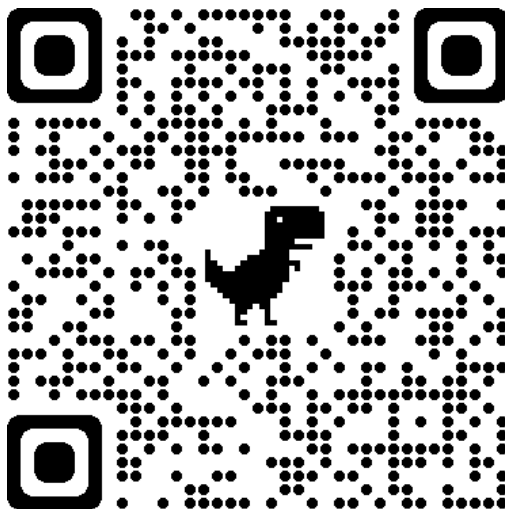


Imagen sacado de <https://somostechies.com/que-es-http2/>

## Comparación de alternativas

### NGINX vs APACHE

La diferencia más grande que hay entre los dos es su arquitectura mientras que Apache abre un montón de procesos para servir peticiones, Nginx abre solo los hilos de ejecución justos y necesarios permitiendo servir millones de peticiones en un corto espacio de tiempo, ya que no requiere tiempo adicional para abrir nuevos procesos y además al no abrir nuevos procesos tampoco consume mas memoria RAM.

El tiempo de respuesta de Nginx es un 150% más rápido que Apache, como se puede ver en la siguiente gráfica:

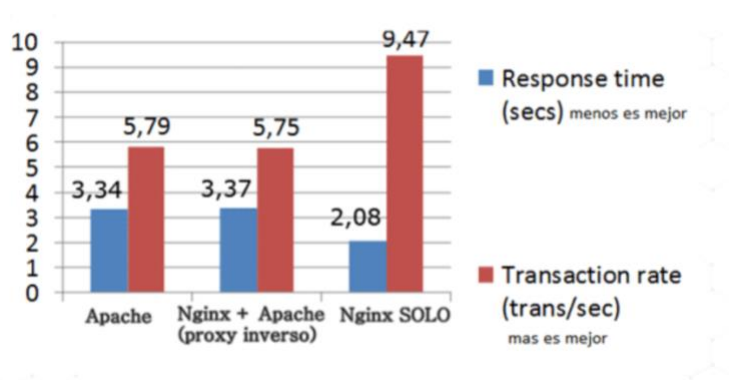


Ilustración 2- Tiempos de respuesta Nginx

Además, con Nginx conseguimos mucha más eficiencia a la hora de servir contenidos y su configuración e instalación es más sencilla que Apache.

## MARIADB vs MYSQL

MariaDB incorpora nuevas tablas a nivel del sistema, que ayudan en las tareas de optimización. No tiene ninguna desventaja frente a MySQL ya que tiene mayor rendimiento, más funcionalidades y distribuye bajo licencia GPL.

MariaDB	MySQL
El rendimiento de MariaDB es mejor.	El rendimiento de MySQL no es tan bueno en comparación con MariaDB.
MariaDB permite la vinculación con software de código cerrado.	MySQL no permite la vinculación con software de código cerrado.
MariaDB proporciona un reemplazo directo para MySQL.	MySQL no proporciona un reemplazo para MariaDB.
MariaDB no tiene un complemento de contraseña de validación de forma predeterminada.	MySQL tiene un complemento de contraseña de validación de forma predeterminada.
MariaDB no admite el tipo JSON nativo, pero admite el tipo de columna de texto.	MySQL admite el tipo JSON nativo.
El protocolo MySQL no se sigue en MariaDB.	MySQL sigue el protocolo MySQL.
Las correcciones de errores se realizan con más frecuencia en MariaDB.	En MySQL, las correcciones de errores se realizan con menos frecuencia.
Las consultas complejas se pueden resolver más rápido.	Las consultas complejas no se resuelven más rápido.
El motor de almacenamiento de memoria es mejor en MariaDB.	En MySQL, los motores de almacenamiento de memoria no son tan rápidos como MariaDB.
El código propietario no es accesible en MariaDB.	Se puede acceder al código propietario en MySQL Enterprise Edition.
Los modelos de bases de datos secundarios son Document store y Graph DBMS.	El modelo de base de datos secundario es el almacén de documentos.
OS X no es compatible como servidor en MariaDB.	OS X es compatible como servidor en MySQL.

Tabla sacada de <https://www.educba.com/mariadb-vs-mysql/>



## Pasos

### Descargar CentOS 8.

El profesor nos proporcionó la imagen de CentOS 8. Para los que no tienen el recurso, los pasos son los siguientes:

1. Visitar la página web oficial de [CentOs](https://www.centos.org) para observar las diferentes opciones e ingresar a download:

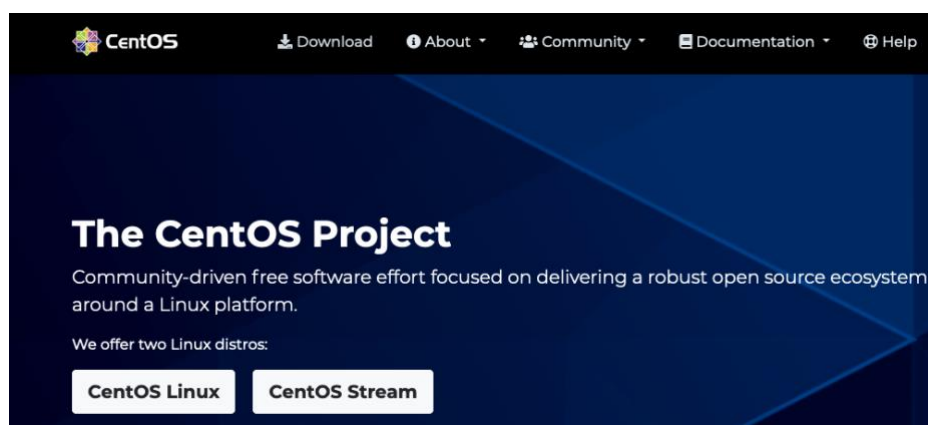


Ilustración 3 - Página principal de CentOS

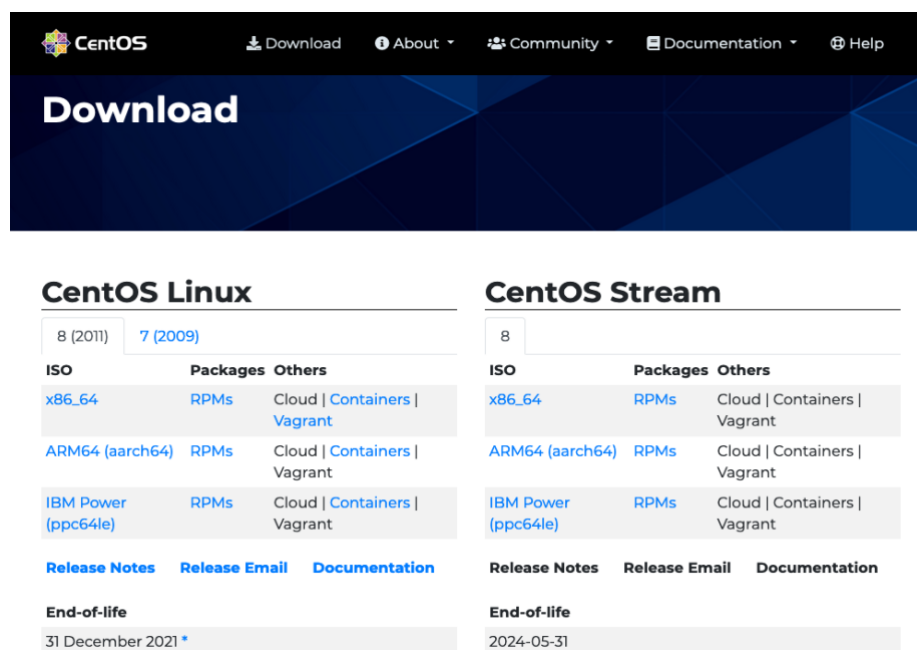


Ilustración 4- Página de descarga de CentOS

2. Se puede observar el panel de **CentOs Linux**, mostrando las diferentes versiones para las distribuciones disponibles para descargar.
3. Dependiendo de tu arquitectura, te dirigirá a las listas de espejos o mirrors dependiendo la arquitectura.
4. Para esta práctica utilizaremos **x86\_64**  
[http://isoredirect.centos.org/centos/8/isos/x86\\_64/](http://isoredirect.centos.org/centos/8/isos/x86_64/)
5. Descargar.



Ilustración 5- Muestra el archivo que se descarga

## Crear una máquina virtual.

### VMWARE

Tener instalado VMWare Fusion, VMWare Workshop o alguna de las alternativas de VMWARE que funcionen como máquina virtual. Se necesita pagar por una licencia para el uso del software. En caso de que no, descargalo y/o comprar la licencia en: <https://www.vmware.com>

1. Si no tienes ningún servicio instalado en VMWare Fusion, te aparecerá la siguiente ventana:



Ilustración 6- Ventana de Vmware

Si no es el caso, buscar por un símbolo de “+” para instalar una máquina virtual nueva.

2. Instalar el sistema operativo desde alguna de las opciones que ofrece VMWare Fusion. Para esto recomendamos simplemente arrastrar el archivo descargado a la parte superior de la ventana.

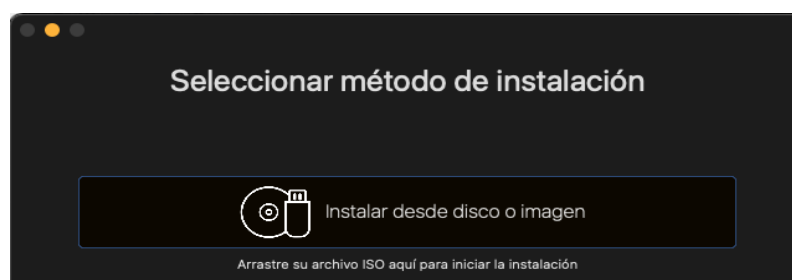


Ilustración 7- Modo de selección

3. Una vez arrastrado, te aparecerá la siguiente ventana con el respectivo archivo del sistema operativo que descargaste. Puedes darle click a *continuar*.

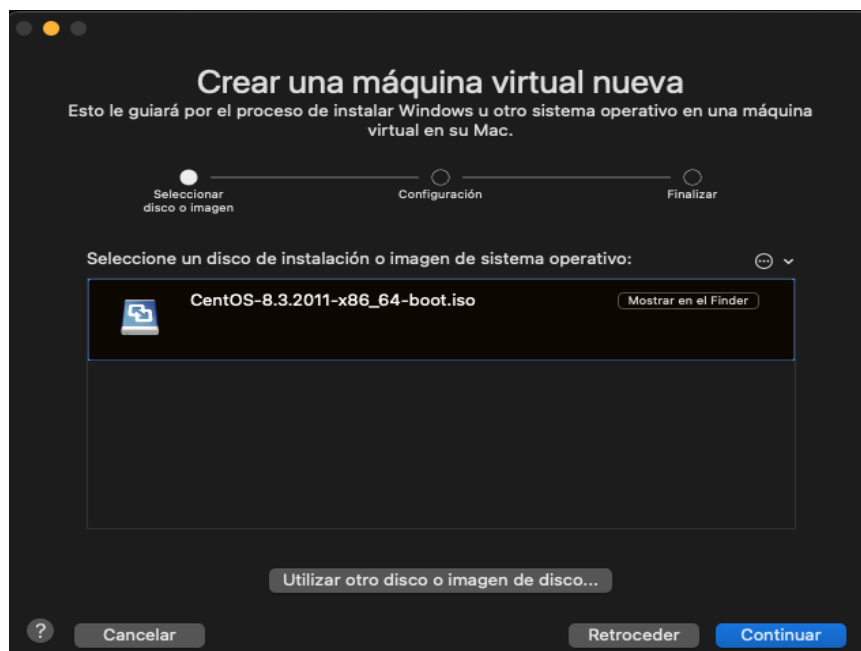


Ilustración 8- Ventana de archivo

4. Para la configuración, busca y selecciona el sistema operativo correspondiente al archivo que subiste. En nuestro caso buscamos CentOS 8 (64 bits).

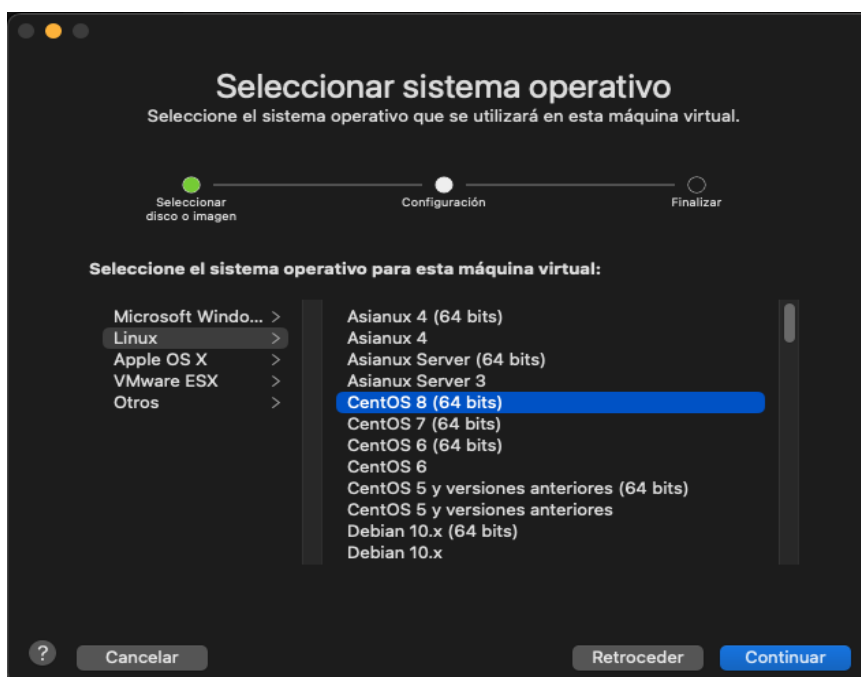


Ilustración 9- Ventana de Selección de Sistema Operativo

5. Seleccionar “BIOS heredado” como tipo de firmware. Darle click a continuar.

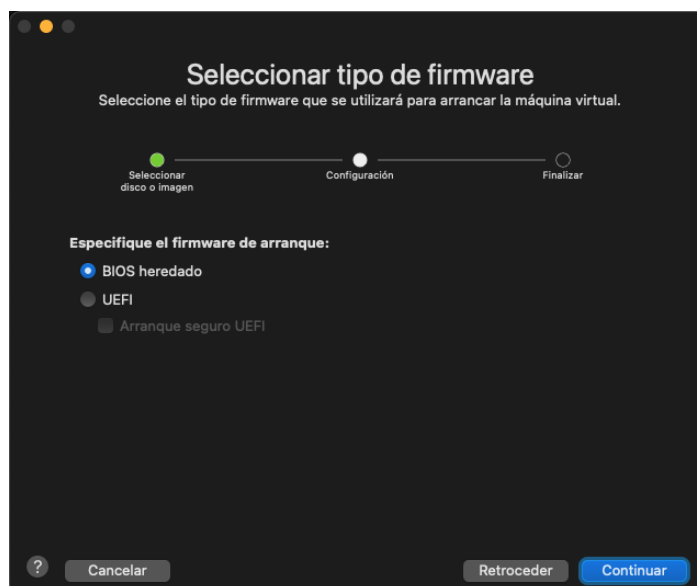


Ilustración 10 - Seleccionar la Bios

6. Se te desplegará la configuración por default de tu sistema operativo. Si estás conforme con la configuración actual, dale click a finalizar. Si no estás conforme, puedes personalizarlo en el botón de “Personalizar configuración”.

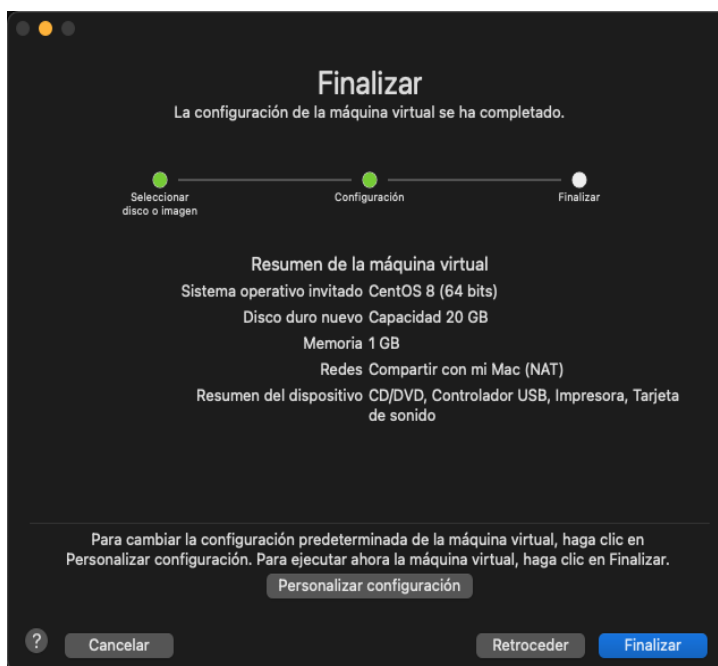


Ilustración 11 - Ventana de Configuración Default

7. Finalmente, dale un nombre a tu máquina virtual (recomendablemente el que viene por default). De igual manera, selecciona en dónde quieres guardar tu máquina virtual. Nuevamente recomendamos la carpeta que aparezca por defecto para una mejor organización de máquinas virtuales.

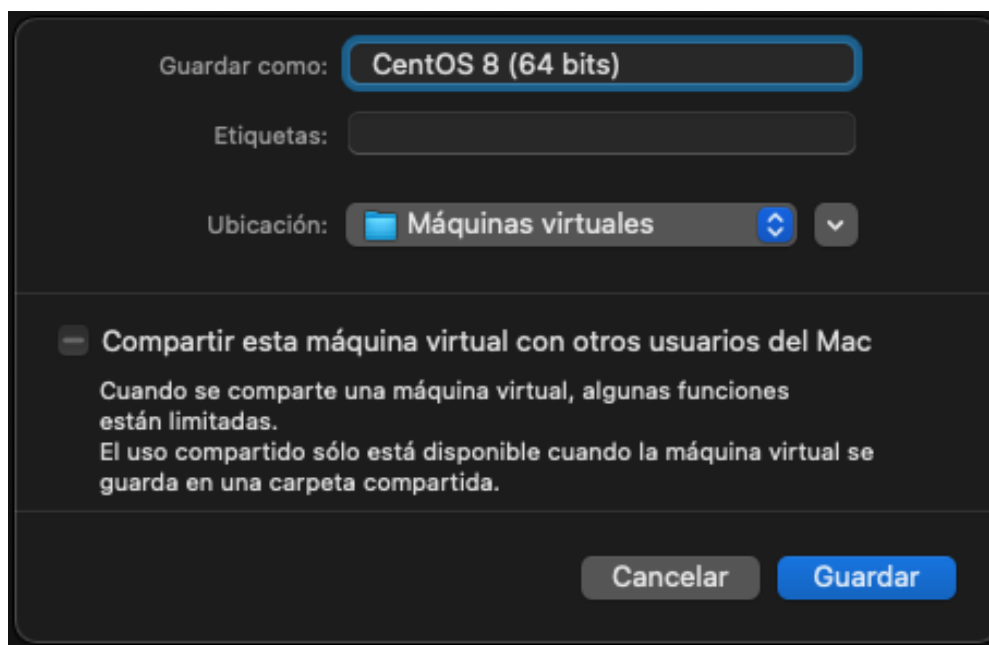


Ilustración 12 -Ventana para guardar la VM

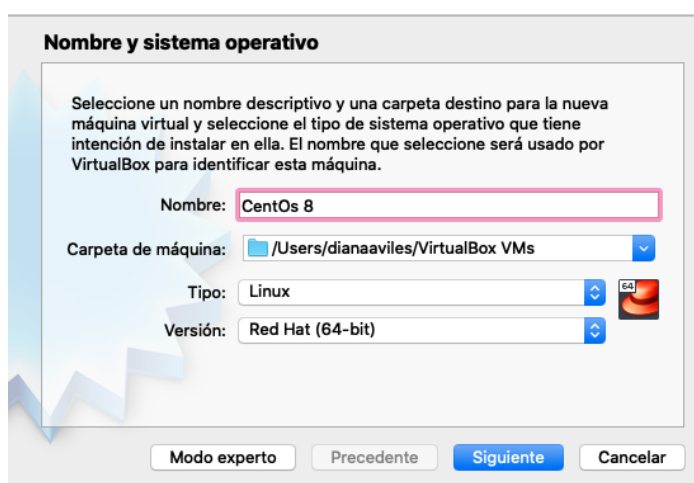
## VIRTUALBOX

Tener VirtualBox instalado, en caso de que no, descargalo en : <https://www.virtualbox.org/>

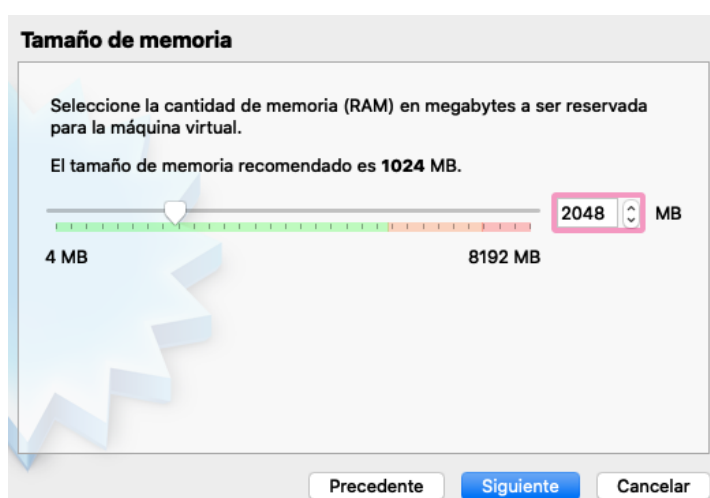
1. Seleccionar nueva para crear la máquina virtual.



2. Indicar el nombre de la máquina y en donde se almacenará.



3. Al darle siguiente, te mostrará el tamaño de memoria, se puede empezar con 2048 MB (2GB). Siguiente



#### 4. Seleccionar “Crear el disco duro virtual ahora”

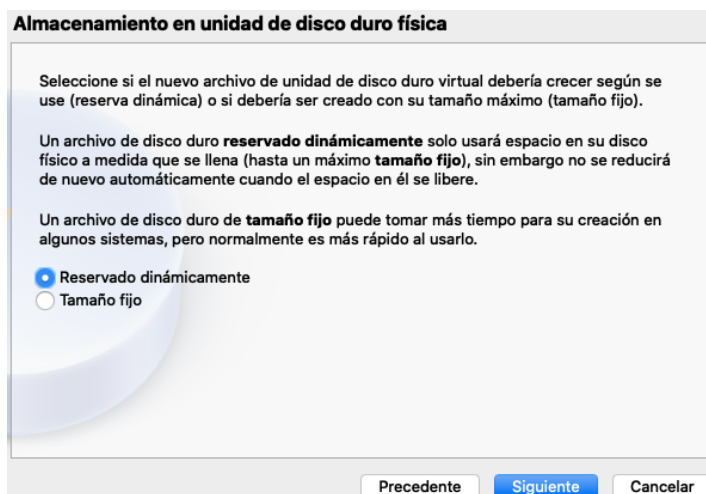


#### 5. Seleccionar para Tipo de archivo de disco duro: Virtual Disk image (VDI)

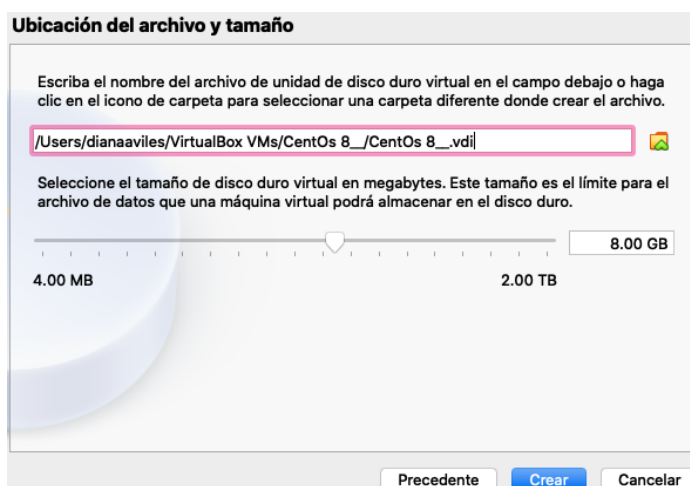




## 6. Seleccionar Reservar dinámicamente.



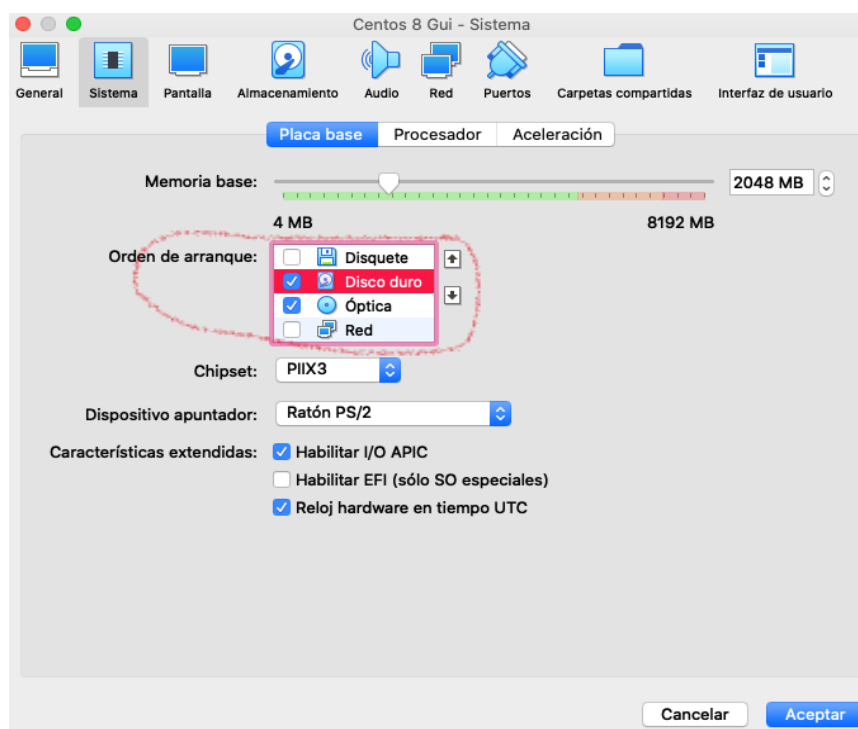
7. Por último, reserva el tamaño que necesites, dependerá en uso que se le dará a futuro. Para esta práctica que actuará como servidor de pruebas y configuraciones, 8GB es perfecto. Da clic en **crear**.



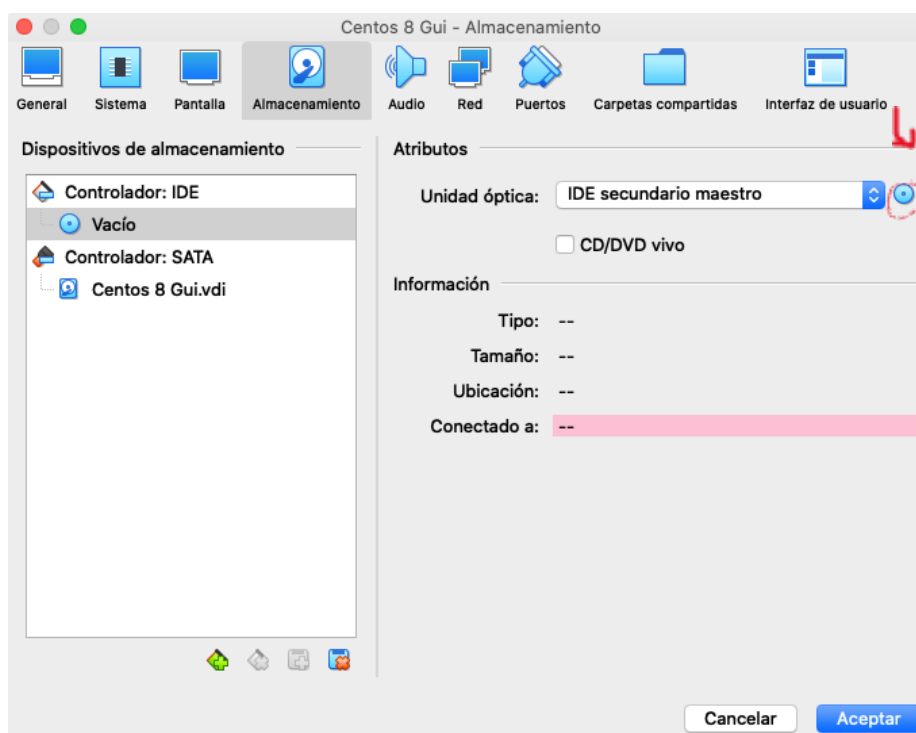
8. Ahora configuraremos la máquina virtual creada. Selecciona configuración.



9. Puedes deshabilitar el disquete en **orden de arranque** y poner al principal al disco duro. También Seleccionar PIX3 e ICH9.



10. En la parte de almacenamiento, se carga la imagen del medio de instalación de CentOS 8, que se carga previamente. Pulsa en el icono para buscar este ISO.



11. Dale en aceptar y por último, selecciona la máquina virtual que se acaba de crear y dale clic en **Iniciar** para instalar CentOS 8.



## Instalar y configurar CentOS 8.

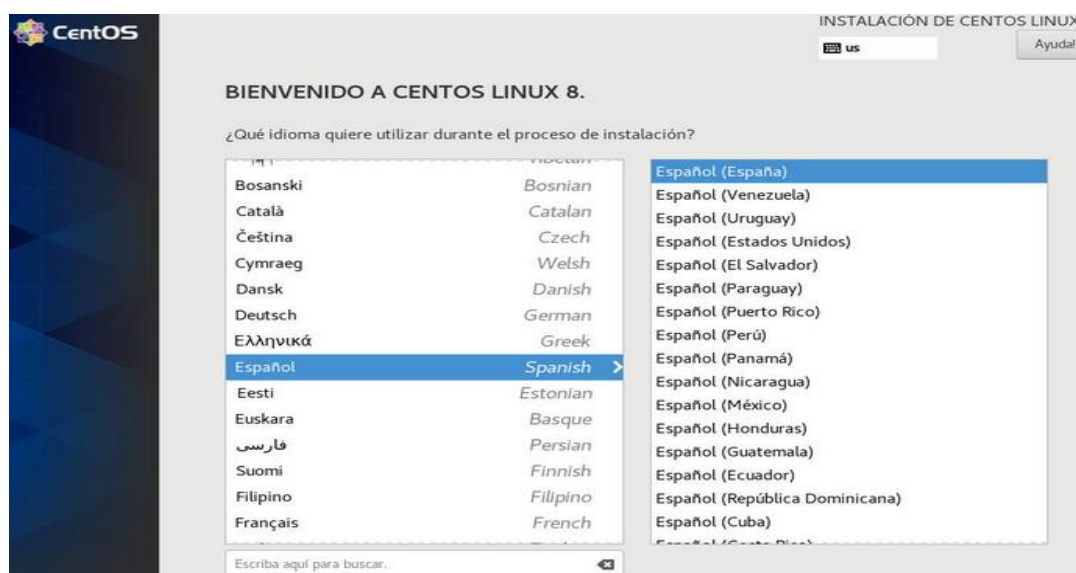
Al arrancar la máquina virtual que se acaba de crear, aparecerá el siguiente menú:



1. Seleccionar “**Install CentOS Linux 8**” y presionar **TAB**
2. Añadir el argumento **vga=792** para que funcione con la 1024x768 píxeles y 24 bits de resolución. Y darle **enter**.

```
> vmlinuz initrd=initrd.img inst.stage2=hd:LABEL=CentOS-8-3-2011-x86_iet vga=792_
```

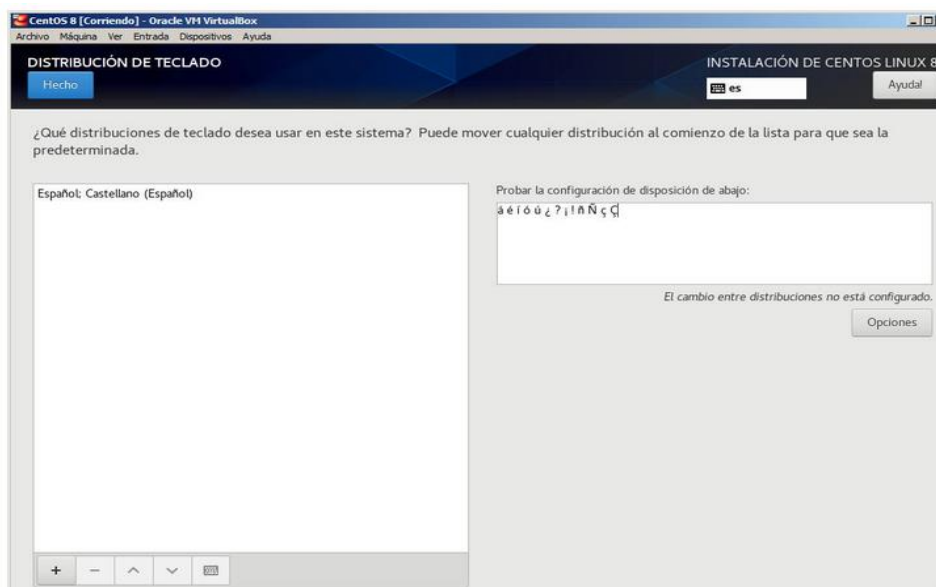
3. Seleccionar el idioma de tu preferencia y continuar.



4. Aparecerán las opciones de instalador, algunos valores los toma por defecto.

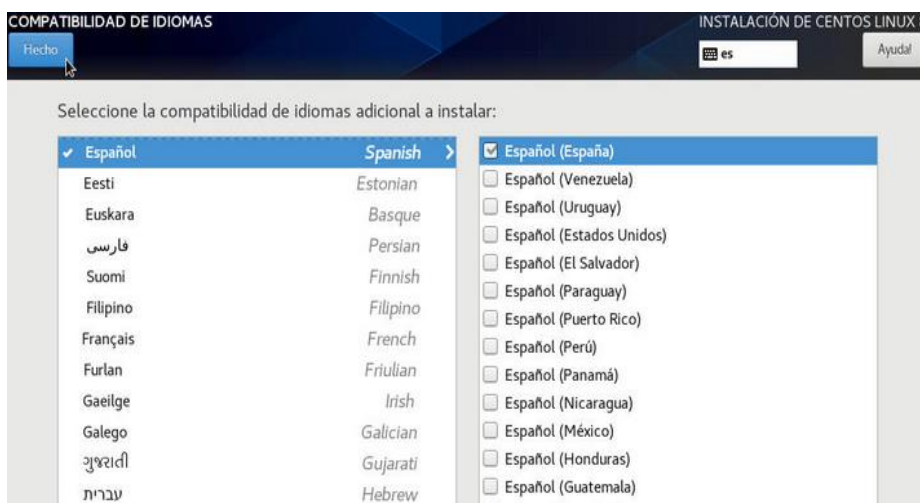


5. Selecciona la distribución de teclado de tu preferencia, puedes añadir más según el idioma.



Siempre selecciona **HECHO** al terminar tu proceso.

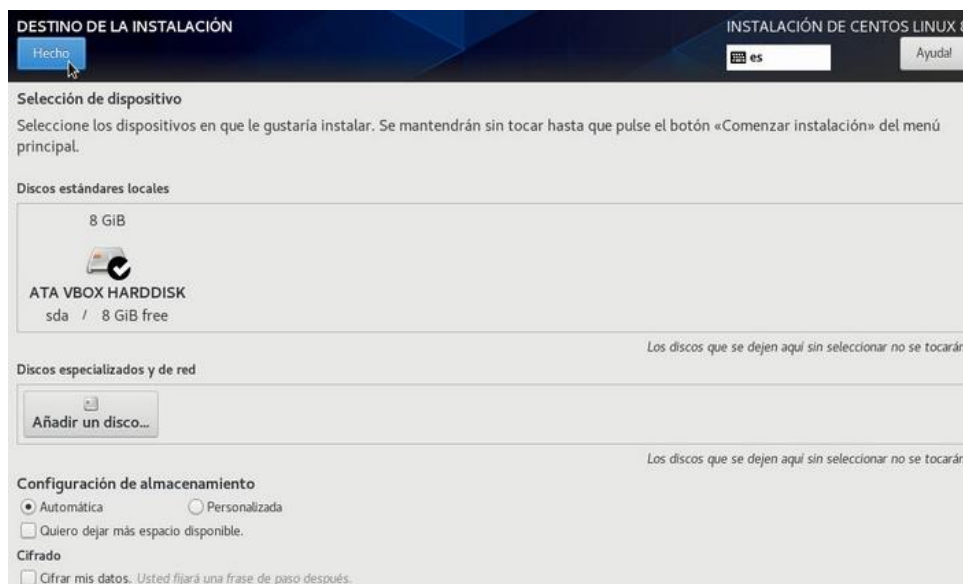
6. Escoge el idioma de tu preferencia.



7. Selecciona la fecha y hora según tu ubicación:



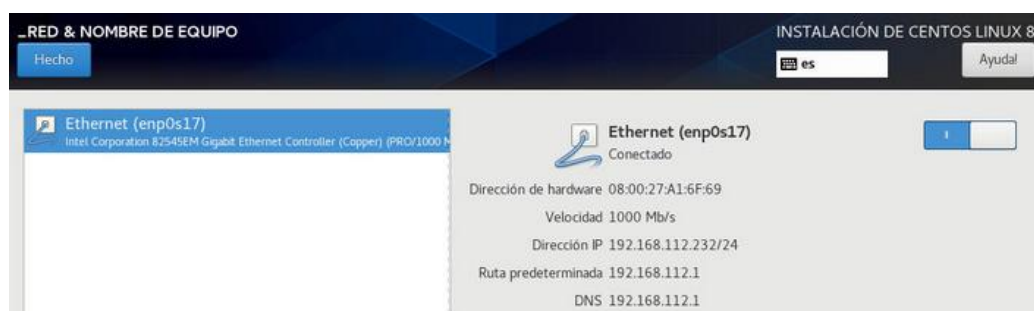
8. Elegir el destino de instalación en caso de que hubiera más de uno, por ahora selecciona el que tienes disponible:



9. Selecciona KDUMP. Este es un sistema de memoria del kernel de CentOS en caso de fallo. Este consume recursos del sistema. Para este caso deshabilitamos la opción.

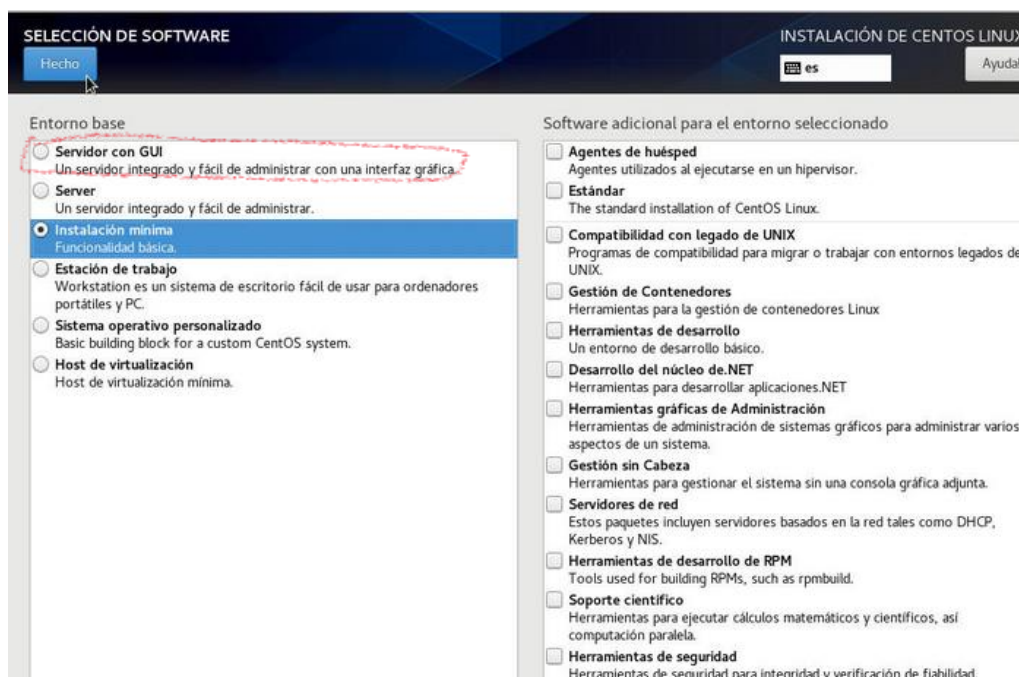


10. Configuramos el nombre de la máquina (opcional su dominio) y posteriormente configuramos el dispositivo de red, activándolo.

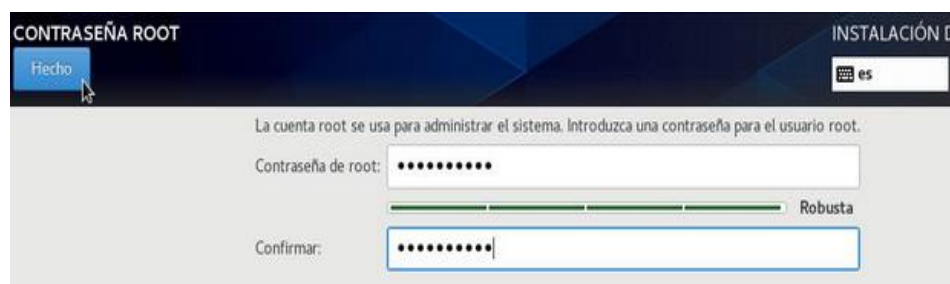




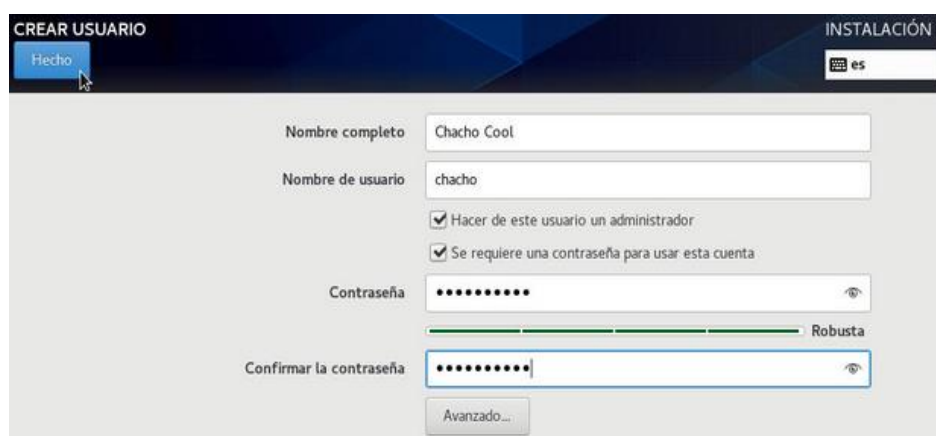
11. En selección de software, elegir el que instalaremos en nuestro sistema CentOS 8 en este caso **Servidor con GUI**



12. Seleccionar una contraseña para el root.



13. Crear un usuario y contraseña.





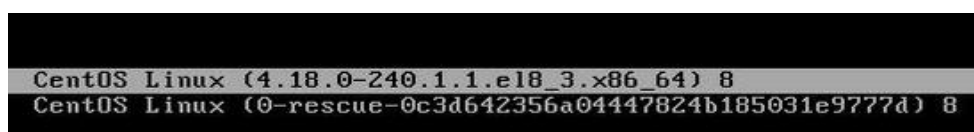
14. ¡Listo! Podrás iniciar la instalación.



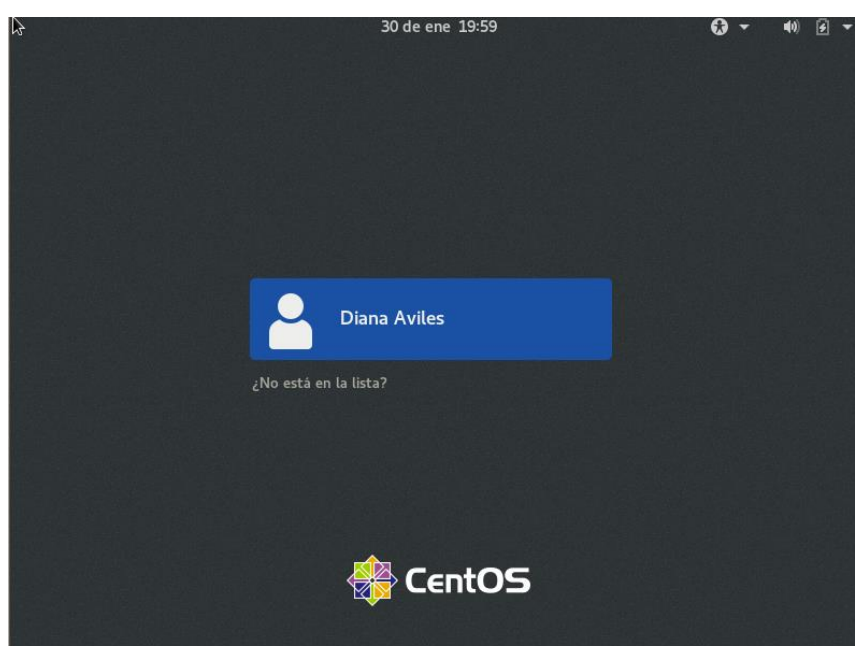
15. Se tardará un poco, dependiendo de los recursos asignados y la potencia de la máquina virtual.

**16. Al concluir reinicia el sistema.**

17. Aparecerá de manera consecuente el siguiente menú de arranque:



18. Seleccionas esa por defecto y ¡Listo! tendremos la pantalla de login para comenzar a utilizarlo.



## Instalar y configurar Nginx.

1. Instalar el paquete Nginx.  
**sudo dnf install nginx**
2. Confirmar lo que desea instalar, dnf instalará Nginx y cualquier dependencia para su servidor.

```
avilesdianae@localhost:~  
Archivo Editar Ver Buscar Terminal Ayuda  
Instalando      : nginx-mod-stream-1:1.14.1-9.module_el8.0.0+184+e3 8/8  
Ejecutando scriptlet: nginx-mod-stream-1:1.14.1-9.module_el8.0.0+184+e3 8/8  
Verificando     : nginx-1:1.14.1-9.module_el8.0.0+184+e34fea82.x86_ 1/8  
Verificando     : nginx-all-modules-1:1.14.1-9.module_el8.0.0+184+e 2/8  
Verificando     : nginx-filesystem-1:1.14.1-9.module_el8.0.0+184+e3 3/8  
Verificando     : nginx-mod-http-image-filter-1:1.14.1-9.module_el8 4/8  
Verificando     : nginx-mod-http-perl-1:1.14.1-9.module_el8.0.0+184 5/8  
Verificando     : nginx-mod-http-xslt-filter-1:1.14.1-9.module_el8. 6/8  
Verificando     : nginx-mod-mail-1:1.14.1-9.module_el8.0.0+184+e34f 7/8  
Verificando     : nginx-mod-stream-1:1.14.1-9.module_el8.0.0+184+e3 8/8  
Installed products updated.  
  
Instalado:  
nginx-1:1.14.1-9.module_el8.0.0+184+e34fea82.x86_64  
nginx-all-modules-1:1.14.1-9.module_el8.0.0+184+e34fea82.noarch  
nginx-filesystem-1:1.14.1-9.module_el8.0.0+184+e34fea82.noarch  
nginx-mod-http-image-filter-1:1.14.1-9.module_el8.0.0+184+e34fea82.x86_64  
nginx-mod-http-perl-1:1.14.1-9.module_el8.0.0+184+e34fea82.x86_64  
nginx-mod-http-xslt-filter-1:1.14.1-9.module_el8.0.0+184+e34fea82.x86_64  
nginx-mod-mail-1:1.14.1-9.module_el8.0.0+184+e34fea82.x86_64  
nginx-mod-stream-1:1.14.1-9.module_el8.0.0+184+e34fea82.x86_64  
¡Listo!  
[avilesdianae@localhost ~]$
```

3. Ya que finalizó, ejecutar los siguientes comandos para habilitar e iniciar el servidor:

```
avilesdianae@localhost:~  
Archivo Editar Ver Buscar Terminal Ayuda  
[avilesdianae@localhost ~]$ sudo systemctl enable nginx  
[sudo] password for avilesdianae:  
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.  
[avilesdianae@localhost ~]$ systemctl start nginx
```

4. Ahora ajustamos las reglas de firewall para permitir conexiones externas al servidor web Nginx, que se ejecutan en el puerto 80 de forma predeterminada. Al ejecutar el siguiente comando, habilitará las conexiones HTTP y HTTPS en el puerto 80.

```
[avilesdiana@avilesdiana ~]$ sudo firewall-cmd --permanent --add-service={http,https}
```

5. Para verificar que el servicio firewall http se añadió, se ejecuta el siguiente:

```
[avilesdianae@localhost ~]$ firewall-cmd --permanent --list-all
```

6. Debe aparecerte activado el http en services:

```
[avilesdiana@avilesdiana ~]$ sudo firewall-cmd --permanent --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client http https ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[avilesdiana@avilesdiana ~]$
```

En caso de que no te aparezcas, tienes problemas de firewall, la forma en que lo solucione esta en este siguiente enlace:

<https://www.digitalocean.com/community/tutorials/initial-server-setup-with-centos-8>

7. Ahora guarda los cambios con el siguiente comando:

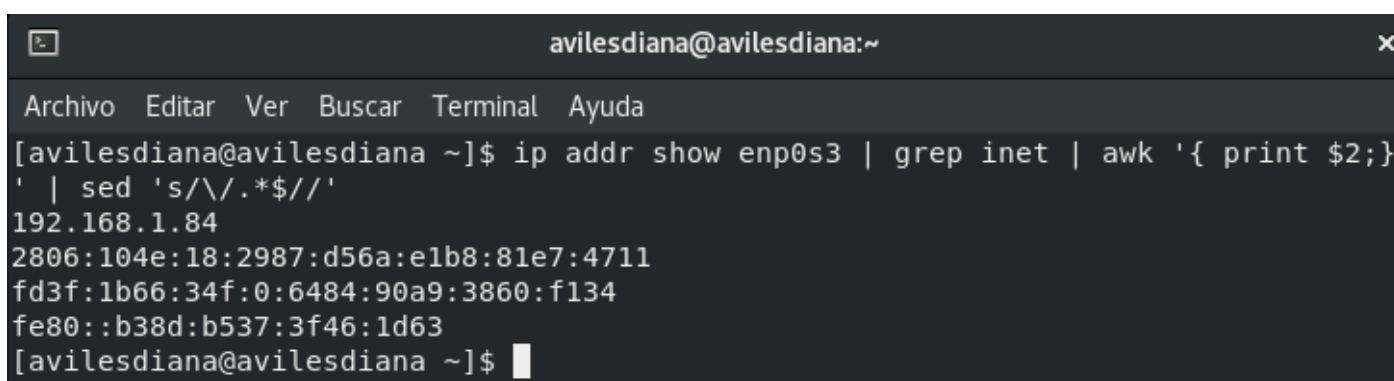
```
[avilesdiana@localhost ~]$ sudo firewall-cmd --reload
[sudo] password for avilesdiana:
success
[avilesdiana@localhost ~]$
```

Ahora comprobaremos que tu servidor web funciona accediendo a su dirección IP pública o nombre de dominio desde el navegador.

Si no tienes un nombre de dominio para el servidor y no conoces tu ip pública, ejecuta el siguiente comando:

```
$ ip addr show eth0 | grep inet | awk '{ print $2; }' | sed 's/\./.$$/'
```

**Eth0 -> Ingresa el nombre de tu red, en mi caso en enp0s3**



```
avilesdiana@avilesdiana:~  
Archivo Editar Ver Buscar Terminal Ayuda  
[avilesdiana@avilesdiana ~]$ ip addr show enp0s3 | grep inet | awk '{ print $2; }'  
' | sed 's/\./.$$/'  
192.168.1.84  
2806:104e:18:2987:d56a:e1b8:81e7:4711  
fd3f:1b66:34f:0:6484:90a9:3860:f134  
fe80::b38d:b537:3f46:1d63  
[avilesdiana@avilesdiana ~]$
```

¡Listo! Escribes la ip en tu navegador web y te llevará a la siguiente página de destino predeterminada por Nginx:



En parte de [Anexo](#) se encuentra como [administrar el proceso en Nginx](#).

## Instalar y configurar MariaDB.

1. Instalar el paquete de MariaDB.

```
avilesdianae@localhost:~  
Archivo Editar Ver Buscar Terminal Ayuda  
[avilesdianae@localhost ~]$ sudo dnf install mariadb-server
```

2. Confirmar todas las acciones para finalizar.

```
avilesdianae@localhost:~  
Archivo Editar Ver Buscar Terminal Ayuda  
Verificando      : mariadb-common-3:10.3.27-3.module_el8.3.0+599+c 3/10  
Verificando      : mariadb-connector-c-3.1.11-2.el8_3.x86_64      4/10  
Verificando      : mariadb-connector-c-config-3.1.11-2.el8_3.noarc 5/10  
Verificando      : mariadb-errmsg-3:10.3.27-3.module_el8.3.0+599+c 6/10  
Verificando      : mariadb-gssapi-server-3:10.3.27-3.module_el8.3. 7/10  
Verificando      : mariadb-server-3:10.3.27-3.module_el8.3.0+599+c 8/10  
Verificando      : mariadb-server-utils-3:10.3.27-3.module_el8.3.0 9/10  
Verificando      : perl-DBD-MySQL-4.046-3.module_el8.1.0+203+e4542 10/10  
Installed products updated.  
  
Instalado:  
mariadb-3:10.3.27-3.module_el8.3.0+599+c587b2e7.x86_64  
mariadb-backup-3:10.3.27-3.module_el8.3.0+599+c587b2e7.x86_64  
mariadb-common-3:10.3.27-3.module_el8.3.0+599+c587b2e7.x86_64  
mariadb-connector-c-3.1.11-2.el8_3.x86_64  
mariadb-connector-c-config-3.1.11-2.el8_3.noarch  
mariadb-errmsg-3:10.3.27-3.module_el8.3.0+599+c587b2e7.x86_64  
mariadb-gssapi-server-3:10.3.27-3.module_el8.3.0+599+c587b2e7.x86_64  
mariadb-server-3:10.3.27-3.module_el8.3.0+599+c587b2e7.x86_64  
mariadb-server-utils-3:10.3.27-3.module_el8.3.0+599+c587b2e7.x86_64  
perl-DBD-MySQL-4.046-3.module_el8.1.0+203+e45423dc.x86_64  
  
¡Listo!  
[avilesdianae@localhost ~]$
```

3. Al completar la instalación iniciar el servicio con systemctl:

```
$ sudo systemctl start mariadb
```

4. Comprobar el estado de servicio con:

```
$ sudo systemctl status mariadb
```

5. Le aparecerá la siguiente pantalla:

```

avilesdianae@localhost:~
Archivo Editar Ver Buscar Terminal Ayuda
● mariadb.service - MariaDB 10.3 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; vendor pr
   Active: active (running) since Sat 2021-01-30 20:25:15 CST; 13s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 37888 ExecStartPost=/usr/libexec/mysql-check-upgrade (code=exited, s
   Process: 37742 ExecStartPre=/usr/libexec/mysql-prepare-db-dir mariadb.servic
   Process: 37718 ExecStartPre=/usr/libexec/mysql-check-socket (code=exited, sta
   Main PID: 37856 (mysqld)
    Status: "Taking your SQL requests now..."
     Tasks: 30 (limit: 11324)
    Memory: 101.4M
   CGroup: /system.slice/mariadb.service
           └─37856 /usr/libexec/mysqld --basedir=/usr

ene 30 20:25:14 localhost.localdomain mysql-prepare-db-dir[37742]: See the Mari
ene 30 20:25:14 localhost.localdomain mysql-prepare-db-dir[37742]: MySQL manual
ene 30 20:25:14 localhost.localdomain mysql-prepare-db-dir[37742]: Please repor
ene 30 20:25:14 localhost.localdomain mysql-prepare-db-dir[37742]: The latest i
ene 30 20:25:14 localhost.localdomain mysql-prepare-db-dir[37742]: You can find
ene 30 20:25:14 localhost.localdomain mysql-prepare-db-dir[37742]: http://dev.m
ene 30 20:25:14 localhost.localdomain mysql-prepare-db-dir[37742]: Consider jo
ene 30 20:25:14 localhost.localdomain mysql-prepare-db-dir[37742]: https://mar
lines 1-23
  
```

6. Comprobar que MariaDb se inicie en el arranque con el comando:

```
[avilesdianae@localhost ~]$ sudo systemctl enable mariadb
```

7. Ahora tenemos MariaDB activo y configurado para que se ejecute en el arranque. A continuación, protegeremos nuestra instalación.

8. Utiliza el siguiente comando para ejecutar la secuencia de comandos de seguridad:

```
[avilesdianae@localhost ~]$ sudo mysql_secure_installation
```

9. En el primer paso se solicita la contraseña root, que no se ha configurado. Por lo tanto, debe pulsar **INTRO** como se indica. A continuación, se le solicitará establecer esa contraseña root. Tenga en cuenta que esto es para el usuario de la base de datos root, no para el usuario root de su propio servidor de CentOS.

10. Escriba **Y** y luego **INTRO** para introducir una contraseña para el usuario de la base de datos root y siga las indicaciones.

11. Después de actualizar la contraseña, acepte todas las sugerencias de seguridad siguientes pulsando **Y** y luego **INTRO**. Con esto se eliminarán usuarios anónimos, **no se permitirá el inicio de sesión remoto de root**, se **eliminará la base de datos de prueba** y se **volverán a cargar las tablas de privilegios**.

12. Ahora probaremos la instalación, para obtener información al respecto estableciendo conexión mediante la herramienta **mysqladmin**, un cliente que le permite ejecutar comandos administrativos. Utilice el siguiente comando para establecer la conexión con Maria DB como **root** :

```
[avilesdiana@localhost ~]$ mysqladmin -u root -p version
```

13. Como resultado:

```
[root@avilesdiana ~]# mysqladmin -u root -p version
Enter password:
mysqladmin Ver 9.1 Distrib 10.3.27-MariaDB, for Linux on x86_64
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Server version          10.3.27-MariaDB
Protocol version        10
Connection              Localhost via UNIX socket
UNIX socket             /var/lib/mysql/mysql.sock
Uptime:                 1 min 41 sec

Threads: 7  Questions: 16  Slow queries: 0  Opens: 17  Flush tables: 1  Open tables: 1
1  Queries per second avg: 0.158
[root@avilesdiana ~]#
```

**ESTO INDICA QUE SE INSTALÓ CORRECTAMENTE.**

En mi caso tuve un error de mysqladmin que no me permitía ver la versión de MariaDb, mi solución fue Desinstalar Mariadb y volver a instalar de cero. El código se encuentra en el [Anexo](#), [solución error Mariadb](#).



## Instalar y configurar PHP.

Para instalar PHP , se pueden utilizar los paquetes disponibles en la distribución, sin repositorios adicionales.

Utiliza este comando para actualizar la información de repositorios:

```
~$ sudo yum update
```

Para esta practica se instalar para la versión **PHP-FPM** más actual para nginx, este significa **“PHP fastCGI process manager”** , este indica a nginx que pase solicitudes php a este software, de igual manera se necesitará **php-mysqlnd**, este módulo permite a php comunicarse con la base de datos de MySQL.

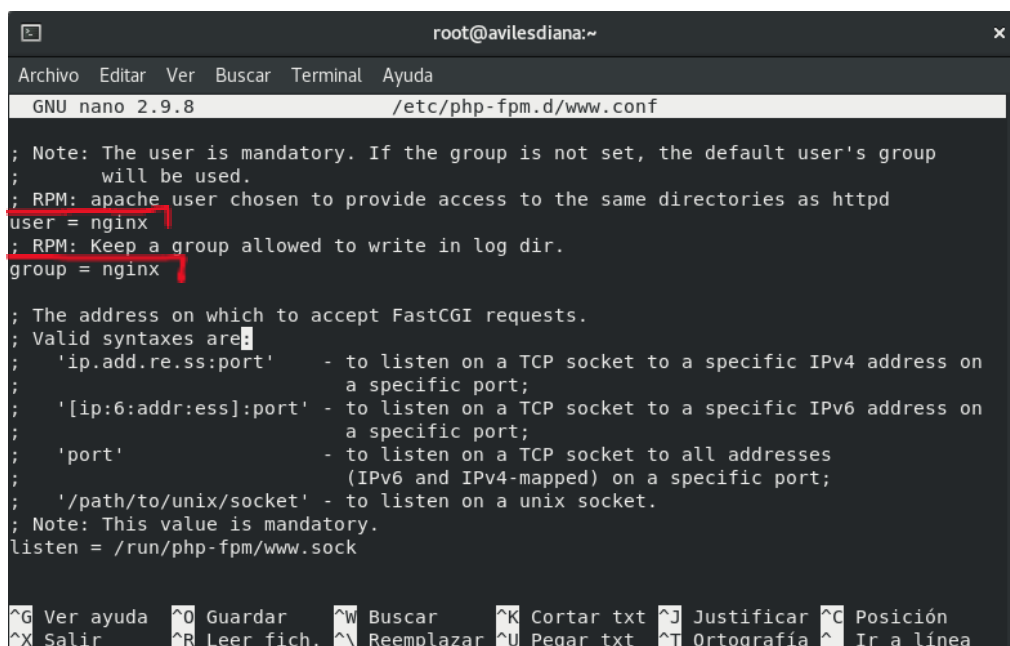
1. Para empezar la instalación ejecutamos lo siguiente:

```
[root@avilesdiana ~]# sudo dnf install php-fpm php-mysqlnd
```

2. Abriremos el archivo de configuración usando el editor de tu preferencia, en este caso es nano.

```
$ sudo nano /etc/php-fpm.d/www.conf
```

3. Buscaras user y group, cambiaras estas variable de **Apache** a **nginx**, guardas con ctrl + x, S



```
root@avilesdiana:~
GNU nano 2.9.8 /etc/php-fpm.d/www.conf

; Note: The user is mandatory. If the group is not set, the default user's group
; will be used.
; RPM: apache_user chosen to provide access to the same directories as httpd
user = nginx
; RPM: Keep a group allowed to write in log dir.
group = nginx

; The address on which to accept FastCGI requests.
; Valid syntaxes are:
; 'ip.add.re.ss:port'   - to listen on a TCP socket to a specific IPv4 address on
;                        a specific port;
; '[ip:6:addr:ess]:port' - to listen on a TCP socket to a specific IPv6 address on
;                        a specific port;
; 'port'               - to listen on a TCP socket to all addresses
;                        (IPv6 and IPv4-mapped) on a specific port;
; '/path/to/unix/socket' - to listen on a unix socket.
; Note: This value is mandatory.
listen = /run/php-fpm/www.sock

^G Ver ayuda  ^O Guardar   ^W Buscar   ^K Cortar txt ^J Justificar ^C Posición
^X Salir      ^R Leer fich. ^_ Reemplazar ^U Pegar txt  ^T Ortografía ^_ Ir a línea
```



4. Ya que se guardó la configuración, habilitamos e iniciamos el servicio de php-fpm con:

```
[root@avilesdiana ~]# sudo systemctl start php-fpm
```

5. Para finalizar, reiniciamos el servidor nginx para cargar los archivos de configuración creados por la instalación php-fpm.

```
[root@avilesdiana ~]# sudo systemctl restart nginx
```

Ahora veremos cómo probar que php funciona con nginx. Cambiaremos los permisos de predeterminada en la carpeta raíz de los documentos de nginx, para poder crear y modificar archivos en ese directorio, sin la necesidad de añadir sudo en cada comando.

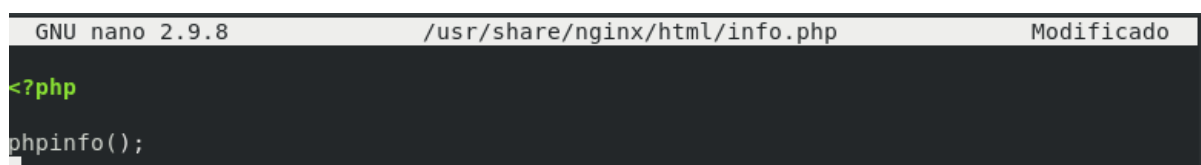
1. Cambiaremos la propiedad de la raíz a un usuario y grupo (con tu nombre de usuario):

```
[root@avilesdiana ~]# sudo chown -R avilesdiana.avilesdiana /usr/share/nginx/html/
```

2. Crearemos una página php como prueba para verificar que el servidor funciona como se espera, con el siguiente comando:

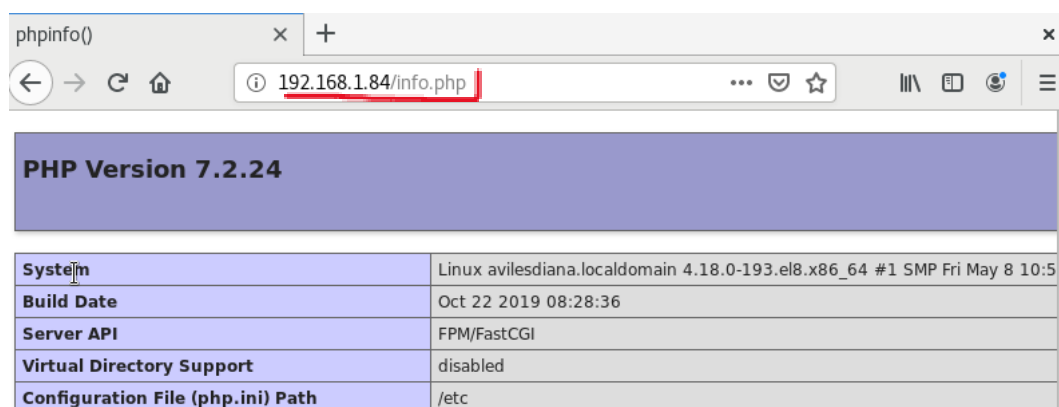
```
[root@avilesdiana ~]# nano /usr/share/nginx/html/info.php
```

3. Ingresamos el siguiente código, al terminar, lo guardas y cierras el archivo.



```
GNU nano 2.9.8 /usr/share/nginx/html/info.php Modificado
<?php
phpinfo();
```

4. Ahora, vamos al navegador e ingresamos la ip de nuestro servidor agregando el archivo php, como se muestra a continuación:



System	Linux avilesdiana.localdomain 4.18.0-193.el8.x86_64 #1 SMP Fri May 8 10:5
Build Date	Oct 22 2019 08:28:36
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc

5. Al comprobar la información, se sugiere eliminar el archivo creado, ya que contiene información confidencial sobre su php y servidor, utilice el siguiente comando para eliminarlo:

```
[root@avilesdiana ~]# rm /usr/share/nginx/html/info.php
```

6. ¡Listo! Ya está configurado php, se puede regenerar este archivo cuando sea necesario.

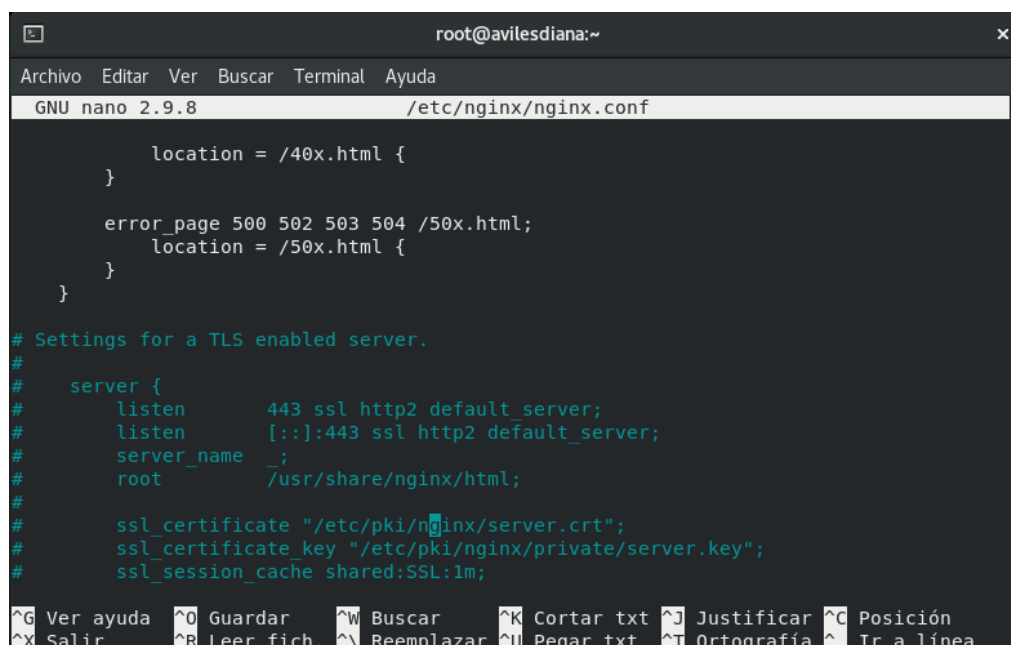
## Configurar conexión HTTPS.

Anteriormente ya se configuró el firewall con http y https. Lo que se configurará ahora es SSL(HTTPS), este cifra las conexiones y es necesario disponer de certificados. Es importante tener certificados firmados por autoridades certificadoras. Como este es un servidor de pruebas se puede utilizar un **certificado autofirmado**.

1. Editaremos el archivo de configuración con el siguiente comando:

```
[root@avilesdiana ~]# sudo nano /etc/nginx/nginx.conf
```

2. Buscamos el bloque server, que aparecer de la siguiente forma:



```
root@avilesdiana:~
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.8 /etc/nginx/nginx.conf

        location = /40x.html {
        }

        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
        }
    }

# Settings for a TLS enabled server.
#
#    server {
#        listen      443 ssl http2 default_server;
#        listen      [::]:443 ssl http2 default_server;
#        server_name  _;
#        root         /usr/share/nginx/html;
#
#        ssl_certificate "/etc/pki/nginx/server.crt";
#        ssl_certificate_key "/etc/pki/nginx/private/server.key";
#        ssl_session_cache shared:SSL:1m;
#    }
```

3. Quitaremos los comentarios para que sea reconocible. Procedemos a dirigirnos a la parte de ssl\_certificate y ssl\_certificate\_key, cambiando las rutas de nuestro certificado y clave privada. (Pon la misma ruta y en server, por el nombre de tu servidor web)

```
server {
    listen      443 ssl http2 default_server;
    listen      [::]:443 ssl http2 default_server;
    server_name  _;
    root         /usr/share/nginx/html;

    → ssl_certificate "/etc/pki/nginx/server.crt";
    → ssl_certificate_key "/etc/pki/nginx/private/server.key";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout 10m;
    ssl_ciphers PROFILE=SYSTEM;
    ssl_prefer_server_ciphers on;
}
```

4. Quedará de la siguiente forma. Lo guardas.

```
server {
    listen      443 ssl http2 default_server;
    listen      [::]:443 ssl http2 default_server;
    server_name _;
    root         /usr/share/nginx/html;

    ssl_certificate "/etc/pki/tls/certs/avilesdiana.localdomain.crt";
    ssl_certificate_key "/etc/pki/tls/private/avilesdiana.localdomain.key";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout 10m;
    ssl_ciphers PROFILE=SYSTEM;
    ssl_prefer_server_ciphers on;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
    }
```

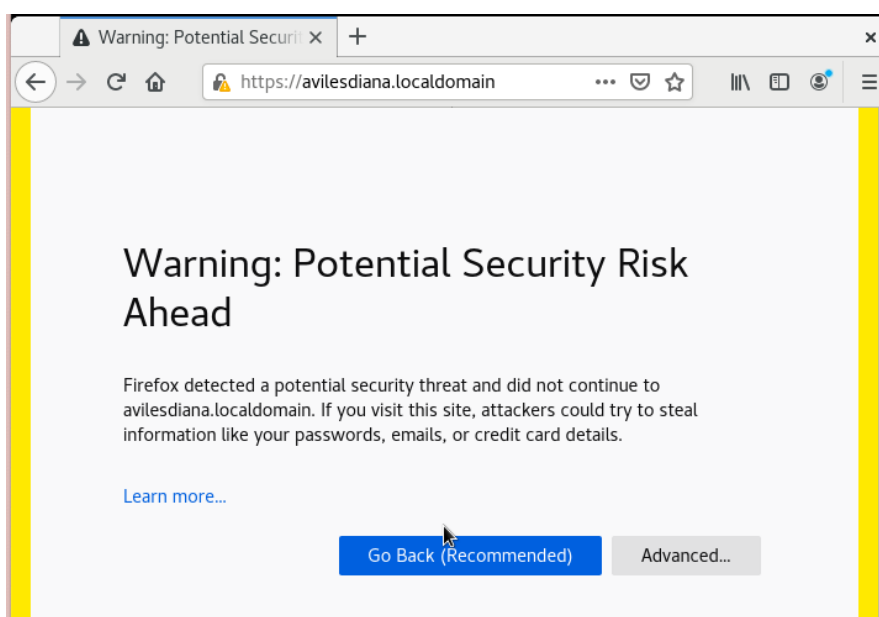
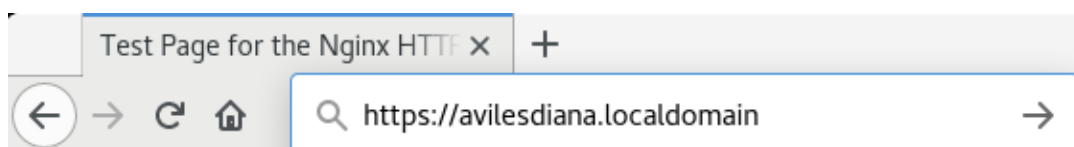
5. Ahora generamos un certificado autofirmado con el comando openssl de la siguiente forma, donde pondrás las mismas rutas que se establecieron anteriormente:

```
[root@avilesdiana ~]# sudo openssl req -newkey rsa:2048 -x509 -nodes -days 3650 -out /etc/pki/tls/certs/avilesdiana.localdomain.crt -keyout /etc/pki/tls/private/avilesdiana.localdomain.key
```

6. Al ejecutarlo se pedirán los siguientes datos, en la antepenúltima parte tienes que poner el nombre de tu servidor, aquí está el ejemplo:

```
[root@avilesdiana ~]# sudo openssl req -newkey rsa:2048 -x509 -nodes -days 3650 -out /etc/pki/tls/certs/avilesdiana.localdomain.crt -keyout /etc/pki/tls/private/avilesdiana.localdomain.key
Generating a RSA private key
.....+++++
writing new private key to '/etc/pki/tls/private/avilesdiana.localdomain.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:MX
State or Province Name (full name) []:edomex
Locality Name (eg, city) [Default City]:ecatepec
Organization Name (eg, company) [Default Company Ltd]:.
Organizational Unit Name (eg, section) []:.
Common Name (eg, your name or your server's hostname) []:avilesdiana.localdomain
Email Address []:avilesdiana@outlook.com
```

7. ¡Listo! Al guardar los cambios, reinicia nginx, accedemos a la web por defecto con el protocolo HTTP Y HTTPS para verificarlos:



## Configurar conexión HTTP v2.

Por lo general la trae por default, pero verificaremos si nuestra versión cuenta con HTTP/2 implementado.

1. Ingresamos de nuevo al archivo de configuración

```
[avilesdiana@avilesdiana ~]$ nano etc/nginx/nginx.conf
```

2. Al dirigirnos a la sección server, en la línea listen verificamos si está activada, en este caso lo está, pero si no solo agrega *http2* como se muestra a continuación:

```
GNU nano 2.9.8 /etc/nginx/nginx.conf

server {
    listen      443 ssl http2 default_server;
    listen      [::]:443 ssl http2 default_server;
    server_name _;
    root        /usr/share/nginx/html;

    ssl_certificate "/etc/pki/tls/certs/avilesdiana.localdomain.crt";
    ssl_certificate_key "/etc/pki/tls/private/avilesdiana.localdomain.key";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout 10m;
    ssl_ciphers PROFILE=SYSTEM;
    ssl_prefer_server_ciphers on;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
    }
}
```

3. Al guardar, verificamos si está activado el bloque con el siguiente comando:

```
[avilesdiana@avilesdiana ~]$ nginx -V
```

4. Nos mostrará el siguiente resultado y verificamos que este el activo el módulo **`--with-http_v2_module`**

```
avilesdiana@avilesdiana:~  
Archivo Editar Ver Buscar Terminal Ayuda  
built by gcc 8.2.1 20180905 (Red Hat 8.2.1-3) (GCC)  
built with OpenSSL 1.1.1 FIPS 11 Sep 2018 (running with OpenSSL 1.1.1g FIPS 2  
1 Apr 2020)  
TLS SNI support enabled  
configure arguments: --prefix=/usr/share/nginx --sbin-path=/usr/sbin/nginx --mo  
dules-path=/usr/lib64/nginx/modules --conf-path=/etc/nginx/nginx.conf --error-l  
og-path=/var/log/nginx/error.log --http-log-path=/var/log/nginx/access.log --ht  
tp-client-body-temp-path=/var/lib/nginx/tmp/client_body --http-proxy-temp-path=  
/var/lib/nginx/tmp/proxy --http-fastcgi-temp-path=/var/lib/nginx/tmp/fastcgi --  
http-uwsgi-temp-path=/var/lib/nginx/tmp/uwsgi --http-scgi-temp-path=/var/lib/ng  
inx/tmp/scgi --pid-path=/run/nginx.pid --lock-path=/run/lock/subsys/nginx --use  
r=nginx --group=nginx --with-file-aio --with-ipv6 --with-http_ssl_module --with  
-http_v2_module --with-http_realip_module --with-http_addition_module --with-ht  
tp_xslt_module=dynamic --with-http_image_filter_module=dynamic --with-http_sub  
module --with-http_dav_module --with-http_flv_module --with-http_mp4_module --w  
ith-http_gunzip_module --with-http_gzip_static_module --with-http_random_index  
module --with-http_secure_link_module --with-http_degradation_module --with-htt  
p_slice_module --with-http_stub_status_module --with-http_perl_module=dynamic -  
with-http_auth_request_module --with-mail=dynamic --with-mail_ssl_module --wit  
h-pcre --with-pcre-jit --with-stream=dynamic --with-stream_ssl_module --with-de  
bug --with-cc-opt='-O2 -g -pipe -Wall -Werror=format-security -Wp,-D_FORTIFY_SO  
URCE=2 -Wp,-D_GLIBCXX_ASSERTIONS -fexceptions -fstack-protector-strong -grecor  
d -gcc-switches -specs=/usr/lib/rpm/redhat/redhat-hardened-cc1 -specs=/usr/lib/rp  
m/redhat/redhat-annobin-cc1 -m64 -mtune=generic -fasynchronous-unwind-tables -f  
stack-clash-protection -fcf-protection' --with-ld-opt='-Wl,-z,relro -Wl,-z,now  
-specs=/usr/lib/rpm/redhat/redhat-hardened-ld -Wl,-E'  
[avilesdiana@avilesdiana ~]$
```

5. ¡Listo! Reiniciamos nginx, y para verificar su funcionamiento ingresamos el siguiente comando, si te regresa un 2 es que está utilizando HTTP/2, si no te devuelve 1.1.

```
[avilesdiana@avilesdiana ~]$ curl -sI https://rm-rf.es -o/dev/null -w '%{http_version}\n'  
2  
[avilesdiana@avilesdiana ~]$
```

## Conclusiones

Lo visto en esta práctica nos pareció muy interesante y útil, aún más cuando estamos tan cerca de salir como egresado de la carrera de Ingeniería en Sistemas de Telecomunicaciones y Sistemas de Cómputo en la IBERO. Esto porque, aunque ya todos conocíamos un poco de NGINX, PHP, MARIADB y HTTP/2, su implementación era un campo desconocido para nosotros. Al haber elaborado esta práctica, creemos que realizar este tipo de actividades nos van a preparar mucho mejor para poder convertirnos en desarrolladores web profesionales. La práctica requirió investigación profunda, buenas prácticas de documentación y organización por parte del equipo. Estos tres factores los consideramos extremadamente importantes aprenderlos en prácticas como estas, debido a que creemos todos que es lo que se busca principalmente en desarrolladores web y de software.

El haber implementado todos estos servicios dentro de una máquina virtual junto con la ayuda de CentOS como sistema operativo creemos que nos ofreció una mejor percepción de lo que estos servicios pueden ofrecer. NGINX básicamente a partir de nada nos ofrece servicios como http-https junto con ciertos protocolos de seguridad para así poder empezar a trabajar en páginas web. MARIADB sabemos que es muy similar a MySQL (derivado del mismo), por lo conocíamos previamente su funcionamiento básico. Aunque su instalación fue algo relativamente sencillo a diferencia de otros, nos hizo percatarnos de la importancia e implicaciones que pueden haber en un error a la hora de implementarlo equívocamente. PHP, por el otro lado, requirió un poco más de trabajo debido a que teníamos que hacer que funcione junto con NGINX. Finalmente, gracias a la investigación previa que tuvimos sobre HTTP, esta nos ayudó a conceptualizar mejor lo que la implementación significaba para la práctica.

Claramente el conocimiento principal que nos dio la elaboración de esta práctica, fue el conocer todas las opciones que existen. Ahora conocemos alternativas a los clásicos Windows/Mac/Linux, Apache, MySQL básico, entre otras. Gracias al trabajo realizado, ya tenemos un panorama y un entorno de desarrollo mucho más amplio que antes, lo que nos permitirá tomar decisiones más informadas para llevar a cabo un trabajo de desarrollo mucho más eficiente para el sistema que deseemos implementar. En general, creemos que esta práctica fue muy interesante porque nos da mucho que aprender sobre conceptos y herramientas que son necesarias en la vida profesional de cualquier desarrollador web.



## Referencias

1. Cómo instalar CentOS 8 en VirtualBox. (2021, 12 enero). Cómo instalar... <https://comoinstalar.me/como-instalar-centos-8-en-virtualbox/>
2. Heidi, E. (2020, 30 septiembre). *Cómo instalar Nginx en CentOS 8*. DigitalOcean. <https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-centos-8-es>
3. ¿Qué es un Servidor virtual y cuáles son sus características? (s. f.). beServices. Recuperado 30 de enero de 2021, de <https://www.beservices.es/que-un-servidor-virtual-caracteristicas-n-5392-es>
4. Servidores físicos vs. Máquinas virtuales. (s. f.). [Gráfico]. VEEAM. <https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.veeam.com%2Fblog%2Fes-lat%2Fwhy-virtual-machine-backups-different.html&psig=AOvVaw2c9wiU2FEgrk-ksxQ2QMAC&ust=1612132467319000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCLDP6szbxO4CFQAAAAAdAAAAABAD>
5. Boucheron, B. (2021, 18 enero). *Cómo instalar MariaDB en CentOS 8*. DigitalOcean. <https://www.digitalocean.com/community/tutorials/how-to-install-mariadb-on-centos-8-es>
6. A. (2019, 28 abril). *Cómo activar HTTP/2 en Nginx*. # rm-rf.es. <https://rm-rf.es/como-activar-http-2-nginx/>
7. Ghedini, A. (2019, 17 octubre). *Experiment with HTTP/3 using NGINX and quiche*. The Cloudflare Blog. <https://blog.cloudflare.com/experiment-with-http-3-using-nginx-and-quiche/>
8. Heidi, E. (2020, 1 septiembre). *Cómo instalar la pila Linux, Nginx, MySQL, PHP (LEMP) en CentOS 8*. DigitalOcean. <https://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mysql-php-lemp-stack-on-centos-8-es>
9. Instalador, E. (2020, 16 junio). *Cómo instalar Nginx en CentOS 7*. Cómo instalar... <https://comoinstalar.me/como-instalar-nginx-en-centos-7/>
10. Kumar, C. (2019, 10 marzo). *¿Cómo habilitar HTTP / 2 en Apache y Nginx?* Geekflare. <https://geekflare.com/es/http2-implementation-apache-nginx/>
11. Kothekar, S. (2021, 6 enero). *MariaDB vs MySQL*. EDUCBA. <https://www.educba.com/mariadb-vs-mysql/>
12. Rungta, K. (2021, 16 enero). *MariaDB vs MySQL: Key Performance Differences*. Guru99. <https://www.guru99.com/mariadb-vs-mysql.html>
13. Goel, A., Morel, B., M., & WebTechLabs. (s. f.). *MariaDB vs MySQL: [2021] Everything You Need to Know*. Hackr.io. Recuperado 10 de febrero de 2021, de <https://hackr.io/blog/mariadb-vs-mysql>
14. Auladell, G. (2019, 26 noviembre). *¿Qué es NGINX?* Drauta. <https://www.drauta.com/que-es-nginx>
15. Jankov, T. (2021, 28 enero). *Nginx vs Apache: Web Server Showdown*. Kinsta. <https://kinsta.com/blog/nginx-vs-apache/>
16. Fontela, A. (2018, 24 abril). *Nginx: ventajas y desventajas*. Raiola Networks. <https://raiolanetworks.es/blog/nginx/>
17. U. (2020, 20 mayo). *¿Qué ventajas tiene el protocolo HTTPS para tu web?* UpToBe. <https://uptobemarketing.com/protocolo-https-ventajas/>
18. Angulo, J. (2016, 8 febrero). *¿Qué es HTTP/2 y qué ventajas tiene sobre HTTP 1.1 ?* SomosTechies. <https://somostechies.com/que-es-http2/>

# Anexos

## Administrar el proceso de Nginx

Ahora que su servidor web funciona, revisaremos la manera de administrar el servicio de Nginx a través de systemctl.

- Siempre que deba detener su servidor web, puede usar lo siguiente:  
`sudo systemctl stop nginx`
- Para iniciar el servidor web cuando no esté activo, escriba lo siguiente:  
`sudo systemctl start nginx`
- Para detener y luego iniciar el servicio de nuevo, puede usar lo siguiente:  
`sudo systemctl restart nginx`
- Nginx puede volver a cargar los cambios en la configuración sin cerrar las conexiones. Para hacer esto, escriba lo siguiente:  
`sudo systemctl reload nginx`
- De forma predeterminada, Nginx está configurado para iniciarse automáticamente cuando lo haga el servidor. Si no es lo que quiere, deshabilitar este comportamiento escribiendo lo siguiente:  
`sudo systemctl disable nginx`
- Para volver a habilitar el servicio y hacer que Nginx se inicie de nuevo en el arranque, puede usar lo siguiente:  
`sudo systemctl enable nginx`

## Solución Error Mariadb

### Ingresar en la consola:

1. `yum remove mariadb mariadb-server`
2. `rm -rf /var/lib/mysql`
3. `rm /etc/my.cnf`
4. Vuelves a instalar Mariadb `yum install mariadb-server`

### Más información:

<https://stackoverflow.com/questions/33362904/completely-remove-mariadb-or-mysql-from-centos-7-or-rhel-7>