

MANUAL TÉCNICO DE INSTALACIÓN INVENTARIO GASTRONÓMICO SENA

(IGS)





Tabla De Contenido

Objetivos	3
Objetivo general	3
Objetivos específicos.....	3
Introducción	4
Requisitos Técnicos Mínimos De Hardware	5
Requisitos Técnicos Mínimos De Software.....	5
Herramientas Utilizadas Para La Instalación Y Ejecución Del Aplicativo	6
Instalación De Ubuntu Server	7
Trabajar en una máquina externa	16
Instalación de Apache	17
Importación Gestor De Base De Datos.....	20
Instalación De Phpmyadmin	22
Importación De Base De Datos	25
Conexión A FileZilla	28
Ejecutar Backend.....	32
IP Estática	37
Ejecutar Frontend.....	40



Objetivos

Objetivo general

Proporcionar al administrador la información necesaria para gestionar la plataforma, incluyendo los programas y herramientas utilizados en el desarrollo y configuración del aplicativo IGS.

Objetivos específicos

- Detallar los requisitos de hardware y software necesarios para instalar y operar el sistema en un ambiente adecuado.

- Describir las funcionalidades técnicas del aplicativo para una mejor comprensión del mismo.

- Indicar las herramientas empleadas en el desarrollo y diseño de la aplicación que fueron usadas para la culminación de la misma.



Introducción

El manejo adecuado de una aplicación depende en gran medida del conocimiento que tengamos sobre ella. Por esta razón, hemos creado este manual con el propósito de detallar los aspectos técnicos necesarios para la instalación y ejecución del sistema. Además, proporciona una visión fundamental de los requisitos mínimos para que el sistema funcione correctamente en un computador con las características indicadas.

IGS (Inventario Gastronómico SENA) se ha desarrollado con el objetivo de mejorar el sistema de inventario utilizado en el área de gastronomía. Este aplicativo permite gestionar la disponibilidad, entrada y salida de insumos en el entorno de formación gastronómica, facilitando un conteo preciso y optimizando el uso de recursos.

El manual está diseñado para ser utilizado por personas responsables de administrar, editar o modificar la aplicación, garantizando así su mantenimiento adecuado y la correcta gestión de los datos almacenados. Cabe mencionar que este proyecto fue implementado en el sistema operativo Ubuntu Server.



Requisitos Técnicos Mínimos De Hardware

- Equipo de cómputo:
 - RAM: 16 GB (recomendado 32 GB o más).
 - Almacenamiento: Mínimo 250 GB SSD (recomendado 500 GB o más).
 - Procesador: Intel Core i5-12400 o superior (recomendado Intel Xeon Silver 4210 o superior).

- Información técnica del servidor recomendado:
 - Procesador: Intel Xeon Gold 6254 (3.10 GHz) o superior.
 - Frecuencia: 3.10 GHz o superior.
 - RAM instalada: 64 GB o más.
 - Tipo de sistema: Sistema operativo de 64 bits, procesador basado en x64.
 - Sistema operativo: Linux Ubuntu Server.
 - Versión: 24.04 LTS (recomendado).
 - Disco duro: SSD NVMe de 1 TB o superior.

Requisitos Técnicos Mínimos De Software

Privilegios: Administrador

Sistema operativo:

- Windows 11 (recomendado)
- Linux (Ubuntu Server 24.04 LTS recomendado)
- macOS Monterey
- iOS
- Android

Navegadores de internet:

- Google Chrome (versión más reciente)
- Mozilla Firefox (versión más reciente)
- Microsoft Edge (versión más reciente)

Herramientas Utilizadas Para La Instalación Y Ejecución Del Aplicativo

- **Node.js:** Es un entorno controlado por eventos diseñado para crear aplicaciones escalables, permitiéndote establecer y gestionar múltiples conexiones al mismo tiempo. Gracias a esta característica, no tienes que preocuparte con el bloqueo de procesos, pues no hay bloqueos.



- **MySQL:** No es solo un controlador: Es una herramienta poderosa que permite conectarse, interactuar y controlar la base de datos MySQL con precisión y facilidad.



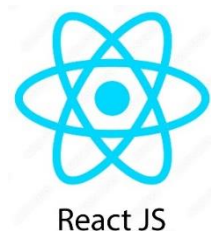
- **npm:** Es un gestor de paquetes para los proyectos de Node.js disponibles para su uso público. Los proyectos disponibles en el registro de npm se llaman «paquetes».



- **Apache:** Apache es un servidor web ampliamente utilizado. Actúa como intermediario entre el cliente (como un navegador web) y el servidor donde se aloja un sitio web. Su función principal es entregar páginas web al navegador del usuario. Apache es conocido por su robustez, flexibilidad y escalabilidad.

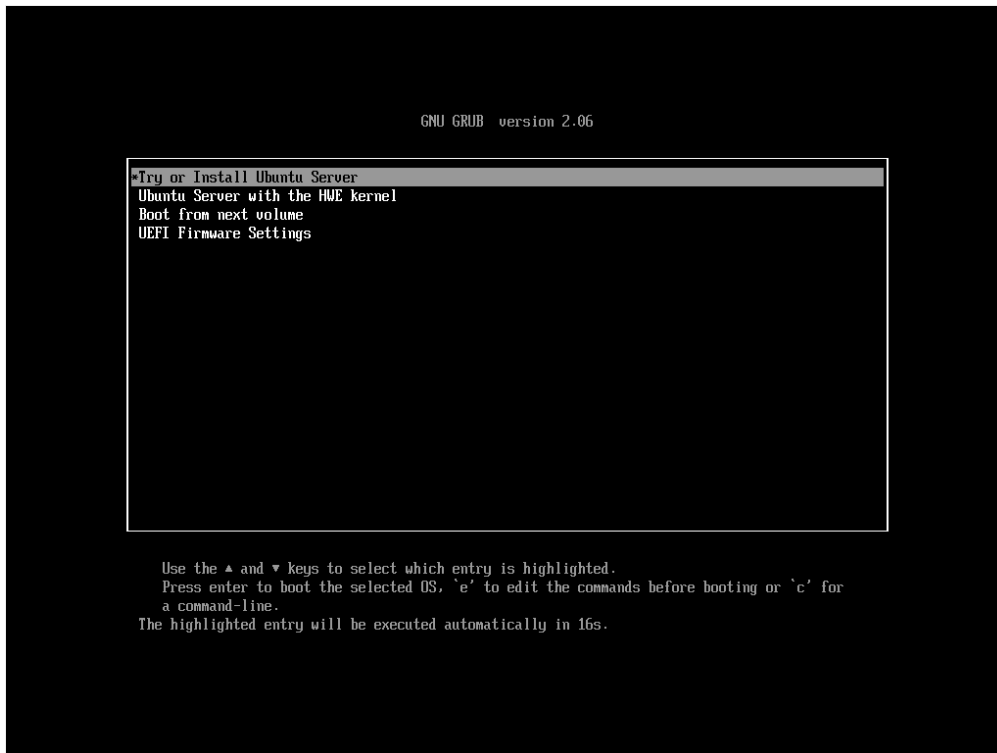


- **ReactJS:** Es una biblioteca de JavaScript ampliamente utilizada para construir interfaces de usuario interactivas. Permite crear componentes reutilizables y actualiza automáticamente la vista cuando los datos cambian.

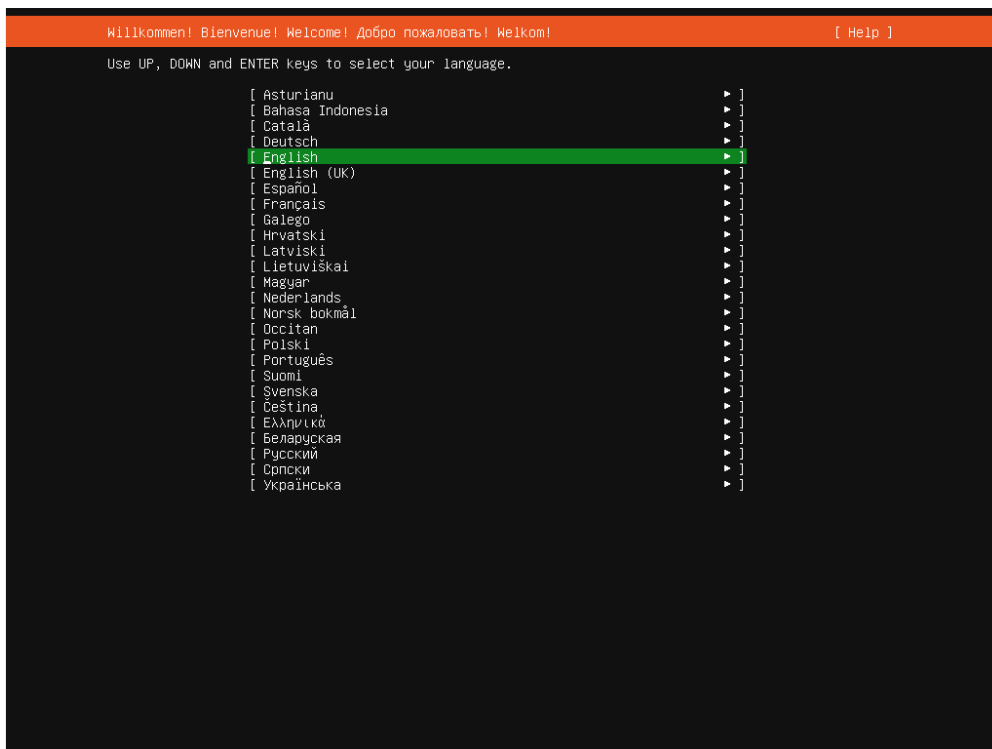


Instalación De Ubuntu Server

1. Pantalla principal de instalación: elegimos la primera opción para correr el instalador de Ubuntu server y presionamos enter.

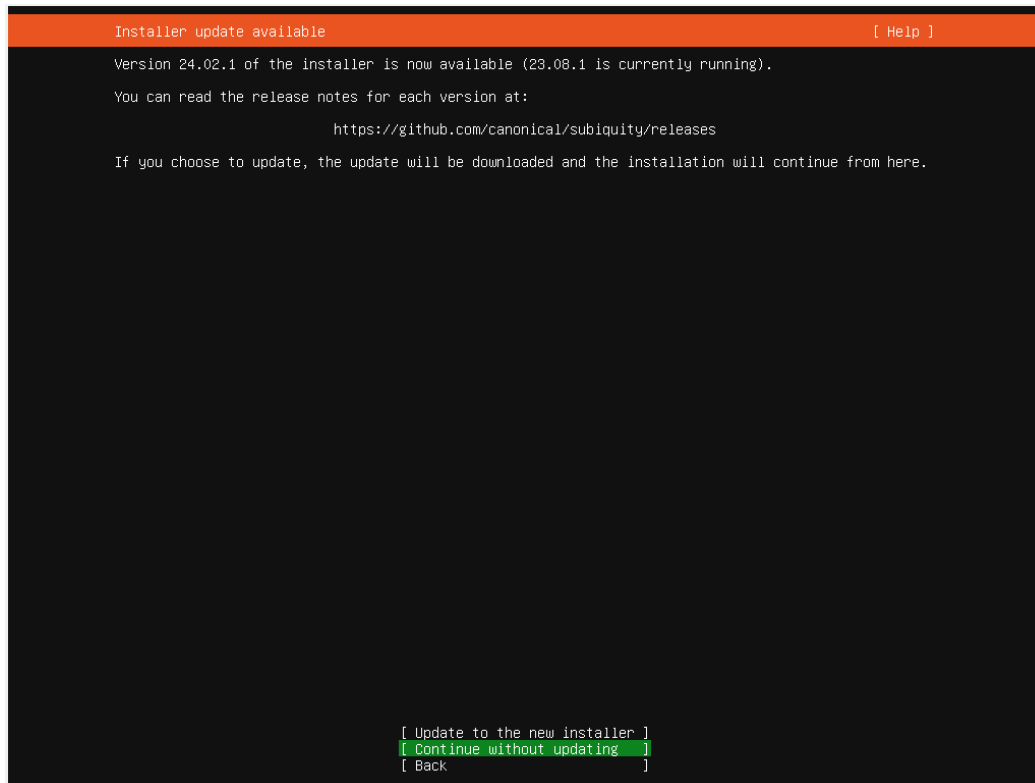


2. Se selecciona el idioma de instalación del sistema operativo que desea trabajar (en este caso selecciono ingles).

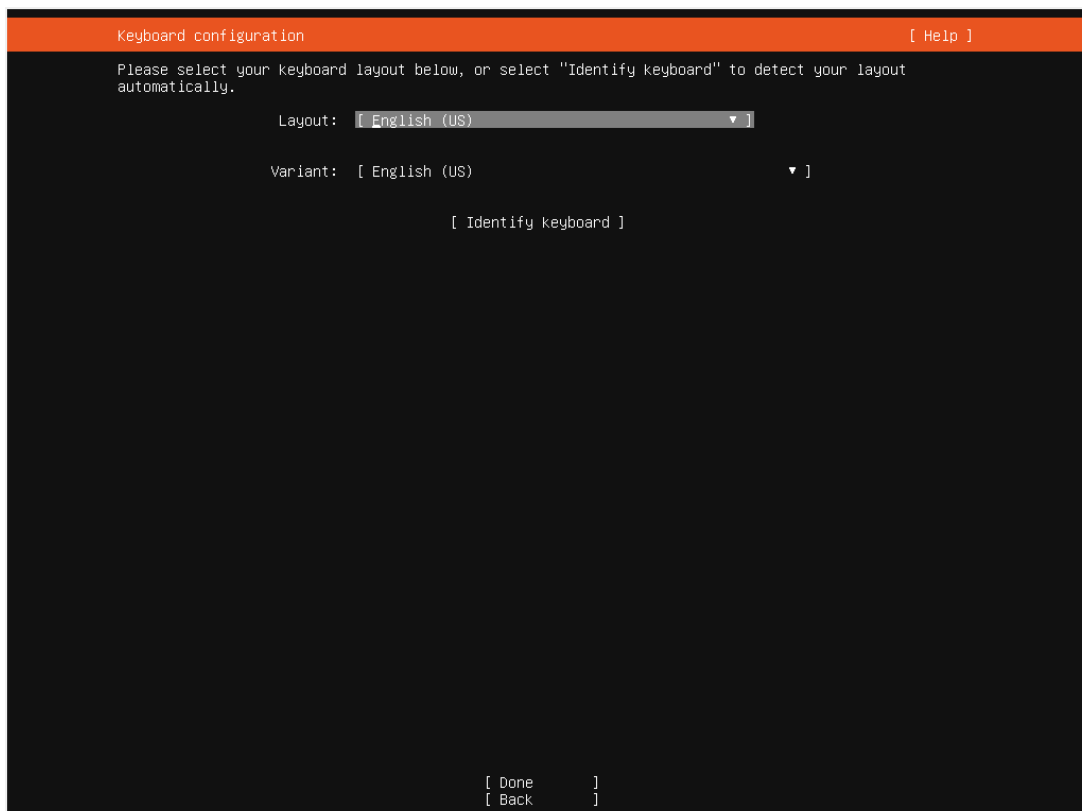




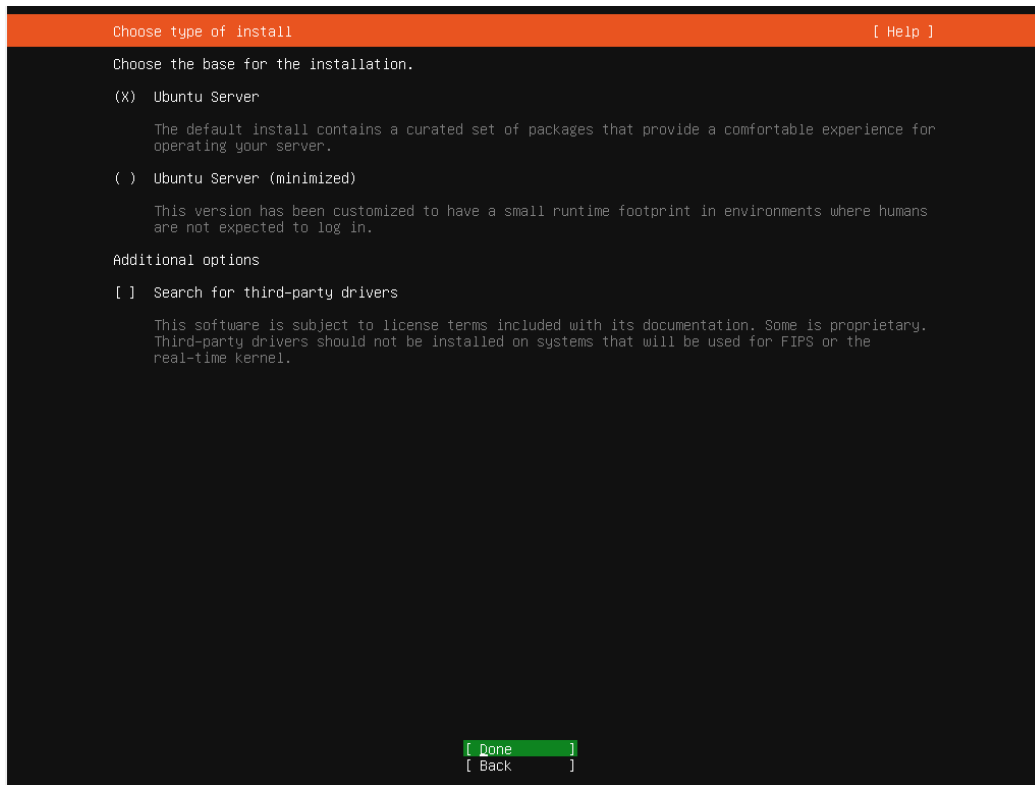
3. Aquí se ofrece una nueva versión de instalación, pero en este caso se omite y se selecciona continuar sin actualizar.



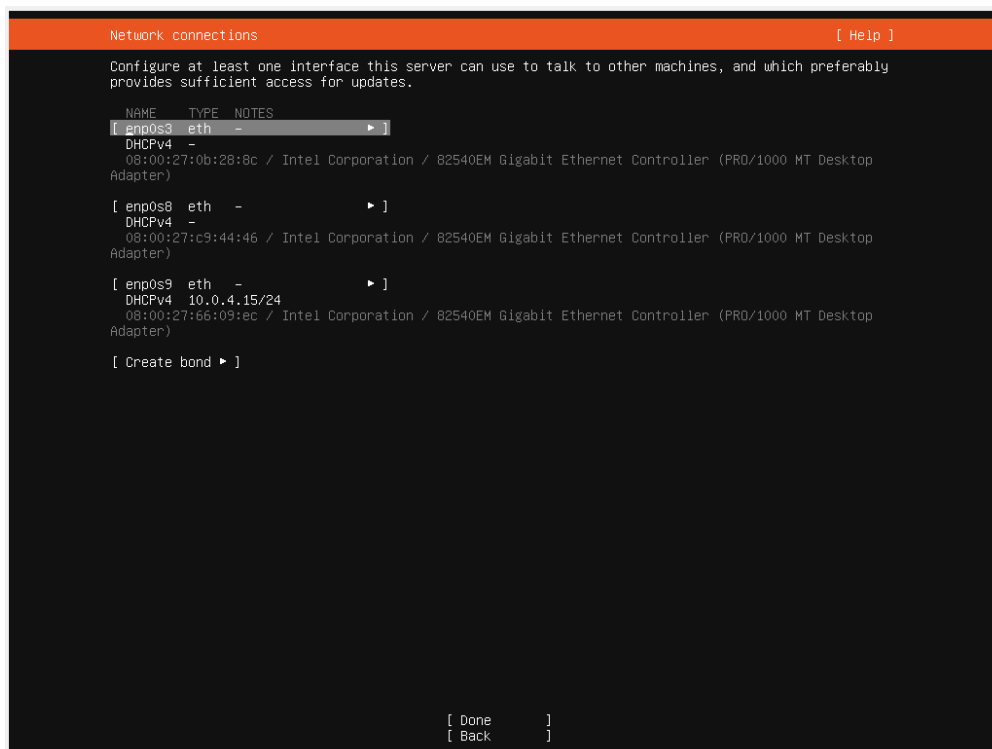
4. Se selecciona el idioma del teclado que se esté utilizando al momento de la instalación.



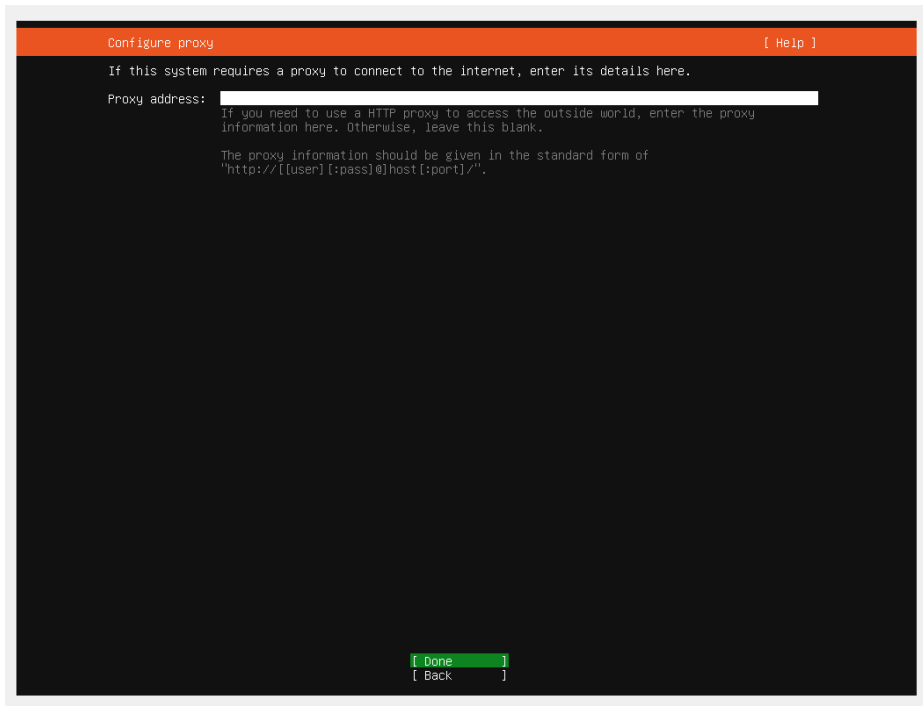
5. Se selecciona Ubuntu server con la tecla tab para su instalación completa y se selecciona “Done”.



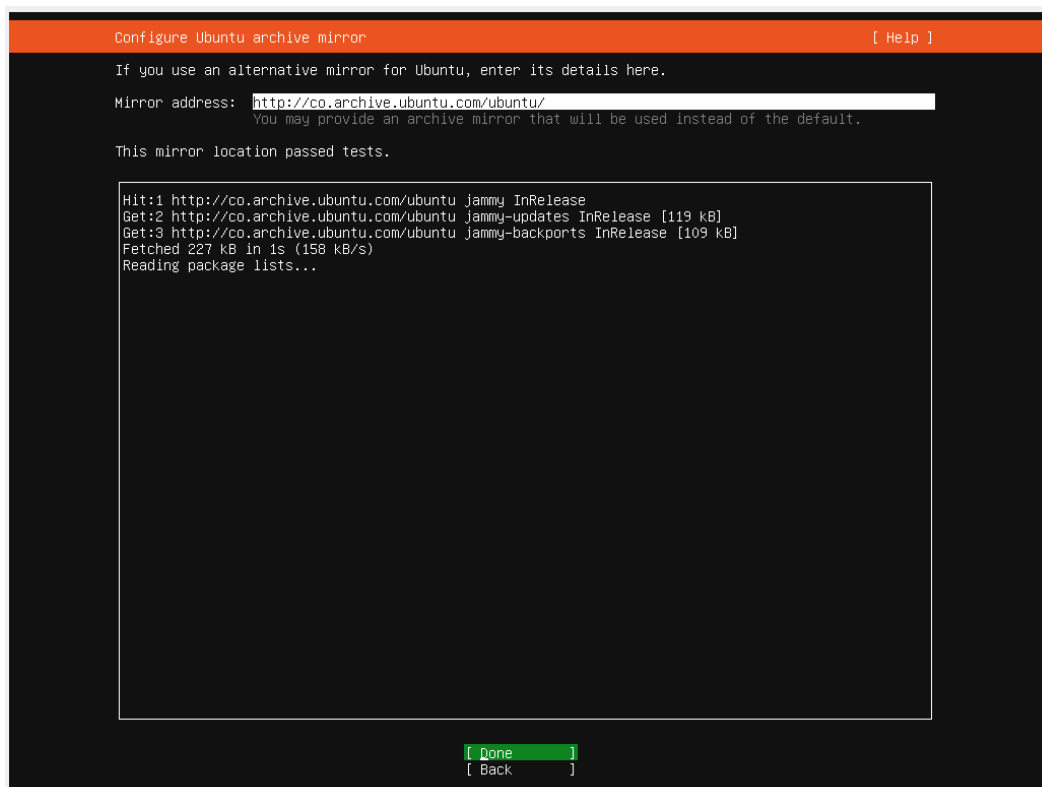
6. Aquí se puede observar los puertos por los cuales la maquina va a tener conexión con los clientes, en este caso se deja de manera dinámica y seleccionamos done.



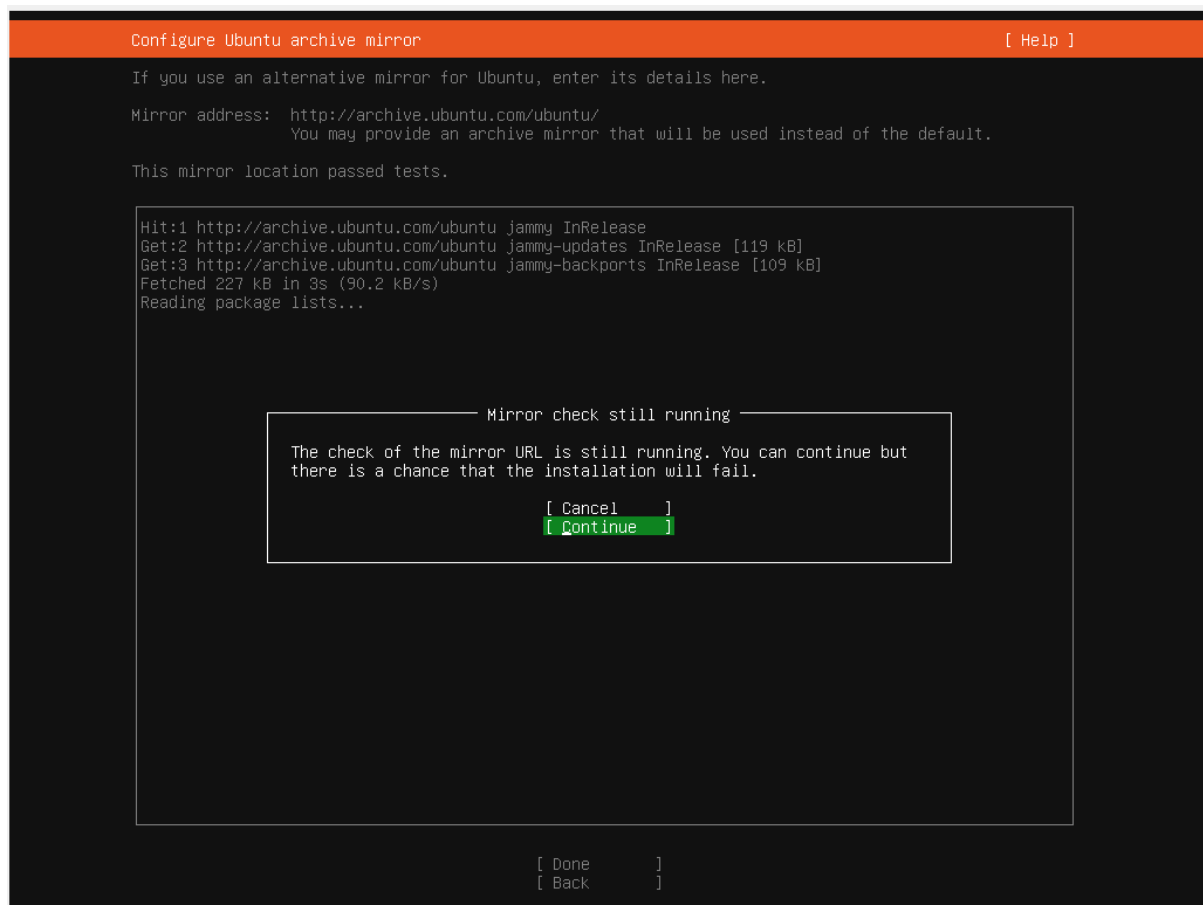
7. Usamos la tecla tab para ir hacia mirror address y se presiona la tecla enter y nos arrojará el siguiente proxy `http://co.archive.ubuntu.com/ubuntu/` luego presionamos enter y esperamos que se descarguen los archivos, una vez descargados presionamos la tecla tab hasta que “Done” esté seleccionado.



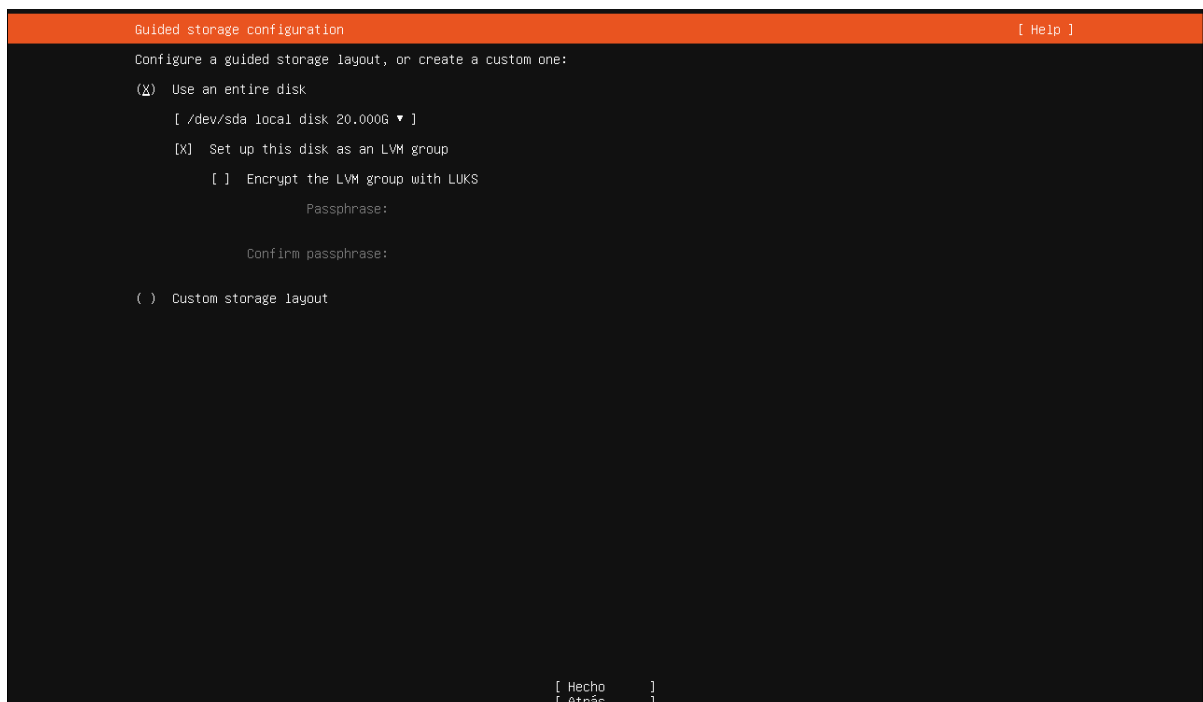
8. Se selecciona el repositorio de Ubuntu cual se desea operar en este caso dejamos `http://co.archive.ubuntu.com/ubuntu/` si desea usar el repositorio de estados unidos se elimina el “co”.



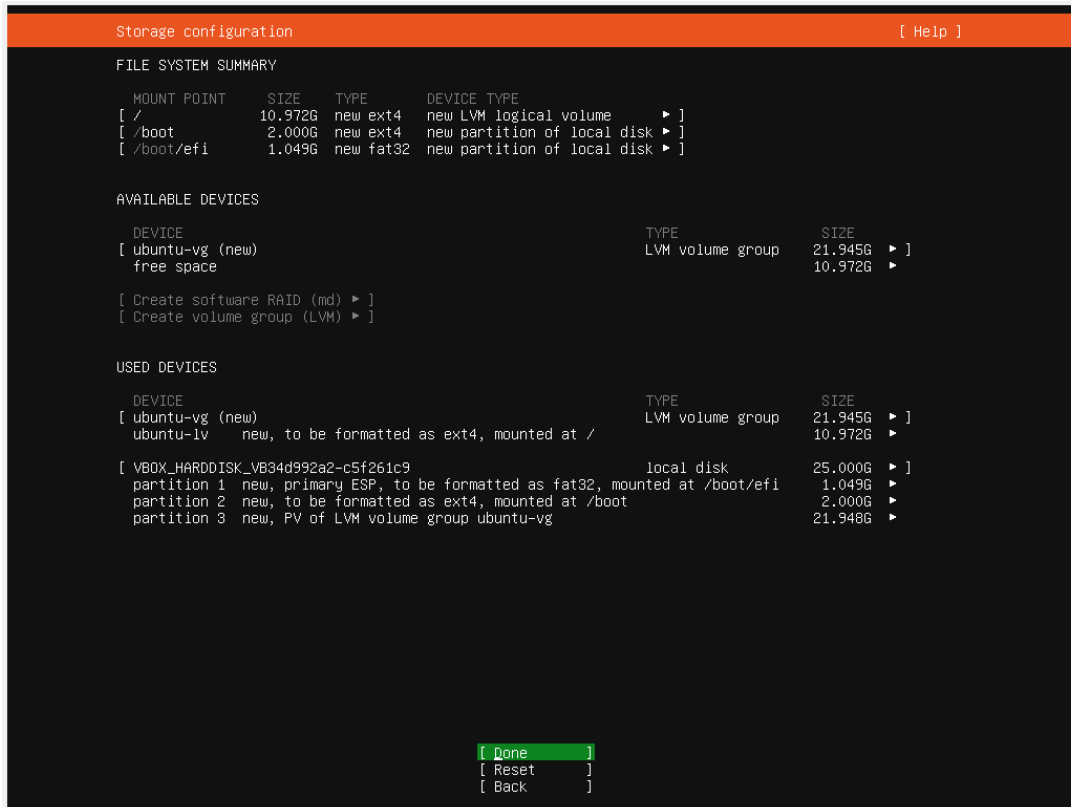
9. Confirmamos que el enlace del repositorio este correcto.



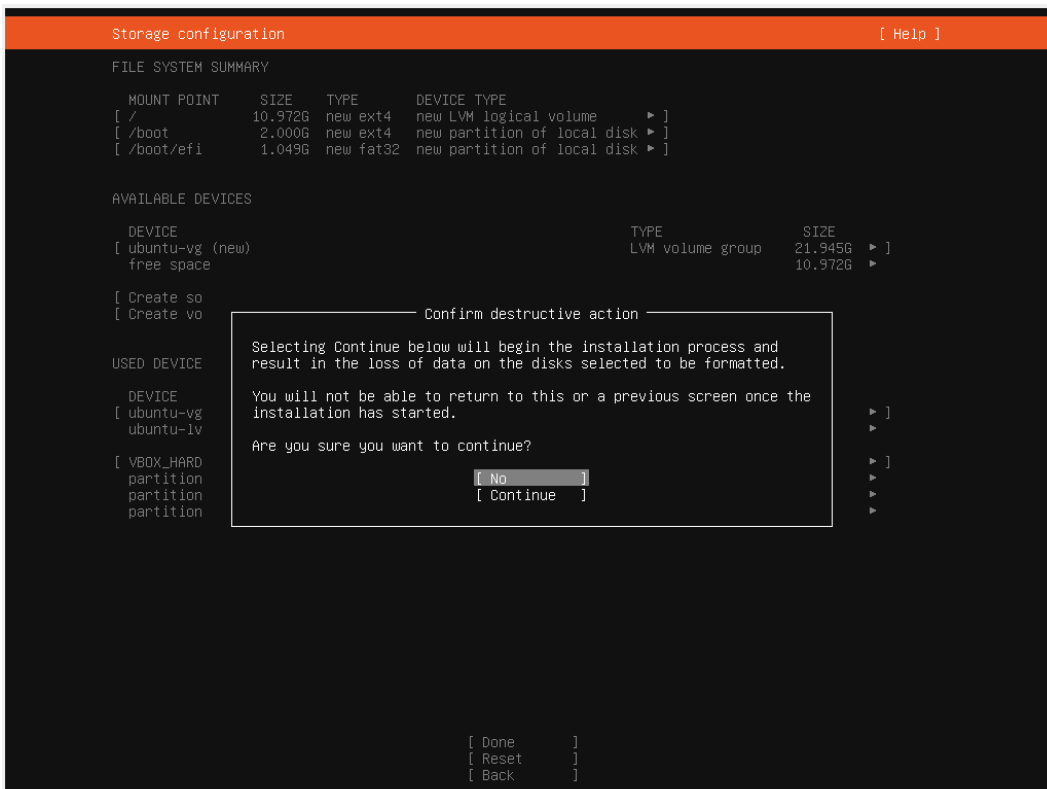
10. En esta parte dejamos las opciones predeterminadas y damos en “Done”



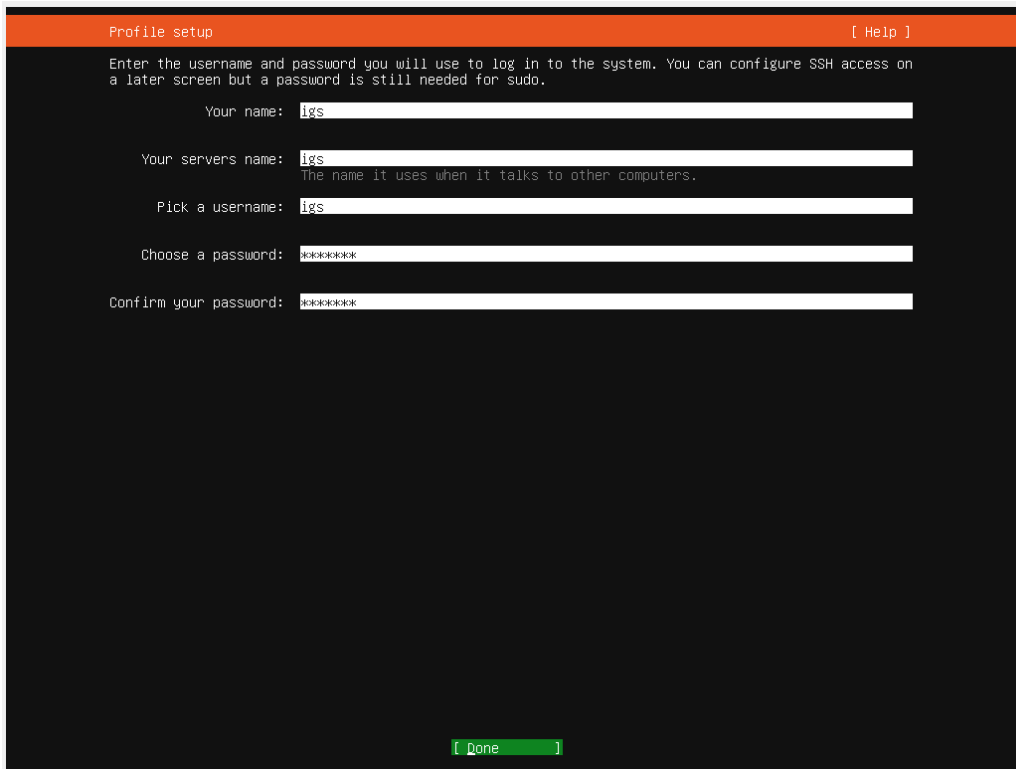
11. La configuración de almacenamiento se dejó como predeterminada y se selecciona done.



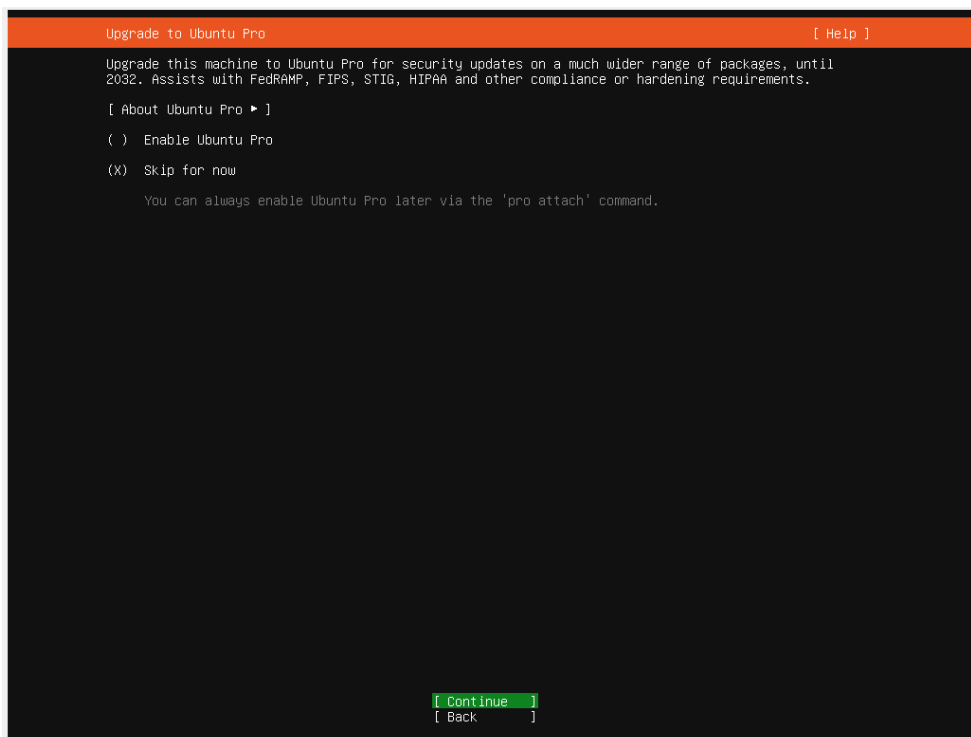
12. Confirmamos la selección que hicimos previamente y seleccionamos continue.



13. Registramos un nombre de usuario, nombre del servidor, y una contraseña para acceder al servidor.



14. Se ofrece una versión pro pero en este caso no la escogemos y seleccionamos “skip for now” presionado la tecla tab y posteriormente seleccionamos “continue”.



15. Se selecciona “install openssh serve” con la tecla espacio para las conexiones seguras, luego se presiona la tecla “tab” hasta llegar a “done” y presionamos enter.

```
SSH Setup [ Help ]

You can choose to install the OpenSSH server package to enable secure remote access to your server.

[X] Install OpenSSH server

Import SSH identity: [ No ]
You can import your SSH keys from GitHub or Launchpad.

Import Username:

[X] Allow password authentication over SSH

[ Done ]
[ Back ]
```

16. Se deja por defecto, presionamos la tecla tab hasta llegar a la opción de “done” y se presiona la tecla enter.

```
Featured Server Snaps [ Help ]

These are popular snaps in server environments. Select or deselect with SPACE, press ENTER to see
more details of the package, publisher and versions available.

[ ] microk8s canonical✓ Kubernetes for workstations and appliances
[ ] nextcloud nextcloud✓ Nextcloud Server - A safe home for all your data
[ ] wekan xet7 Open-Source kanban
[ ] kata-containers katacontainers✓ Build lightweight VMs that seamlessly plug into the
[ ] docker canonical✓ Docker container runtime
[ ] canonical-livepatch canonical✓ Canonical Livepatch Client
[ ] rocketchat-server rocketchat✓ Rocket.Chat server
[ ] mosquito mosquito✓ Eclipse Mosquitto MQTT broker
[ ] etcd canonical✓ Resilient key-value store by CoreOS
[ ] powershell microsoft-powershell✓ PowerShell for every system!
[ ] sabnzbd safihre SABnzbd
[ ] wormhole snapcrafters get things from one computer to another, safely
[ ] aws-cli aws✓ Universal Command Line Interface for Amazon Web Serv
[ ] google-cloud-sdk google-cloud-sdk✓ Google Cloud SDK
[ ] slcli softlayer Python based SoftLayer API Tool.
[ ] doctl digitalocean✓ The official DigitalOcean command line interface
[ ] conjure-up canonical✓ Package runtime for conjure-up spells
[ ] postgresql10 cmd✓ PostgreSQL is a powerful, open source object-relatio
[ ] heroku heroku✓ CLI client for Heroku
[ ] keepalived keepalived-project✓ High availability VRRP/BFD and load-balancing for Li
[ ] prometheus canonical✓ The Prometheus monitoring system and time series dat
[ ] juju canonical✓ Juju - a model-driven operator lifecycle manager for

[ Done ]
[ Back ]
```

17. Ahora comienza a descargarse los controladores del servidor. Cuando finalice damos enter sobre la opción reiniciar ahora.

```
Installing system [ Help ]

installing system
executing curtin install initial step
executing curtin install partitioning step
  curtin command install
    configuring storage
      running 'curtin block-meta simple'
      curtin command block-meta
        removing previous storage devices
        configuring disk: disk-sda
        configuring partition: partition-0
        configuring format: format-0
        configuring partition: partition-1
        configuring format: format-1
        configuring partition: partition-2
        configuring lvm_voigroup: lvm_voigroup-0
        configuring lvm_partition: lvm_partition-0
        configuring format: format-2
        configuring mount: mount-2
        configuring mount: mount-1
        configuring mount: mount-0
    executing curtin install extract step
      curtin command install
        writing install sources to disk
        running 'curtin extract'
        curtin command extract
          acquiring and extracting image from cp:///tmp/tmpqmcpgd748/mount
    configuring keyboard
    curtin command in-target
  executing curtin install curthooks step
    curtin command install
      configuring installed system
      running 'curtin curthooks'
      curtin command curthooks
        configuring apt configuring apt
        installing missing packages
        installing packages on target system: ['efibootmgr', 'grub-efi-amd64',
'grub-efi-amd64-signed', 'shim-signed']
        configuring iscsi service
        configuring raid (mdadm) service
        installing kernel \

[ View full log ]
```

18. Una vez reiniciado nos pide ingresar un nombre de usuario en este caso ingresamos igs y presionamos enter, luego ingresamos una contraseña (no se verá que se ingresa, pero si se está ingresando) asignada.

```
ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBDD0uW0uf4wIaz1C43QR06hroU6Dw+GmB64NogKtufLLju42dsq3V9Jf
V8zPy52zbbkSDQofB48j+GCzSajbDzk= root@igs
ssh-ed25519 AAAAC3NzaC11ZDI1NTE5AAAAILXMG2y9E2xLh0FcGLr9i0Y5cGKj77E24gSPt3RjEds6 root@igs
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQDeKggzVQ70c158nDqnZ26s12c0uM0Fa1gM7hge0wnfPiB2LLS1kIDM0ho01prAn0Uz5fFuy8wsFyW7sLiALE14D2ZL
6x+607XFzcEblDB11NPdyArxyTumD2zgXfr9x4e4GKkJ1SiGpsn9UyilgIssSkz31u1gPNrW9dQDN0e2k800UbiM215gwwF023s+yyMxs150yVs5m8h84P4IvT6Nkf
UwFOX0Ha0s5R9vUNOeegASRXwUmNrkhaY1PnscYQPfT7nPTQn6tBW01Y0d20HyxS2VNMIFLsz8p6PoJ7UVssAszGzsaHYx81cVY1Ec1ycvc2A+9vK53GhmXBbeP2Pad1
MSGfURS8c1Ea7WgJytqcJ4L60hV7mG6dAKHMJ9YqteUc2GTSGjzskJQf/+208K0AQ84YXtMIb204cA4I0R+nuUgviZQtMU0BVybNqmK728kofHXXPdBFQXbzNj46pU
wTQA4fgQMps7LstYBp5KchUKJxe86gE51B2UEo8= root@igs
-----END SSH HOST KEY KEYS-----
[ 135.489628] cloud-init[1469]: Cloud-init v. 23.2.1-0ubuntu0~22.04.1 finished at Tue, 27 Feb 2024 13:13:22 +0000. DataSource D
ataSourceNone. Up 135.48 seconds
[ 135.489941] cloud-init[1469]: 2024-02-27 13:13:22,448 - cc_final_message.py[WARNING]: Used fallback datasource
igs
Password:
```

19. Ahora ingresamos el comando "ip address" el cual nos arrojará la dirección IP de nuestro servidor en nuestra red local, ya nuestro servidor está montado, sin ninguna dependencia instalada

```
igs@igs:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:cb:de:bc brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.0.105/24 metric 100 brd 192.168.0.255 scope global dynamic ens33
        valid_lft 7112sec preferred_lft 7112sec
    inet6 fe80::20c:29ff:feeb:debc/64 scope link
        valid_lft forever preferred_lft forever
```

Trabajar en una máquina externa

Ya que se tiene el servidor montado, podemos trabajar con él desde otro computador conectado a la misma red de internet para mayor comodidad en su configuración, en este caso se usará un computador con Windows, para ellos se procede a realizar lo siguiente:

1. Abrimos un terminal, ya sea cmd o Windows PowerShell e ingresamos el comando ssh (en caso de estar en Linux no es necesario escribir ssh) espacio nombre de usuario seguido de una @ y la dirección IP, presionamos enter y nos preguntará si queremos continuar, escribimos "yes":

```
ssh igs@192.168.0.105
```

```
C:\Users\ander>ssh igs@192.168.0.105
The authenticity of host '192.168.0.105 (192.168.0.105)' can't be established.
ED25519 key fingerprint is SHA256:BbKskeZb3EUOXfMUomaYOXyW8924t0fxKe6Q6xKnAyM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.105' (ED25519) to the list of known hosts.
```

2. Nos pide ingresar la contraseña que previamente establecimos (paso # 13 de la anterior sección) la digitamos (no se verá porque Ubuntu server no la muestra, pero se está digitando internamente), presionamos enter y veremos un mensaje como el siguiente donde al final en letras verdes se verá el nombre del servidor (igs) seguido de una @ y luego el nombre de usuario (igs) seguido de dos puntos un ~ y \$:

```
igs@192.168.0.105's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-97-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of mar 05 mar 2024 21:02:13 UTC

System load:  0.169921875   Processes:    210
Usage of /:   47.9% of 9.75GB Users logged in:    1
Memory usage: 18%          IPv4 address for ens33: 192.168.0.105
Swap usage:   0%

El mantenimiento de seguridad expandido para Applications está desactivado
Se pueden aplicar 58 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Mar 5 20:48:50 2024
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

igs@igs:~$
```

3. Para corroborar que accedimos exitosamente ingresamos el comando "ip address" y nos arrojará la misma dirección ip que se obtuvo estando en nuestro equipo donde está instalado Ubuntu server:

```
igs@igs:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:cb:de:bc brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.0.105/24 metric 100 brd 192.168.0.255 scope global dynamic ens33
        valid_lft 6166sec preferred_lft 6166sec
    inet6 fe80::20c:29ff:febc:debc/64 scope link
        valid_lft forever preferred_lft forever
```

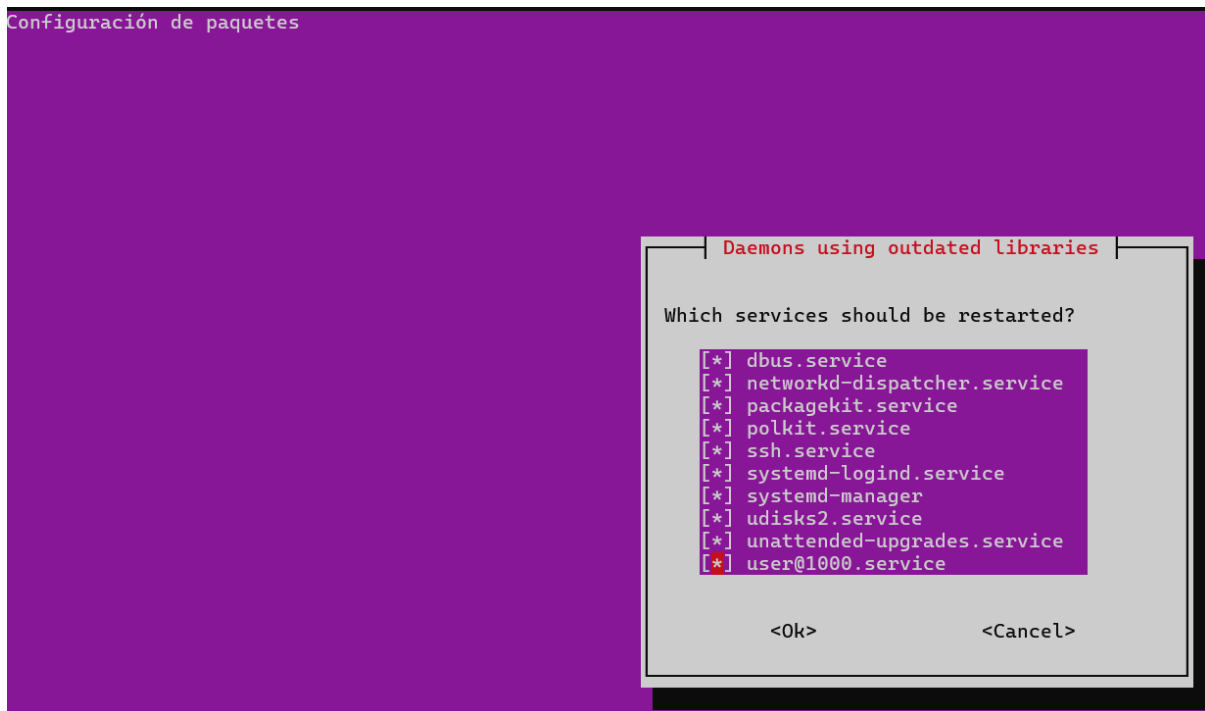

Instalación de Apache

1. Una vez indicada sesión lo primero que se hace es ingresar el siguiente comando el cual lista y posteriormente actualiza los cambios más recientes del repositorio luego presionamos enter y nos pide la contraseña la ingresamos (no se verá que se ingresa, pero si se está ingresando) y presionamos enter nuevamente:

```
sudo apt update && sudo apt upgrade -y
```

```
igs@igs:~$ sudo apt update && sudo apt upgrade -y
[sudo] password for igs:
```

2. Se selecciona todos servicios para tener un mejor rendimiento para ello nos movemos con las teclas flecha arriba y abajo, importante los marcamos todos con la tecla espacio espacio y se remarcaran con asterisco, finalmente seleccionamos ok y presionamos la tecla enter.



3. Ingresamos los comandos por separado

```
curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash - &&\
```

```
sudo apt-get install -y nodejs
```

En resumen, estos comandos descargan e instalan Node.js en el sistema utilizando un script descargado de https://deb.nodesource.com/setup_20.x, todo ello con permisos de superusuario y sin necesidad de intervención del usuario para confirmar la instalación.

```
igs@igs:~$ curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash - &&\
```

```
igs@igs:~$ sudo apt-get install -y nodejs
```

4. Para confirmar la versión de node, escribimos el siguiente comando en la consola:

```
node -v
```

```
igs@igs:~$ node -v
v12.22.9
```

5. Se actualiza la lista de paquetes disponibles en los repositorios configurados en el sistema ingresando el siguiente comando:

```
sudo apt-get update
```

```
igs@igs:~$ sudo apt-get update
Obj:1 http://co.archive.ubuntu.com/ubuntu jammy InRelease
Ign:2 https://deb.nodesource.com/node_20.x nodistro InRelease
Obj:3 http://co.archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:4 http://co.archive.ubuntu.com/ubuntu jammy-backports InRelease
Obj:5 http://co.archive.ubuntu.com/ubuntu jammy-security InRelease
Ign:2 https://deb.nodesource.com/node_20.x nodistro InRelease
Ign:2 https://deb.nodesource.com/node_20.x nodistro InRelease
Err:2 https://deb.nodesource.com/node_20.x nodistro InRelease
  Invalid response from proxy: HTTP/1.1 400 Bad Request Server: nginx/1.20.1 Date: Tue, 05 Mar 2024 21:
  29:46 GMT Content-Type: text/html Content-Length: 157 Connection: close [IP: 200.25.7.49 443]
Leyendo lista de paquetes... Hecho
W: Fallo al obtener https://deb.nodesource.com/node_20.x/dists/nodistro/InRelease Invalid response from
  proxy: HTTP/1.1 400 Bad Request Server: nginx/1.20.1 Date: Tue, 05 Mar 2024 21:29:46 GMT Content-Type:
  text/html Content-Length: 157 Connection: close [IP: 200.25.7.49 443]
W: No se han podido descargar algunos archivos de índice, se han omitido, o se han utilizado unos antiguo
  s en su lugar.
```

6. Ingresamos el siguiente comando para instalar apache, nos preguntará si deseamos continuar y digitamos “yes” o “sí” dependiendo del idioma escogido.

```
sudo apt-get install apache2
```

```
igs@igs:~$ sudo apt-get install apache2
```

7. Se inicia el servicio de Apache, permitiendo que el servidor web comience a servir contenido, para ello usamos el siguiente comando:

```
sudo systemctl start apache2
```

```
igs@igs:~$ sudo systemctl start apache2
```

8. Para ver las configuraciones de IP en el servidor de Linux necesitaremos la herramienta net-tools la cual se instala con el siguiente comando.

```
sudo apt install net-tools
```

```
igs@igs:~$ sudo apt install net-tools_
```

9. Ingresamos el comando “ifconfig” para obtener la dirección IP de la máquina virtual y nos arroja la siguiente dirección IP:

```
igs@igs:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.105 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::20c:29ff:febc:debc prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:cb:de:bc txqueuelen 1000 (Ethernet)
    RX packets 32662 bytes 44016424 (44.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 9754 bytes 1281284 (1.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 260 bytes 26738 (26.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 260 bytes 26738 (26.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

20. Si las configuraciones fueron realizadas correctamente, al momento de ingresar a un navegador con la dirección IP dada, se mostrará la siguiente pantalla. En caso tal de no ver esta página volver a realizar la instalación.



← → ↻ 🔒 192.168.0.105 🌐 ☆ 📄 ☰



Apache2 Default Page

Ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

Importación Gestor De Base De Datos

1. Se inicia apache2 con el siguiente comando.

```
sudo systemctl start apache2
```

```
igs@igs:~$ sudo systemctl start apache2_
```

2. A continuación, se actualiza apache2 con el siguiente comando.

```
sudo apt update
```

```
igs@igs:~$ sudo apt update
Obj:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Obj:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Obj:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Obj:5 https://deb.nodesource.com/node_20.x nodistro InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
igs@igs:~$
```

3. Ahora almacenamos el cache de mysql-server con el siguiente comando.

```
sudo apt-cache search mysql-server
```

```
igs@igs:~$ sudo apt-cache search mysql-serever
```

4. Se instala la versión 8 de mysqlq-server con el siguiente comando, en algún punto preguntará si se desea continuar ingresamos "yes" o "si" dependiendo del idioma elegido.

```
sudo apt install mysql-server-8.0
```

```
igs@igs:~$ sudo apt install mysql-server-8.0
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
mysql-server-8.0 ya está en su versión más reciente (8.0.36-0ubuntu0.22
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualiz
igs@igs:~$
```

5. Una vez instalado se accede a mysql con el siguiente comando.

```
sudo mysql
```

```
igs@igs:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, 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 sta
mysql> _
```

6. Una vez dentro, creamos un usuario y contraseña.

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password
BY 'igs1532';
```

```
mysql> alter user 'root'@'localhost' identified with mysql_native_password by 'igs1532';
Query OK, 0 rows affected (0,02 sec)
```

7. Una vez creado el usuario y contraseña, digitamos el comando exit.

exit

```
mysql> alter user 'root'@'localhost' identified with mysql_native_pas
Query OK, 0 rows affected (0,02 sec)

mysql> exit_
```

8. Ingresamos el siguiente comando, para que mysql pueda ejecutarse.

sudo systemctl start mysql.service

```
igs@igs:~$ sudo systemctl is-enabled mysql.service
enabled
```

9. Se fortifica mysql con el siguiente comando, para que este siempre pida la contraseña para ingresar, seguidamente nos solicitará ingresar la contraseña previamente establecida (paso 6) y presionamos enter

sudo mysql_secure_installation

```
igs@igs:~$ sudo mysql_secure_installation
```

10. Seguidamente a la instalación del paso anterior, se nos hará una serie de preguntas las cuales se deben responder negativamente ingresando la tecla "n".

```
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : n

... skipping.

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? (Press y|Y for Yes, any other key for No) : n

... skipping.
By default, MySQL 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.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : n

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

Instalación De Phpmyadmin

1. Se descargan las ultimas actualizaciones con el siguiente comando:

```
sudo apt update
```

```
igs@igs:~$ sudo apt update
```

2. Se ingresa el siguiente comando para instalar PHP y sus extensiones necesarias para ejecutar phpMyAdmin

```
sudo apt-get install -y php php-tcpdf php-cgi php-pear php-mbstring libapache2-mod-php php-common php-phpseclib php-mysql
```

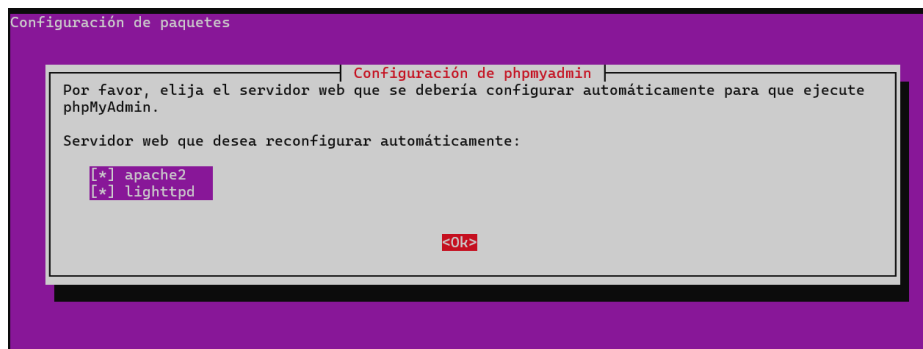
```
sudo apt-get install -y php php-tcpdf php-cgi php-pear php-mbstring libapache2-mod-php php-common php-phpseclib php-mysql
```

3. Se ingresa el comando para instalar phpmyadmin, en algún punto de la instalación nos preguntará si deseamos continuar, respondemos afirmativamente:

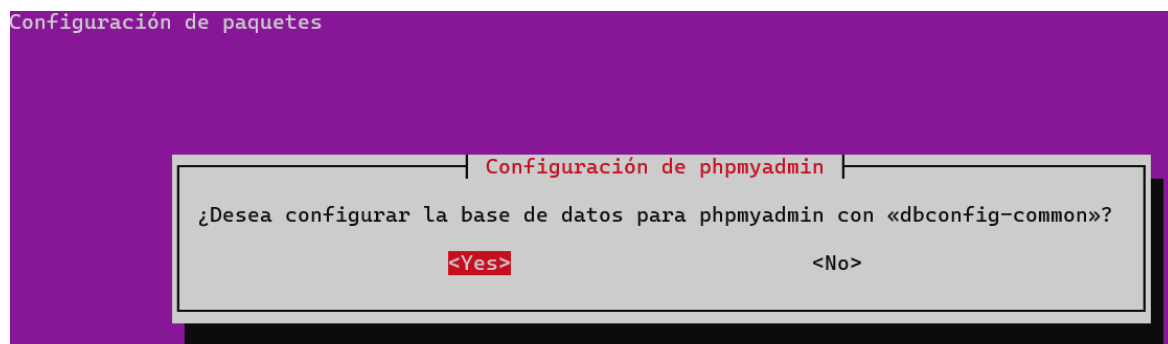
```
sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
```

```
igs@igs:~$ sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
```

4. Nos preguntará que servidor web desea configurar, marcamos las dos opciones bajando con la tecla tab y presionando la tecla espacio, tienen que estar las dos opciones con un asterisco, luego vamos a "ok" y presionamos enter.



5. Nos preguntará si deseamos configurar una base de datos para phpmyadmin, respondemos "yes" presionando la tecla tab y luego presionamos la tecla enter.



6. Nos pedirá ingresar la contraseña que establecimos previamente (paso 6 anterior sección).

Configuración de paquetes

Configuración de phpmyadmin

Proporcione una contraseña para que phpmyadmin se registre con el servidor de base de datos. Si deja este campo en blanco se generará un contraseña aleatoria.

Contraseña de aplicación MySQL para phpmyadmin:

<Ok> <Cancel>

7. Nos pedirá confirmar la contraseña, se ingresa nuevamente, escogemos "ok" y presionamos la tecla enter.

Configuración de paquetes

Configuración de phpmyadmin

Confirmación de contraseña:

<Ok> <Cancel>

8. Se habilita el módulo mbstring de PHP, requerido por phpMyAdmin, ingresamos el siguiente comando:

```
sudo phpenmod mbstring
```

```
igs@igs:~$ sudo phpenmod mbstring
```

9. Reiniciamos el apache2 con el siguiente comando.

```
sudo systemctl restart apache2
```

```
igs@igs:~$ sudo systemctl restart apache2
```

10. Ingresamos a phpMyAdmin en un navegador con la dirección IP que previamente se estableció con la ruta de phpmyadmin:

IPasignada/phpmyadmin/

← → ↻ 🔒 192.168.0.105/phpmyadmin/ ☆ 📌 >> ≡



Bienvenido a phpMyAdmin

Idioma - *Language*

Español - Spanish ▼

Iniciar sesión ⓘ

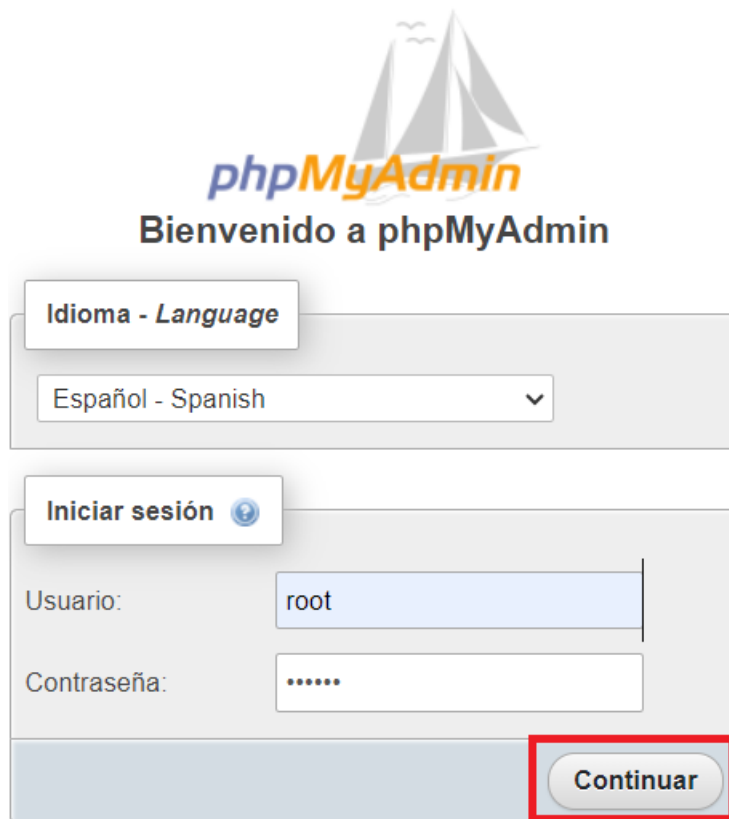
Usuario:

Contraseña:

Continuar

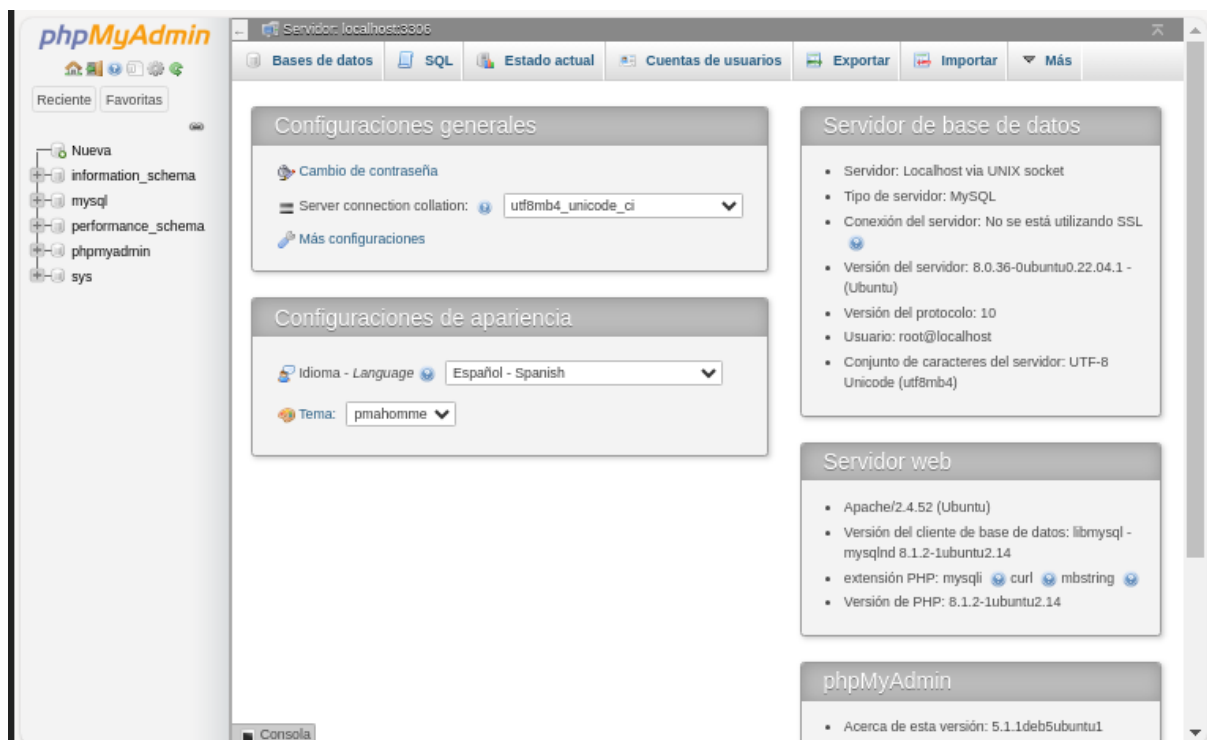
Importación De Base De Datos

1. Ingresamos nuestro usuario, contraseña y damos clic en el botón “Continuar”.

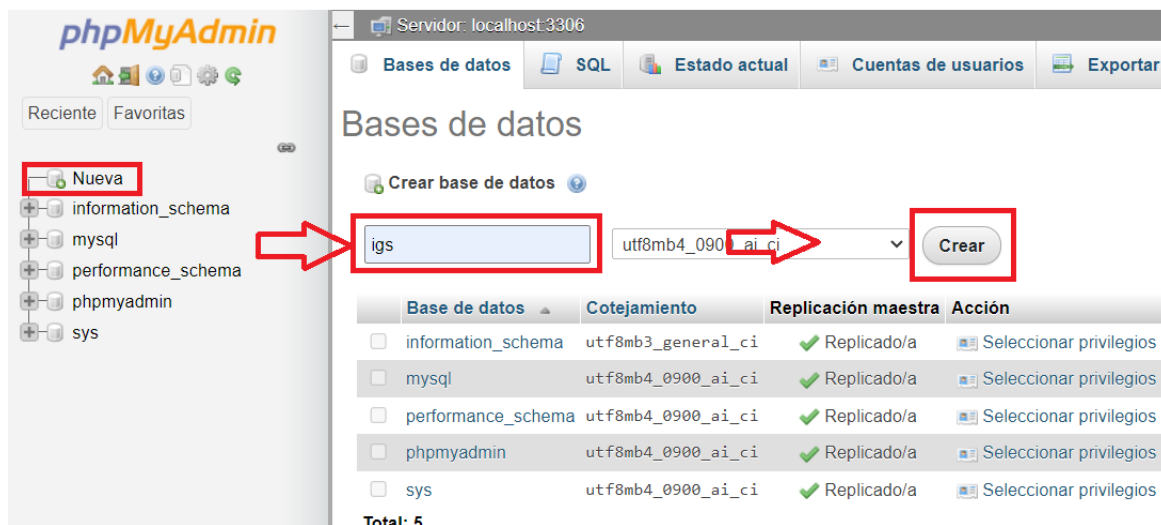


The image shows the phpMyAdmin login interface. At the top, there is a logo with a sailboat and the text "phpMyAdmin" and "Bienvenido a phpMyAdmin". Below this, there is a section for "Idioma - Language" with a dropdown menu set to "Español - Spanish". Underneath is a section for "Iniciar sesión" (Login) with fields for "Usuario:" (Username) containing "root" and "Contraseña:" (Password) with masked characters. A red rectangle highlights the "Continuar" (Continue) button at the bottom right of the login section.

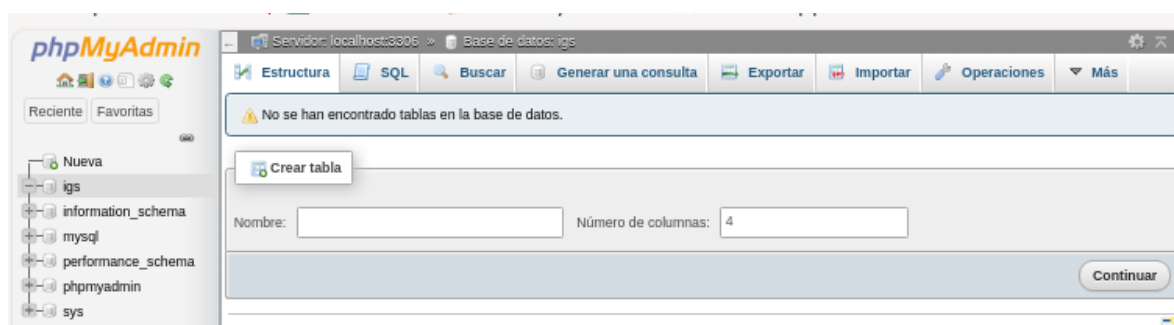
2. Una vez se haya ingresado, esta es la vista que se desplegará.



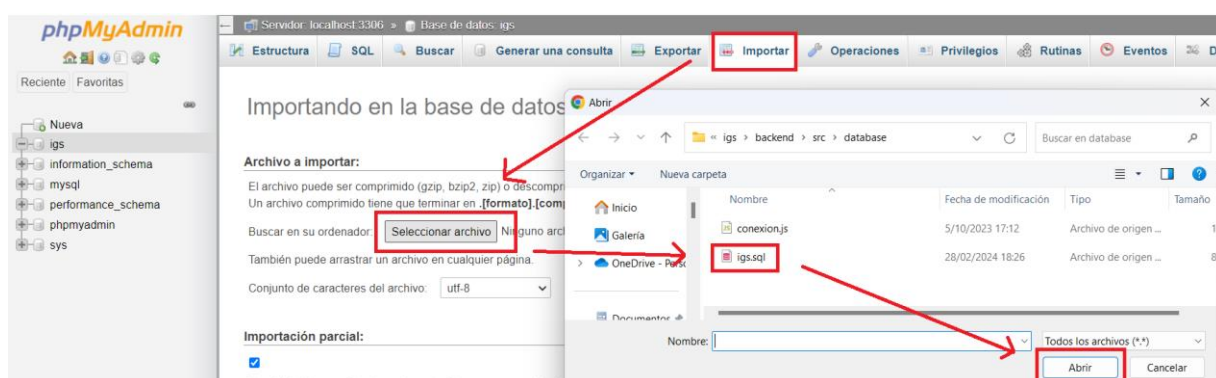
3. Creamos la nueva base de datos, vamos a “nueva”, luego en la sección de crear base de datos ingresamos el nombre de esta, y se da clic en crear.



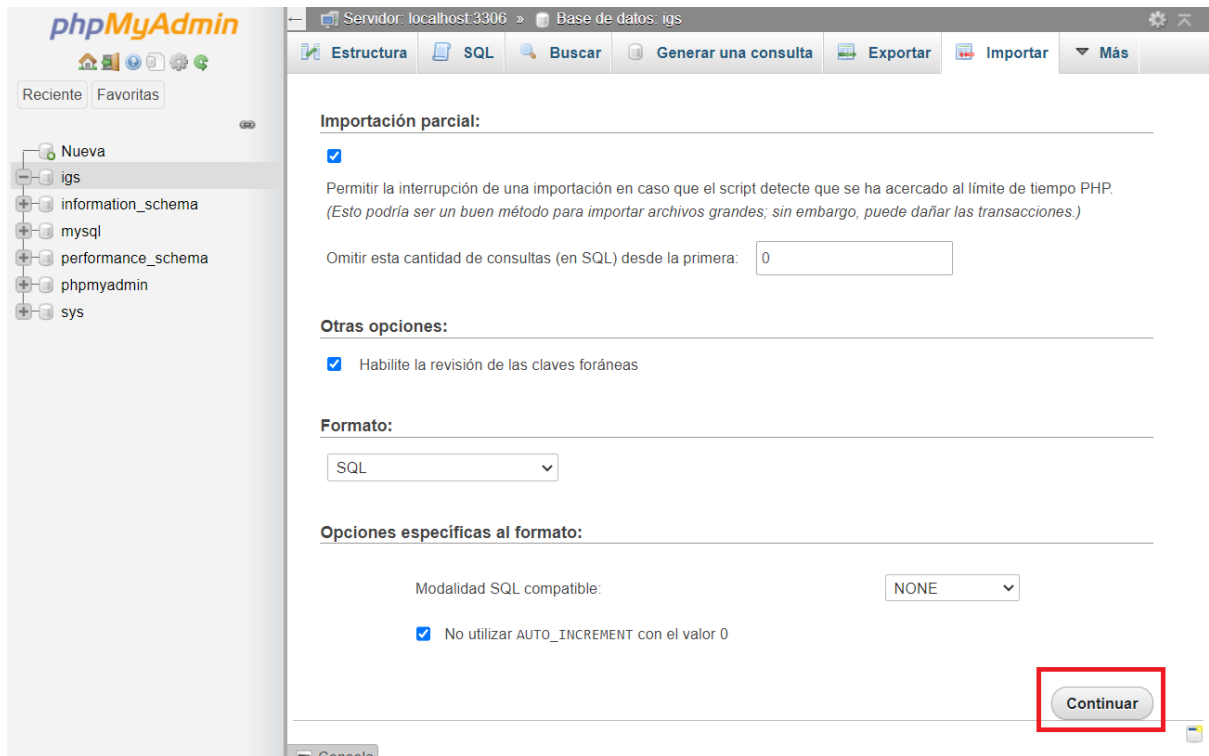
4. Si se creó correctamente observaremos lo siguiente.



5. Para importar la base de datos vamos a la pestaña “importar”, luego en la sección de archivo en importar, damos clic en el botón “elegir archivo” buscamos nuestro archivo sql y finalmente presionamos el botón “Abrir”.

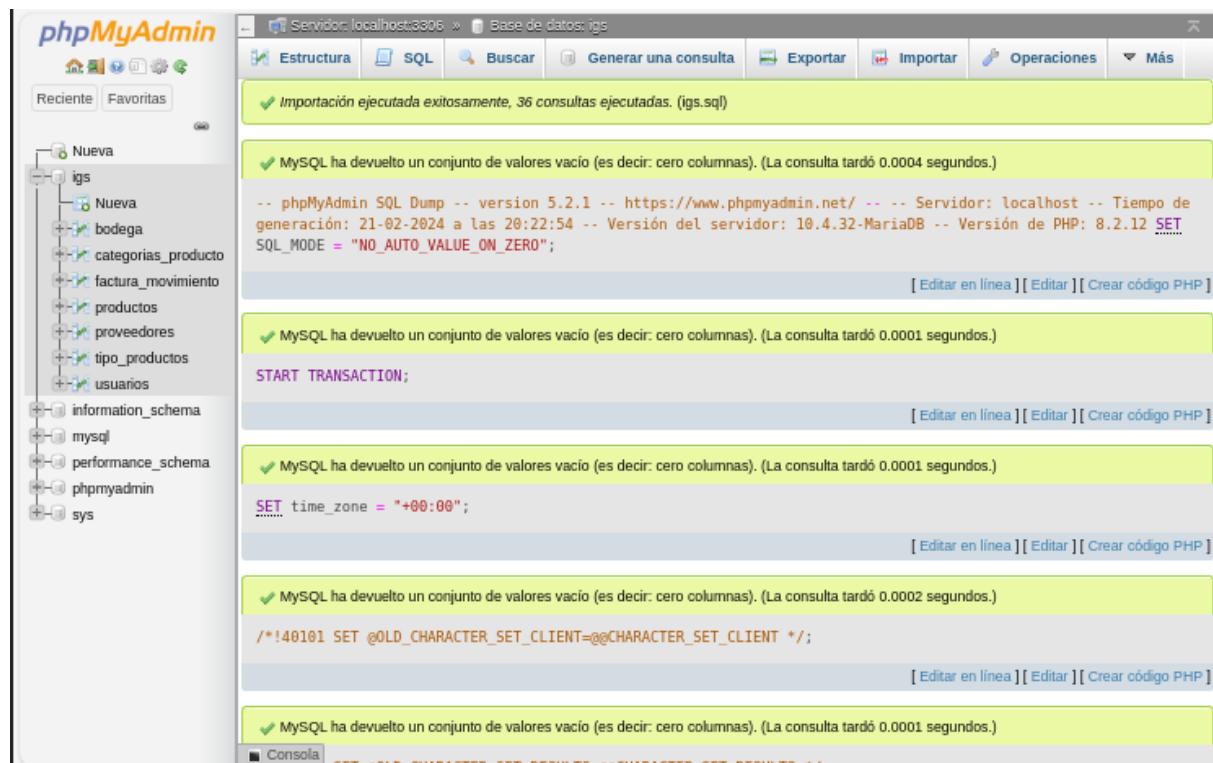


6. Una vez ya cargado el archivo bajamos y damos clic en continuar.



The screenshot shows the phpMyAdmin interface for the 'igs' database. The 'Import' tab is selected. The 'Importación parcial' section has the checkbox checked. The 'Otras opciones' section has the checkbox 'Habilite la revisión de las claves foráneas' checked. The 'Formato' dropdown is set to 'SQL'. The 'Opciones específicas al formato' section has the 'Modalidad SQL compatible' dropdown set to 'NONE' and the checkbox 'No utilizar AUTO_INCREMENT con el valor 0' checked. A red box highlights the 'Continuar' button at the bottom right.

7. Si la importación fue exitosa aparecerá el siguiente mensaje.

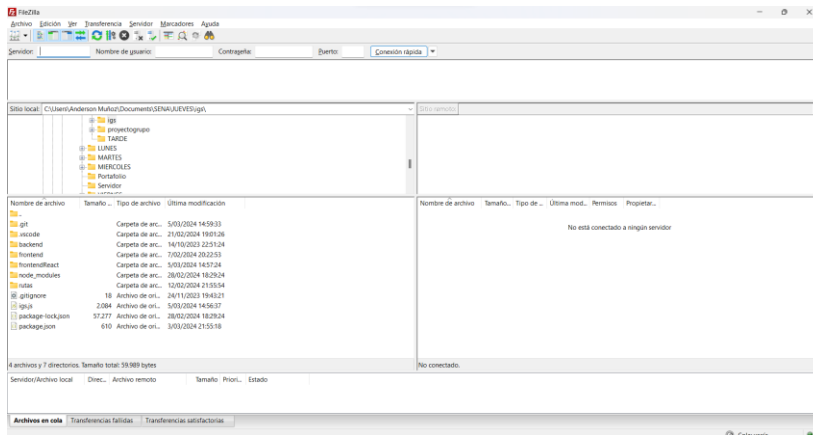


The screenshot shows the phpMyAdmin interface with the 'Console' tab selected. The console displays the following messages:

- Importación ejecutada exitosamente, 36 consultas ejecutadas. (igs.sql)
- MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0004 segundos.)
- phpMyAdmin SQL Dump -- version 5.2.1 -- https://www.phpmyadmin.net/ -- -- Servidor: localhost -- Tiempo de generación: 21-02-2024 a las 20:22:54 -- Versión del servidor: 10.4.32-MariaDB -- Versión de PHP: 8.2.12 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
- MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0001 segundos.)
- START TRANSACTION;
- MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0001 segundos.)
- SET time_zone = "+00:00";
- MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0002 segundos.)
- /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
- MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0001 segundos.)
- SET OLD_CHARACTER_SET_RESULT=@@CHARACTER_SET_RESULT; */;

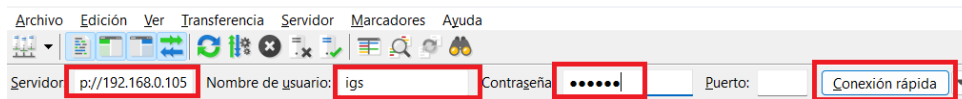
Conexión A FileZilla

1. Abrimos FileZilla

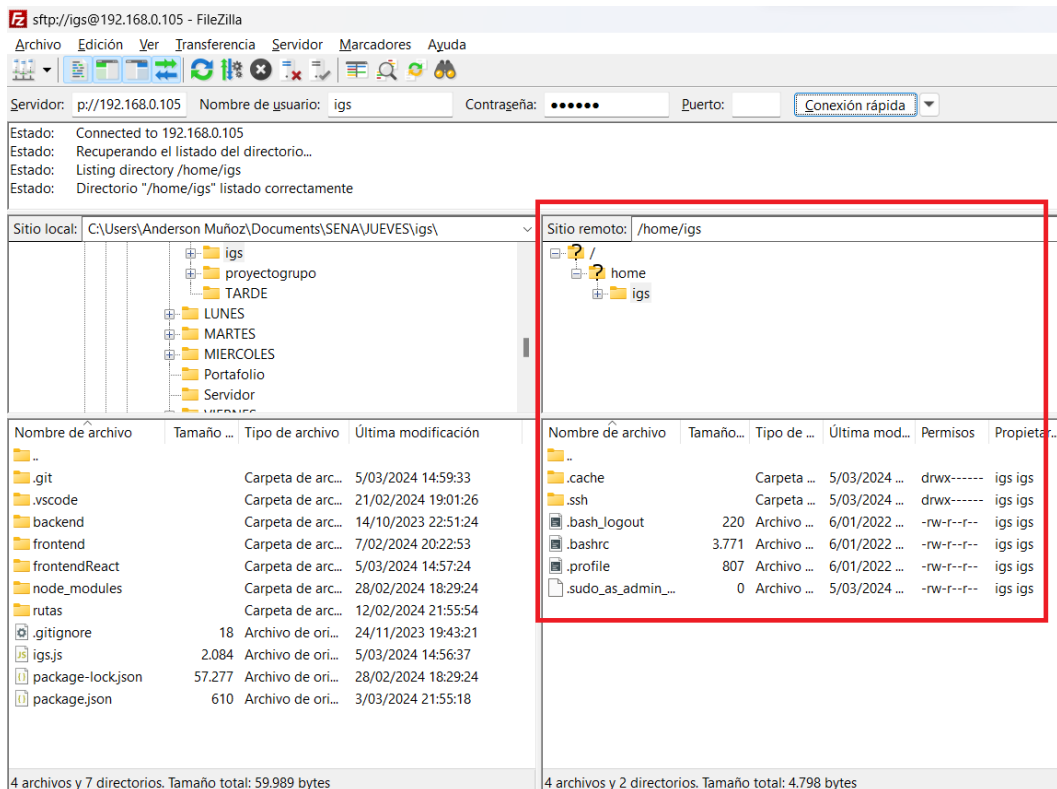


- El servidor se ingresa la URL del servidor local que se configuro, en nombre de usuario y contraseña ingresamos los datos que se establecieron previamente, y en puerto ingresamos 22 (se queda por defecto), y posteriormente damos clic en conexión rápida, luego nos aparecerá un recuadro y damos clic en aceptar.

La ruta del servidor se ingresa así: `sftp://192.168.0.105`



- Si la conexión fue exitosa, en la parte derecha se podrá visualizar los datos del servidor indicando que la conexión fue exitosa.



- Ahora en el directorio remoto (recuadro derecho), creamos una carpeta llamada igs y accedemos a ella

Sitio local: C:\Users\Anderson Muñoz\Documents\SENA\JUEVES\igs\

igs

proyectogrupo

TARDE

LUNES

MARTES

MIERCOLES

Portafolio

Servidor

JUEVES

Nombre de archivo	Tamaño ...	Tipo de archivo	Última modificación
..			
.git		Carpeta de arc...	5/03/2024 14:59:33
.vscode		Carpeta de arc...	21/02/2024 19:01:26
backend		Carpeta de arc...	14/10/2023 22:51:24
frontend		Carpeta de arc...	7/02/2024 20:22:53
frontendReact		Carpeta de arc...	5/03/2024 14:57:24
node_modules		Carpeta de arc...	28/02/2024 18:29:24
rutas		Carpeta de arc...	12/02/2024 21:55:54
.gitignore	18	Archivo de ori...	24/11/2023 19:43:21
igs.js	2.084	Archivo de ori...	5/03/2024 14:56:37
package-lock.json	57.277	Archivo de ori...	28/02/2024 18:29:24
package.json	610	Archivo de ori...	3/03/2024 21:55:18

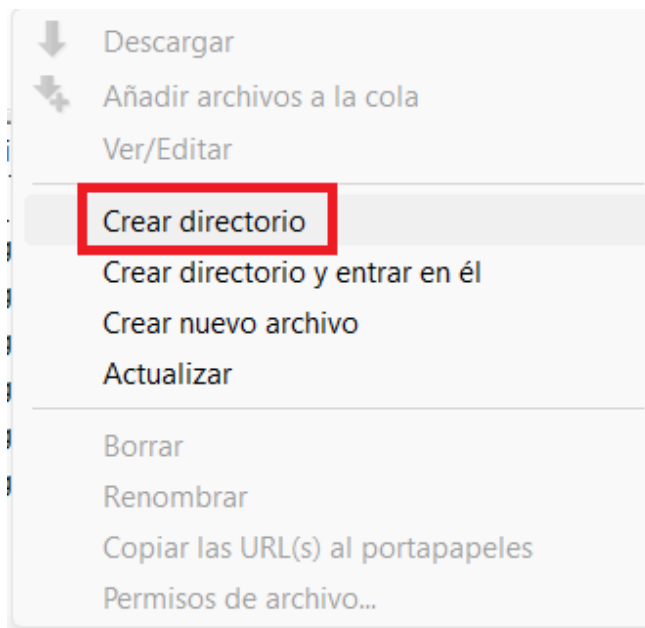
Sitio remoto: /home/igs

home

igs

Nombre de archivo	Tamaño...	Tipo de ...	Última mod...	Permisos	Propietar.
..					
.cache		Carpeta ...	5/03/2024 ...	drwx-----	igs igs
.ssh		Carpeta ...	5/03/2024 ...	drwx-----	igs igs
igs		Carpeta ...			
.bash_logout	220	Archivo ...	6/01/2022 ...	-rw-r--r--	igs igs
.bashrc	3.771	Archivo ...	6/01/2022 ...	-rw-r--r--	igs igs
.profile	807	Archivo ...	6/01/2022 ...	-rw-r--r--	igs igs
.sudo_as_admin...	0	Archivo ...	5/03/2024 ...	-rw-r--r--	igs igs

- Dentro de nuestra carpeta igs y creamos un nuevo directorio llamado “frontendReact” dando clic derecho y seleccionamos “Crear Directorio”



- Ingresamos el nombre “frontendReact” y presionamos “Aceptar”

Crear directorio

Por favor, introduzca el nombre del directorio que debe crearse:

/home/igs/igs/frontendReact

Aceptar

Cancelar

7. Deberíamos tener así el interior de nuestra carpeta igs.

Nombre de archivo	Tamaño...	Tipo de ...	Última mod...	Permisos	Propietar...
..					
frontendReact		Carpeta ...	7/03/2024 ...	drwxrwx...	igs igs

8. Ahora en el directorio local (recuadro izquierdo) buscamos nuestro proyecto igs, seleccionamos las carpetas backend, los archivos igs.js, package-lock.json y package.json, y los arrastramos hacia el directorio remoto (recuadro de la derecha) y esperamos a que carguen (la velocidad de transferencia dependerá de la velocidad la conexión a internet)

sftp://igs@192.168.0.105 - FileZilla

Archivo Edición Ver Transferencia Servidor Marcadores Ayuda

Servidor: p://192.168.0.105 Nombre de usuario: igs Contraseña: Puerto: Cor

Estado: Transferencia correcta, transferidos 2.751 bytes en 1 segundo
 Estado: Comenzando la subida de C:\Users\Anderson Muñoz\Documents\SENA\JUEVES\igs\frontendReact\node_modules\array.pr
 Estado: Transferencia correcta, transferidos 159 bytes en 1 segundo
 Estado: Comenzando la subida de C:\Users\Anderson Muñoz\Documents\SENA\JUEVES\igs\frontendReact\node_modules\array.pr

Nombre de archivo	Tamaño ...	Tipo de archivo	Última modificación
..			
.git		Carpeta de arc...	7/03/2024 14:30:25
.vscode		Carpeta de arc...	21/02/2024 19:01:26
backend		Carpeta de arc...	14/10/2023 22:51:24
frontend		Carpeta de arc...	7/02/2024 20:22:53
frontendReact		Carpeta de arc...	7/03/2024 14:32:14
node_modules		Carpeta de arc...	7/03/2024 14:31:40
rutas		Carpeta de arc...	12/02/2024 21:55:54
.gitignore	18	Archivo de ori...	24/11/2023 19:43:21
igs.js	2.084	Archivo de ori...	6/03/2024 18:01:37
package-lock.json	57.277	Archivo de ori...	7/03/2024 14:31:40
package.json	610	Archivo de ori...	3/03/2024 21:55:18

Seleccionado 3 archivos y 2 directorios. Tamaño total: 59.971 bytes

Nombre de archivo	Tamaño...	Tipo de ...
..		
backend		Carpeta ...
frontendReact		Carpeta ...
igs.js	2.084	Archivo ...
package-lock.json	57.277	Archivo ...
package.json	610	Archivo ...

3 archivos y 2 directorios. Tamaño total: 59.971

Servidor/Archivo local	Direc...	Archivo remoto	Tamaño	Priori...	Estado
sftp://igs@192.168.0...					
C:\Users\Anderson ...	-->	/home/igs/igs/frontend...	2.926	Norm...	Transfiriendo
00:00:00 transcurrido		Quedan --:--:--	100.0%		2.926 bytes (? B/s)

Archivos en cola (45044) Transferencias fallidas Transferencias satisfactorias (167)

9. Ahora accedemos a nuestra carpeta “frontendReact” en el sitio local (recuadro de la izquierda) y también en el sitio remoto (recuadro de la derecha), ahora del recuadro de la izquierda arrastramos hacia el recuadro de la derecha con clic sostenido a todos los archivos excepto la carpeta “node_modules”:

Servidor:
Nombre de usuario:
Contraseña:
Puerto:

Estado: Directorio "/home/igs/" listado correctamente
Estado: Recuperando el listado del directorio "/home/igs/igs/frontendReact" ...
Estado: Listing directory /home/igs/igs/frontendReact
Estado: Directorio "/home/igs/igs/frontendReact" listado correctamente

Sitio local:

env

public

routes

validation

views

js

frontend

frontendReact

Nombre de archivo	Tamaño ...	Tipo de archivo	Última modificación
..			
img		Carpeta de arc...	5/03/2024 14:56:37
node_modules		Carpeta de arc...	20/02/2024 18:34:09
public		Carpeta de arc...	6/12/2023 20:01:51
src		Carpeta de arc...	3/03/2024 15:24:33
.eslintrc.cjs	546	Archivo de ori...	24/11/2023 19:43:21
.gitignore	277	Archivo de ori...	24/11/2023 19:43:21
index.html	354	Chrome HTML ...	6/12/2023 20:01:51
package-lock.json	189.569	Archivo de ori...	7/03/2024 14:29:39
package.json	1.586	Archivo de ori...	7/03/2024 14:29:39
README.md	459	Archivo de ori...	24/11/2023 19:43:21
vite.config.js	174	Archivo de ori...	24/11/2023 19:43:21
vite.config.js.timest...	1.266	Archivo de ori...	10/12/2023 22:04:59

Sitio remoto:

controllers

database

env

public

routes

validation

views

frontendReact

Nombre de archivo	Tamaño...	Tipo de ...	Última mod...	Permisos	Propietar...
..					
dist		Carpeta ...	7/03/2024 ...	drwxrwx...	igs igs
img		Carpeta ...	6/03/2024 ...	drwxrwx...	igs igs
node_modules		Carpeta ...	6/03/2024 ...	drwxrwx...	igs igs
public		Carpeta ...	6/03/2024 ...	drwxrwx...	igs igs
src		Carpeta ...	7/03/2024 ...	drwxrwx...	igs igs
.eslintrc.cjs	546	Archivo ...	7/03/2024 ...	-rw-rw-r--	igs igs
.gitignore	277	Archivo ...	7/03/2024 ...	-rw-rw-r--	igs igs
index.html	354	Chrome ...	7/03/2024 ...	-rw-rw-r--	igs igs
package-lock.json	189.569	Archivo ...	7/03/2024 ...	-rw-rw-r--	igs igs
package.json	1.586	Archivo ...	7/03/2024 ...	-rw-rw-r--	igs igs
README.md	459	Archivo ...	7/03/2024 ...	-rw-rw-r--	igs igs
vite.config.js	174	Archivo ...	7/03/2024 ...	-rw-rw-r--	igs igs
vite.config.js.tim...	1.266	Archivo ...	7/03/2024 ...	-rw-rw-r--	igs igs

Seleccionado 8 archivos y 3 directorios. Tamaño total: 194.231 bytes

8 archivos y 5 directorios. Tamaño total: 194.231 bytes

Ejecutar Backend

1. Una vez los archivos del proyecto fueron cargados a Filezilla, volvemos a nuestro servidor, e instalamos las dependencias para npm con el siguiente comando:

```
sudo apt install npm
```

```
igs@igs:~$ sudo apt install npm
```

2. Ahora accedemos la carpeta creada desde FileZilla donde arrastramos nuestro proyecto con el siguiente comando

```
cd igs
```

```
igs@igs:~$ cd igs
igs@igs:~/igs$ |
```

3. Instalamos el paquete pm2 con el siguiente comando:

```
sudo npm install -g pm2
```

```
igs@igs:~$ sudo npm install -g pm2
```

4. Ahora accedemos a nuestro archivo .env de nuestro backend para configurar la contraseña de acceso (ingresamos la que usamos para acceder a phpmyadmin)

```
cd igs/backend/src/env/
```

```
igs@igs:~$ cd igs/backend/src/env/
igs@igs:~/igs/backend/src/env$ |
```

5. Ingresamos el siguiente comando para editar dicho archivo

```
sudo nano .env
```

```
igs@igs:~/igs/backend/src/env$ sudo nano .env
```

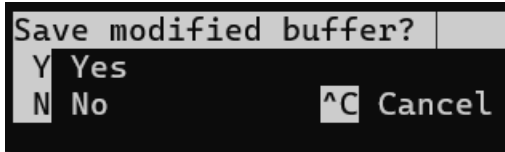
6. Una vez ingresado no redirigirá al archivo .env donde lo podremos editar, en esta caso con las teclas de flechas nos podremos mover a lo largo y ancho, y agregamos la contraseña que se asignó al usuario de phpmyadmin en la sección de Password

```
GNU nano 6.2
DB_HOST = localhost
DB_USER = root
DB_PASSWORD = 123456
DB_PORT = 3306
DB_DATABASE = igs

AUT_SECRET = tokenUsuario
AUT_EXPIRE = 8h
```




7. Ahora para guardar, presionamos las teclas Control + X, y nos preguntará si deseamos guardar los cambios, si se considera que la contraseña es la correcta presionamos la tecla “y” y luego la tecla enter, en caso contrario la tecla “n”, volvemos a ingresar la contraseña y realizamos los mismos pasos para guardar.



8. Al presionar enter en el paso anterior volveremos a esta vista:

```
igs@igs:~/igs/backend/src/env$ sudo nano .env
```

9. Ahora se necesita volver a la carpeta igs, para ello ingresamos estos dos comandos

cd para volver a raíz

```
igs@igs:~/igs/backend/src/env$ cd
```

cd igs para ir a la carpeta igs

```
igs@igs:~$ cd igs
igs@igs:~/igs$
```

10. Ingresamos el comando:

npm install

```
igs@igs:~/igs$ npm install
```

¡¡IMPORTANTE!!: Los pasos 11, 12 y 13 solo son necesarios si no tiene una versión reciente de nodejs, para verificar ello escribimos en consola **node -v**, si nuestra versión es mayor a 16, omitimos estos pasos y pasamos directamente al paso 14

11. Necesitamos actualizar nuestro Node a la versión 16.14, para ello instalamos el gestor de versiones de node llamado nvm con el siguiente comando, una vez haya cargado, cerramos nuestra terminal y volvemos acceder al servidor (ya que estamos en remoto usamos ssh [igs@192.168.0.105](https://192.168.0.105) para acceder nuevamente en una nueva terminal)

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
```

```
igs@igs:~/igs$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
```

12. En nuestra nueva terminal ya ingresados en el servidor, volvemos a la ruta de igs con el comando **cd igs** y añadimos el siguiente comando para instalar la versión necesaria de node:

```
nvm install 20.11.1
```



```
igs@igs:~/igs$ nvm install 16.14
Downloading and installing node v16.14.2...
Downloading https://nodejs.org/dist/v16.14.2/node-v16.14.2-linux-x64.tar.xz...
##### 100,0%
Computing checksum with sha256sum
Checksums matched!
Now using node v16.14.2 (npm v8.5.0)
Creating default alias: default -> 16.14 (-> v16.14.2)
```

13. Ahora ingresamos el comando para usar esa versión de node:

```
nvm use 20.11.1
```

```
igs@igs:~/igs$ nvm use 16.14
Now using node v16.14.2 (npm v8.5.0)
```

14. Ahora ingresamos el comando para ejecutar nuestro backend, y debe arrojar el mensaje de conexión exitosa, en caso contrario volver al paso 4 de esta sección y realizar los mismos pasos:

```
node igs.js
```

```
igs@igs:~/igs$ node igs.js
Servidor IGS ejecutando en http://localhost:3000
Conexión a base de datos exitosa.
```

15. Dado que la conexión fue exitosa, procedemos a presionar las teclas “Control” + “c” para detener la conexión y proceder con la verificación del backend:

```
igs@igs:~/igs$ node igs.js
Servidor IGS ejecutando en http://localhost:3000
Conexión a base de datos exitosa.
^C
igs@igs:~/igs$
```

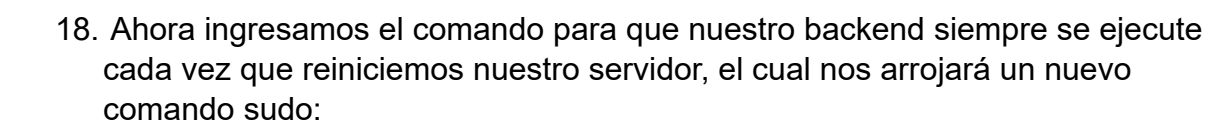
16. Iniciamos nuestro backend con el siguiente comando para mantener la conexión de la base de datos (nodo-1 se puede cambiar por cualquier nombre deseado sin caracteres especiales) y nos arrojará el siguiente cuadro, indicando que el backend está funcionando:

```
pm2 start igs.js --name 'nodo-1'
```

id	name	mode	u	status	cpu	memory
0	nodo-1	fork	0	online	0%	30.5mb

17. Para verificar que si está funcionando vamos a alguna de las rutas de nuestro backend, en este ejemplo se usará la ruta de usuarios y la función que nos lista, la dirección ip tiene que ser la que se generó inicialmente para el servidor, pegamos esta dirección en cualquier navegador, y si nos lista o nos pide un token, realizamos el proceso de manera correcta:

```
http://192.168.0.105:3000/usuario/listar
```



```
igs@igs:~/igs$ pm2 startup
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/usr/bin:/usr/lib/node_modules/pm2/bin/pm2 startup systemd -u igs --hp /home/igs
```

[illegible]



20. Ahora ingresamos el siguiente comando para guardar los cambios y que nuestro backend se ejecute cada vez que reiniciamos el servidor:

```
pm2 save
```

```
igs@igs:~/igs$ pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /home/igs/.pm2/dump.pm2
```

21. Procedemos a reiniciar nuestro servidor, ingresamos el siguiente comando:

```
sudo reboot
```

```
igs@igs:~/igs$ sudo reboot
Connection to 192.168.0.106 closed by remote host.
Connection to 192.168.0.106 closed.
```

22. Una vez reiniciado, ingresamos con nuestro usuario y contraseña y posteriormente ingresamos el siguiente comando para verificar que ya es encuentra ejecutando nuestro backend cada vez se inicia el servidor:

```
pm2 list
```

```
igs@igs:~$ pm2 list
```

id	name	mode	↻	status	cpu	memory
0	backend	fork	0	online	0%	75.1mb

IP Estática

1. Obtenemos nuevamente nuestra dirección IP con el siguiente comando:

`ip address`

```
igs@igs:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:cb:de:bc brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.0.105/24 metric 100 brd 192.168.0.255 scope global dynamic ens33
        valid_lft 3961sec preferred_lft 3961sec
    inet6 fe80::20c:29ff:febc:debc/64 scope link
        valid_lft forever preferred_lft forever
```

2. Obtenemos la ip del router con el siguiente comando:

`ip route show`

```
igs@igs:/etc/netplan$ ip route show
default via 192.168.0.1 dev ens33 proto static
192.168.0.0/24 dev ens33 proto kernel scope link src 192.168.0.106
```

3. Nos dirigimos a la siguiente ruta para editar un archivo en específico.

`cd /etc/netplan`

```
igs@igs:~$ cd /etc/netplan
igs@igs:/etc/netplan$
```

4. Listamos con el siguiente comando para verificar que se encuentre el archivo "00-installer-config.yaml":

`ls`

```
igs@igs:/etc/netplan$ ls
00-installer-config.yaml
```

5. Si el archivo se encuentra, ingresamos el siguiente comando para modificarlo y nos pedirá nuestra contraseña de usuario del servidor, la ingresamos:

`sudo nano 00-installer-config.yaml`

```
igs@igs:/etc/netplan$ sudo nano 00-installer-config.yaml
```

6. Una vez dentro del editor, dejamos de la siguiente manera nuestro archivo modificando los dos últimos números de nuestras IP, guardamos presionando las teclas "Control" + "x":

`network:`

`renderer: networkd`

`ethernets:`

`ens33:`

`addresses:`

```
- 192.168.0.106/24
nameservers:
  addresses: [8.8.8.8, 8.8.4.4]
routes:
  - to: default
    via: 192.168.0.1
version: 2
```

```
GNU nano 6.2                                00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  renderer: networkd
  ethernets:
    ens33:                                     Modificación
      addresses:
        - 192.168.0.106/24
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
      routes:
        - to: default
          via: 192.168.0.1                    IP del router del paso 2
version: 2
```

7. Confirmamos presionando “y” y posteriormente enter:

```
Save modified buffer?
Y Yes
N No      ^C Cancel
```

8. Luego para aplicar los cambios ingresamos el siguiente comando:

```
sudo netplan apply
```

```
igs@igs:/etc/netplan$ sudo netplan apply
```

NOTA: Algunas veces cuando se ejecuta este comando arroja un error donde se menciona la palabra “openvswitch”, si ocurre esto, ingresar el siguiente comando en consola y se presiona enter, es posible que pida ingresar la contraseña de usuario nuevamente, una vez finalice la instalación se vuelve a ingresar el comando sudo netplan apply y enter:

```
sudo apt install openvswitch-switch
```

9. Una vez aplicado el comando anterior, nuestro servidor cambiará de IP, en este caso la IP era 192.168.0.105, pero se modificó a 192.168.0.106, dado que hemos estado trabajando desde un equipo remoto la conexión con nuestro servidor se verá interrumpida dado que se cambió de IP, como se observa a continuación.

```
igs@igs:/etc/netplan$ sudo netplan apply

** (generate:34684): WARNING **: 19:41:49.480: Permissions for /etc/netplan/00-installer-config.yaml
are too open. Netplan configuration should NOT be accessible by others.
WARNING:root:Cannot call Open vSwitch: ovsdb-server.service is not running.

** (process:34682): WARNING **: 19:41:50.400: Permissions for /etc/netplan/00-installer-config.yaml
are too open. Netplan configuration should NOT be accessible by others.

** (process:34682): WARNING **: 19:41:51.195: Permissions for /etc/netplan/00-installer-config.yaml
are too open. Netplan configuration should NOT be accessible by others.

** (process:34682): WARNING **: 19:41:51.195: Permissions for /etc/netplan/00-installer-config.yaml
are too open. Netplan configuration should NOT be accessible by others.
Failed to reload network settings: No such file or directory
WARNING:root:Falling back to a hard restart of systemd-networkd service
igs@igs:/etc/netplan$ client_loop: send disconnect: Connection reset
```

10. Luego debemos verificar si se cambió la dirección ip mediante el comando “ip address” tengamos en cuenta que nuestra dirección ip anterior era 192.168.0.105 y ahora ya se visualiza la modificación que hicimos:

```
igs@igs:/etc/netplan$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:cb:de:bc brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.0.106/24 brd 192.168.0.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:febc:debc/64 scope link
        valid_lft forever preferred_lft forever
```

Ejecutar Frontend

1. Nos dirigimos a la ruta de frontendReact con el siguiente comando

```
cd igs/frontendReact/
```

```
igs@igs:~$ cd igs/frontendReact/
igs@igs:~/igs/frontendReact$
```

2. Ingresamos el comando para instalar las dependencias de npm

```
npm install
```

```
igs@igs:~/igs/frontendReact$ npm install
```

3. Ingresamos el siguiente comando para iniciar el frontend

```
pm2 start npm --name "frontend-igs" -- run dev
```

```
igs@igs:~/igs/frontendReact$ pm2 start npm --name "Frontend-igs" -- run dev
[PM2] Applying action restartProcess on app [npm](igs: [ 1 ])
[PM2] [Frontend-igs](1)
[PM2] Process successfully started
```

id	name	mode	id	status	cpu	memory
0	Backend-igs	fork	0	online	0%	66.0mb
1	Frontend-igs	fork	30	online	0%	21.9mb

4. Ahora ingresamos el comando para que nuestro backend siempre se ejecute cada vez que reiniciemos nuestro servidor, el cual nos arrojará un nuevo comando sudo el cual copiaremos:

```
pm2 startup
```

```
igs@igs:~/igs/frontendReact$ pm2 startup
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/usr/bin /usr/lib/node_modules/pm2/bin/pm2 startup systemd -u igs --hp /home/igs
```

5. Copiamos este nuevo comando que empieza por "sudo", lo pegamos en la misma consola y finalmente presionamos enter, luego ingresamos la contraseña nuevamente.

```
igs@igs:~/igs/frontendReact$ sudo env PATH=$PATH:/usr/bin /usr/lib/node_modules/pm2/bin/pm2 startup systemd -u igs --hp /home/igs
[sudo] password for igs:
[PM2] Init System found: systemd
Platform systemd
Template
[Unit]
Description=PM2 process manager
Documentation=https://pm2.keymetrics.io/
After=network.target
```

6. Ahora ingresamos el siguiente comando para guardar los cambios y que nuestro backend se ejecute cada vez que reiniciamos el servidor:

```
pm2 save
```

```
igs@igs:~/igs/frontendReact$ pm2 save
[PM2] Saving current process list...
[PM2] Successfully saved in /home/igs/.pm2/dump.pm2
```


7. Procedemos a reiniciar nuestro servidor, ingresamos el siguiente comando:

```
sudo reboot
```

```
igs@igs:~/igs/frontendReact$ sudo reboot
Connection to 192.168.174.128 closed by remote host.
Connection to 192.168.174.128 closed.
```

8. Una vez reiniciado, ingresamos con nuestro usuario y contraseña y posteriormente ingresamos el siguiente comando para verificar que ya es encuentra ejecutando nuestro frontend cada vez se inicia el servidor:

```
pm2 list
```

```
igs@igs:~$ pm2 list
```

id	name	mode	pid	status	cpu	memory
0	Backend-igs	fork	0	online	0%	78.9mb
1	Frontend-igs	fork	0	online	0%	66.9mb

9. Ahora para que nuestro frontend detecte el backend debemos de modificar el archivo portConexion.js, para vamos a la siguiente ruta

```
cd igs/frontendReact/src/const
```

```
igs@igs:~$ cd igs/frontendReact/src/const
igs@igs:~/igs/frontendReact/src/const$ _
```

10. Dentro de esa carpeta, editamos el archivo portConexion.js con el siguiente comando, presionamos enter, ingresamos la contraseña y enter nuevamente

```
sudo nano portConexion.jsx
```

```
igs@igs:~/igs/frontendReact/src/const$ sudo nano portConexion.jsx
[sudo] password for igs:
```

11. Dentro de este archivo editamos el valor que tenemos entre comillas simples por nuestra IP

```
GNU nano 6.2 portConexion.jsx
const portConexion = '192.168.174.128';
export default portConexion;
```

12. Una vez editada la dirección IP, presionamos las teclas “Control” + “X”, confirmamos presionando la tecla “y” y luego la tecla enter.

```
Save modified buffer?
Y Yes
N No      ^C Cancel
```

13. Ahora vamos a nuestra dirección IP con el siguiente puerto y se debería de ver IGS

192.168.xxx.xxx:5173

