

MANUAL DEL SISTEMA



Elaborado por:

Basir Jeffrey, Xavier Alexandro
Morán Marcano, Diana Carolina

2023

MANUAL DEL SISTEMA

DEL PORTAL WEB DE EGRESADOS

Para facilitar el la comprensión y el mantenimiento del **Portal web**, se ha confeccionado un Manual del Sistema que indicará las funcionalidades del programa, ahorrando tiempo y colaborando efectivamente con su trabajo.

ELABORADO POR:

BASIR JEFFREY, XAVIER ALEXANDRO.
MORÁN MARCANO, DIANA CAROLINA.

INDICE

INDICE	2
INDICE DE TABLAS.....	¡Error! Marcador no definido.
INTRODUCCIÓN.....	4
Requerimientos Tecnicos	5
REQUERIMIENTOS MÍNIMOS DE HARDWARE	¡Error! Marcador no definido.
REQUERIMIENTOS MÍNIMOS DE SOFTWARE	5
Configuración del entorno productivo	6
Pasos principales	¡Error! Marcador no definido.
Instalar Git	12
Instalar MySQL	23
Cambiar tu contraseña root de MySQL	23
Creacion de bases de datos	23
Instalar Node y npm	¡Error! Marcador no definido.
Instalar Angular	¡Error! Marcador no definido.
Instalar Ngnex	¡Error! Marcador no definido.
Instalar Python.....	¡Error! Marcador no definido.
Instalar de php	¡Error! Marcador no definido.
Instalación de Apache.....	¡Error! Marcador no definido.
. Administrar el proceso de Apache	¡Error! Marcador no definido.
Instalación de composer	¡Error! Marcador no definido.
Instalacion y configuración del portal web.....	6
Instalacion del microservicio 1.....	17
Instalacion del microservicio 2 (Desarrollado en python).....	23
Instalacion del Api Gateway	20
Instalacion del entorno visual	24
Diagramas del Modelado.....	27
Casos de uso.....	27
Diagrama relacional.....	35
Diccionario de datos.....	38
Definición de arquitectura.....	49
Diagrama navegacional	50

INTRODUCCIÓN

Este manual está dirigido para el usuario interesado en conocer su herramienta de trabajo, extraer el máximo rendimiento del portal web, ahorrar tiempo.

El manual detalla los aspectos técnicos e informáticos del portal web con la finalidad de explicar la estructura del aplicativo al personal que quiera administrarlo, editarlo o configurarlo.

Antes de comenzar a hacer uso concreto del software, le recomendamos leer atentamente las instrucciones de este manual.

REQUERIMIENTOS TECNICOS

REQUERIMIENTOS MÍNIMOS DE SOFTWARE

- Un usuario con privilegios de Sudo.
- Conexión a internet estable.
- Ventana de terminal/Línea de comandos.
- El sistema operativo que ejecuta Ubuntu 20.04 Linux.
- Asegúrese de que Apache o cualquier otro proceso no se esté ejecutando en el puerto 80 o 443.
- Sistema Operativo: Este manual esta orientado a su configuración en un entorno Linux bajo la distribución de Ubuntu 20.04, sin embargo, si ya cuenta con su entorno de desarrollo completo en otro sistema operativo puede instalar sin ningún problema el portal web revisando la sección de instalación y configuración del portal web

CONFIGURACIÓN DEL ENTORNO PRODUCTIVO

A efectos del manual se utilizará como dominios los siguientes, los cuales se establecerán como nombre de carpetas, como configuración en el host y variables de entorno del proyecto en los pasos siguientes.

Microservicio general: **mpwe-general.com**

Api Gateway: **api-gateway-pwe.com**

Entorno visual: **front-pwe.com**

Micro servicio de estadísticas: **mpwe-general.com**

Otra observación a tener en cuenta es que la configuración de este manual será tomada como los tres proyectos en un mismo servidor, sin embargo lo ideal sería tener un servidor destinado a cada proyecto,

VERIFICACION E INSTALACION DE PAQUETES

Ejecutar los siguientes comandos

sudo apt-get update -y

sudo apt-get upgrade -y

Al ejecutar el comando, se le pedirá ingresar la contraseña del usuario root, seguidamente se procederá a descargar los paquetes necesarios y una vez finalice la instalación se mostrará como la siguiente figura:

```
micropwe@micropwe:~$ sudo apt-get update -y
[sudo] password for micropwe:
Obj:1 http://ve.archive.ubuntu.com/ubuntu focal InRelease
Des:2 http://ve.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Des:3 http://ve.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Des:4 http://ve.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Des:5 http://ve.archive.ubuntu.com/ubuntu focal/main Translation-es [342 kB]
Des:6 http://ve.archive.ubuntu.com/ubuntu focal/restricted Translation-es [2.152 B]
Des:7 http://ve.archive.ubuntu.com/ubuntu focal/universe Translation-es [1.326 kB]
Des:8 http://ve.archive.ubuntu.com/ubuntu focal/multiverse Translation-es [70,0 kB]
Des:9 http://ve.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [23,5 kB]
Des:10 http://ve.archive.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [12,0 kB]
Descargados 2.112 kB en 9s (223 kB/s)
Leyendo lista de paquetes... Hecho
micropwe@micropwe:~$
```

1. Verificar si las siguientes aplicaciones se encuentran instaladas, si no estan instaladas proceder a su respectiva instalación:

Aplicación	Comando para verificar si está instalado
------------	------------------------------------------

Git	git -v
Composer	composer
Npm	npm -v
Angular	ng --version
Python3	python3 --version
Nginex	nginx -v
php	Php -v
Mysql	mysql -v

INSTALACION DEL SERVIDOR WEB CON NGNEX

REQUISITOS PREVIOS

- Asegúrese de que Apache o cualquier otro proceso no se esté ejecutando en el puerto 80 o 443

INSTALAR NGINX EN UBUNTU

Verificar si tiene instalado Nginx.

nginx -v

```
micropwe@micropwe:~$ nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
```

En caso de no tener instalado Nginx, es decir, en caso de que no te salga la versión del Nginx proceder a realizar la instalación. Para ello ejecute el siguiente comando:

sudo apt install nginx

Para verificar si la instalación se realizó de forma correcta vuelve a ejecutar el comando:

nginx -v

Habilita el servidor web, para ello ejecute el siguiente comando:

sudo systemctl enable nginx

```
micropwe@micropwe:~$ sudo systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemd-sysv-install
.
Executing: /lib/systemd/systemd-sysv-install enable nginx
micropwe@micropwe:~$ _
```

Para verificar el estado de su servidor web, ejecute este comando:

sudo systemctl status nginx

```
micropwe@micropwe:~$ sudo systemctl status nginx
• nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-02-09 04:34:15 UTC; 8min ago
     Docs: man:nginx(8)
    Main PID: 1539 (nginx)
      Tasks: 2 (limit: 6853)
     Memory: 5.5M
    CGroup: /system.slice/nginx.service
            └─1539 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
               └─1540 nginx: worker process

feb 09 04:34:15 micropwe systemd[1]: Starting A high performance web server and a reverse proxy ser
feb 09 04:34:15 micropwe systemd[1]: Started A high performance web server and a reverse proxy serv
lines 1-13/13 (END)
```

CONFIGURAR CORTAFUEGOS DE AJUSTE DE NGINX

Habilitar ufw, para ello ejecute el siguiente comando:

sudo ufw enable

```
micropwe@micropwe:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

Consulta los ufw perfiles de aplicación disponibles, para ello ejecuta el siguiente comando

sudo ufw app list

```
micropwe@micropwe:~$ sudo ufw app list
Available applications:
  Nginx Full
  Nginx HTTP
  Nginx HTTPS
  OpenSSH
micropwe@micropwe:~$ _
```

Habilita el perfil más restrictivo que seguirá permitiendo el tráfico que configuró en el puerto 80:

sudo ufw allow 'Nginx Full'

```
micropwe@micropwe:~$ sudo ufw allow 'Nginx Full'
Rules updated
Rules updated (v6)
micropwe@micropwe:~$ _
```

Verifica el cambio realizado con el siguiente comando:

sudo ufw status numbered

```
micropwe@micropwe:~$ sudo ufw status numbered
Status: active

    To Action From
    --
[ 1] Nginx Full ALLOW IN Anywhere
[ 2] OpenSSH ALLOW IN Anywhere
[ 3] Nginx HTTP ALLOW IN Anywhere
[ 4] 22/tcp ALLOW IN Anywhere
[ 5] Nginx Full (v6) ALLOW IN Anywhere (v6)
[ 6] OpenSSH (v6) ALLOW IN Anywhere (v6)
[ 7] Nginx HTTP (v6) ALLOW IN Anywhere (v6)
[ 8] 22/tcp (v6) ALLOW IN Anywhere (v6)
```

Si tiene otras aplicaciones o servicios para permitir, asegúrese de configurar su firewall para permitir el tráfico.

Por ejemplo, usar el comando **sudo ufw allow 'OpenSSH'** para habilitar inicios de sesión seguros y encriptados a través de la red.

PROBAR INSTALACIÓN Y CONFIGURACIÓN CORRECTA DE NGINEX

Para verificar que se instaló correctamente Nginx, abra un navegador web y escriba en la barra de direcciones http://server_ip_address

Si no sabe cuál es su `server_ip_address` puede ejecutar el siguiente comando:

sudo lshw -short -C network

```
micropwe@micropwe:~$ sudo lshw -short -C network
H/W path      Device      Class      Description
=====
/0/100/3      enp0s3      network    82540EM Gigabit Ethernet Controller
micropwe@micropwe:~$ _
```

Seguidamente el valor obtenido en **device**, en caso de la imagen anterior y como referencia al ejemplo: **enp0s3**, ingrésalo en el siguiente comando.

ip addr show enp0s3 | grep inet | awk '{ print \$2; }' | sed 's/\./.*\$//'

```
micropwe@micropwe:~$ ip addr show enp0s3 | grep inet | awk '{ print $2; }' | sed 's/\./.*$//'
10.0.2.15
fe80::a00:27ff:feb7:fa8b
micropwe@micropwe:~$
```

El valor que obtengas, en caso de la imagen anterior 10.0.2.15, ingresalo en tu url quedando de la siguiente forma **http://10.0.2.15**

Después de abrir esta URL, puede ver la página predeterminada de Nginx.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

CONFIGURACIÓN DEL HOST VIRTUAL PARA EL MICROSERVICIO GENERAL Y EL API GATEWAY EN LARAVEL

En caso de alojar los servicios en un mismo servidor, crea un directorio raíz de documentos y un archivo de configuración separados para cada sitio web. Por lo tanto, deberá crear una estructura de directorios para ambos proyectos dentro de la raíz web de Nginx:

sudo mkdir /var/www/html/mpwe-general.com

```
micropwe@micropwe:~$ sudo mkdir /var/www/html/mpwe-general.com
```

sudo mkdir /var/www/html/api-gateway-pwe.com

```
micropwe@micropwe:~$ sudo mkdir /var/www/html/api-gateway-pwe.com
```

Para crear un archivo de configuración de host virtual Nginx para el microservicio general y para el api gateway. Ejecute el siguiente comando según sea el caso:

```
sudo nano /etc/nginx/sites-available/mpwe-general.com.conf
```

```
sudo nano /etc/nginx/sites-available/api-gateway-pwe.com.conf
```

Ingresa las siguientes líneas en el editor de texto:

```
Server{
    Listen port;
    listen [::]:port;
    server_name sitedomain;
    root /var/www/html/sitedomain/public;
    index index.html index.htm index.php index.nginx-debian.html;
    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-Content-Type-Options "nosniff";
    charset utf-8;
    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }
    location = /favicon.ico { access_log off; log_not_found off; }
    location = /robots.txt { access_log off; log_not_found off; }
    error_page 404 /index.php;
    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php7.4-fpm.sock;
    }
    location .ht{
        deny all;
    }
}
```

****Nota:** sustituye **sitedomain** por el dominio o host de su sitio y **port** por el puerto deseado.

Después de actualizar su archivo de host virtual, presione **CTRL+X**, **Y** y la tecla **Intro** para guardar el archivo de host virtual actualizado.

```
server{
    listen 80;
    listen [::]:80;

    server_name mpwe-general.com;

    root /var/www/html/mpwe-general/public;
    index index.html index.htm index.php index.nginx-debian.html;
    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-Content-Type-Options "nosniff";
    charset utf-8;
    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }
    location = /favicon.ico { access_log off; log_not_found off; }
    location = /robots.txt { access_log off; log_not_found off; }
    error_page 404 /index.php;

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php7.4-fpm.sock;
    }

    location ~ /\.ht{
        deny all;
    }
}
```

```
server{
    listen 80;
    listen [::]:80;

    server_name api-gategay-pwe.com;

    root /var/www/html/api-gateway-pwe.com/public;
    index index.html index.htm inde.php index.nginx-debian.html;
    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-Content-Type-Options "nosniff";
    charset utf-8;
    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }
    location = /favicon.ico{ access_log off; log_not_found off;}
    location = /robots.txt {access_log off; log_not_found off;}
    error_page 404 /index.php;

    location ~ /\.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php7.4-fpm.sock;
    }

    location ~ /\.ht{
        deny all;
    }
}
```

CONFIGURACIÓN DEL HOST VIRTUAL PARA EL ENTORNO VISUAL DEL SISTEMA EN ANGULAR

sudo mkdir /var/www/html/front-pwe.com

```
micropwe@micropwe:~$ sudo mkdir /var/www/html/front-pwe.com
```

Edite el archivo con las siguientes líneas de código

```
server {
    listen 80;
    root /var/www/html/dist/fuse;
    index index.html index.htm index.nginx-debian.html;
    server_name _;

    location / {
        try_files $uri $uri/ =404;
    }
    location ~ /\.(!well-known).*{
        deny all;
    }
}
```

Para crear un archivo de configuración de host virtual Nginx para el front. Ejecute el siguiente comando según sea el caso:

sudo nano /etc/nginx/sites-available/front-pwe.com.conf

Ingresa las siguientes líneas en el editor de texto:

CONFIGURACIÓN DEL HOST VIRTUAL PARA EL MICROSERVICIO DE ESTADÍSTICAS EN PYTHON

mkdir /var/www/html/mpwe-stats.com

```
micropwe@micropwe:/$ sudo mkdir /var/www/html/mpwe-stats.com/
```

Edite el archivo con las siguientes líneas de código

```
server {
    listen 8200;
    listen [::]:8200;
    root /var/www/html/mpwe-stats.com;

    index index.html index.htm index.nginx-debian.html;

    server_name mpwe-stats.com;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

INSTALACION Y CONFIGURACION DE MODULOS REQUERIDOS EN TODOS LOS ENTORNOS

INSTALAR GIT

Verificar si tiene instalado Git.

git --version

```
micropwe@micropwe:~$ git --version
git version 2.25.1
```

En caso de no tener instalado Git, es decir, en caso de que no te salga la versión del Git, proceder a realizar la instalación. Para ello ejecute el siguiente comando:

sudo apt-get install git

Verifica que la instalación se haya hecho correctamente volviendo a usar el comando:

git --versión

CLONAR UN REPOSITORIO DE GIT

Acceda a la carpeta donde aloja el sitio, para muestra del manual, los repositorios se alojaron en carpetas definidas con el nombre del dominio destinado a cada proyecto. Clone el repositorio deseado.

Interfaz gráfica:

cd /var/www/html/front-pwe.com

```
micropwe@micropwe:~$ cd /var/www/html/front-pwe.com/
```

sudo rm -rf * .*

```
micropwe@micropwe:/var/www/html/front-pwe.com$ sudo rm -rf * .*
rm: refusing to remove '.' or '..' directory: skipping '.'
rm: refusing to remove '.' or '..' directory: skipping '..'
```

sudo git clone <https://github.com/XBasir/front-pwe> .

```
micropwe@micropwe:/var/www/html/front-pwe.com$ sudo git clone https://github.com/XBasir/front-pwe .
Cloning into '.'...
remote: Enumerating objects: 1204, done.
remote: Counting objects: 100% (1204/1204), done.
remote: Compressing objects: 100% (726/726), done.
remote: Total 1204 (delta 677), reused 976 (delta 449), pack-reused 0
Receiving objects: 100% (1204/1204), 2.90 MiB | 107.00 KiB/s, done.
Resolving deltas: 100% (677/677), done.
```

API Gateway:

```
cd /var/www/html/api-gateway-pwe.com
```

```
micropwe@micropwe:/ $ cd /var/www/html/api-gateway-pwe.com/
```

```
sudo rm -rf *.*
```

```
micropwe@micropwe:/var/www/html/api-gateway-pwe.com$ sudo rm -rf *.*
rm: refusing to remove '.' or '..' directory: skipping '.'
rm: refusing to remove '.' or '..' directory: skipping '..'
```

```
sudo git clone https://github.com/XBasir/apigateway-pwe .
```

```
micropwe@micropwe:/var/www/html/api-gateway-pwe.com$ sudo git clone https://github.com/XBasir/apigat
eway-pwe .
Cloning into '.'...
remote: Enumerating objects: 358, done.
remote: Counting objects: 100% (358/358), done.
remote: Compressing objects: 100% (208/208), done.
remote: Total 358 (delta 198), reused 281 (delta 121), pack-reused 0
Receiving objects: 100% (358/358), 61.85 KiB | 173.00 KiB/s, done.
Resolving deltas: 100% (198/198), done.
```

Microservicio general

```
cd /var/www/html/mpwe-general.com
```

```
micropwe@micropwe:/ $ cd /var/www/html/mpwe-general.com/_
```

```
sudo rm -rf *.*
```

```
micropwe@micropwe:/var/www/html/mpwe-general.com$ sudo rm -rf *.*
rm: refusing to remove '.' or '..' directory: skipping '.'
rm: refusing to remove '.' or '..' directory: skipping '..'
```

```
sudo git clone https://github.com/XBasir/micropwe .
```

```
micropwe@micropwe:/var/www/html/mpwe-general.com$ sudo git clone https://github.com/XBasir/micropwe
.
Cloning into '.'...
remote: Enumerating objects: 446, done.
remote: Counting objects: 100% (446/446), done.
remote: Compressing objects: 100% (249/249), done.
remote: Total 446 (delta 248), reused 363 (delta 165), pack-reused 0
Receiving objects: 100% (446/446), 72.47 KiB | 455.00 KiB/s, done.
Resolving deltas: 100% (248/248), done.
```

Microservicio de estadísticas

```
cd /var/www/html/mpwe-stats.com
```

```
micropwe@micropwe:/ $ cd /var/www/html/mpwe-stats.com/_
```

```
sudo rm -rf *.*
```

```
micropwe@micropwe:/var/www/html/mpwe-stats.com$ sudo rm -rf *.*
rm: refusing to remove '.' or '..' directory: skipping '.'
rm: refusing to remove '.' or '..' directory: skipping '..'
```

```
micropwe@micropwe:~$ sudo git clone https://github.com/XBasir/micropwe-stats
Cloning into '.'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 24 (delta 2), reused 24 (delta 2), pack-reused 0
Unpacking objects: 100% (24/24), 5.75 KiB | 111.00 KiB/s, done.
```

****Nota:** El comando **sudo rm -rf *** es utilizado para eliminar cualquier contenido de la carpeta donde se clonara el proyecto.

****Nota:** Agregue el "." al final del comando git clone para clonar los archivos del proyecto en el mismo directorio.

INSTALAR MYSQL

Verificar si tiene instalado mysql.

sudo mysql -v

```
micropwe@micropwe:~$ sudo mysql -v
[sudo] password for micropwe:
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

En caso de no tener instalado Mysql ejecute el siguiente comando:

sudo apt install mysql-server

```
micropwe@micropwe:~$ sudo apt install mysql-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libbcgi-fast-perl libbcgi-pm-perl libencode-locale-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmecab2 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8
  mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server-8.0
  mysql-server-core-8.0
Paquetes sugeridos:
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinyca
Se instalarán los siguientes paquetes NUEVOS:
  libbcgi-fast-perl libbcgi-pm-perl libencode-locale-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmecab2 libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8
  mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server mysql-server-8.0
  mysql-server-core-8.0
0 actualizados, 25 nuevos se instalarán, 0 para eliminar y 3 no actualizados.
Se necesita descargar 36,7 MB de archivos.
Se utilizarán 319 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] _
```

Confirma el proceso ingresando la letra S y seguidamente presionando enter.

Valida el estado de MySQL.

sudo systemctl status mysql

Una vez ejecutado el comando anterior se mostrará algo como la siguiente figura.

```
micropwe@micropwe:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-02-09 04:23:46 UTC; 27s ago
     Main PID: 16207 (mysqld)
    Status: "Server is operational"
      Tasks: 39 (limit: 6853)
     Memory: 362.6M
    CGroup: /system.slice/mysql.service
            └─16207 /usr/sbin/mysqld

feb 09 04:23:43 micropwe systemd[1]: Starting MySQL Community Server...
feb 09 04:23:46 micropwe systemd[1]: Started MySQL Community Server.
```

Asegurar la instalación de MySQL ya que por defecto su configuración es propensa a vulnerabilidades, para ello ejecuta el siguiente comando:

sudo mysql_secure_installation

Establece si se deseas la contraseña de root para MySQL, si es así debe ingresarla y confirmarla.

Completa las opciones de seguridad para asegurar MySQL, después de esto accede a MySQL para verificar la instalación con el siguiente comando, esta acción pedirá ingresar la contraseña de root que has creado

sudo mysql -u root -p

```
micropwe@micropwe:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.32-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Para salir del modo mysql escribe la palabra **exit** y presiona enter.

```
mysql> exit
Bye
micropwe@micropwe:~$
```

INSTALACION DE LOS SERVIDORES DONDE SE ENCONTRARÁ ALOJADO EL MICROSERVICIO GENERAL Y EL API GATEWAY

INSTALAR PHP EN UBUNTU

Laravel se basa en PHP, por lo que deberá instalar PHP y módulos relacionados.

Para instalar PHP y módulos relacionados, ejecute los siguientes comandos:

sudo apt-get install software-properties-common -y


```
micropwe@micropwe:~$ sudo apt-get install software-properties-common -y
[sudo] password for micropwe:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
software-properties-common ya está en su versión más reciente (0.99.9.10).
fijado software-properties-common como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

sudo add-apt-repository ppa:ondrej/php -y

```
micropwe@micropwe:~$ sudo add-apt-repository ppa:ondrej/php -y
Obj:1 http://ve.archive.ubuntu.com/ubuntu focal InRelease
Des:2 http://ppa.launchpad.net/ondrej/php/ubuntu focal InRelease [23,9 kB]
Des:3 http://ve.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Des:4 http://ppa.launchpad.net/ondrej/php/ubuntu focal/main amd64 Packages [115 kB]
Des:5 http://ve.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Des:6 http://ppa.launchpad.net/ondrej/php/ubuntu focal/main Translation-en [37,5 kB]
Des:7 http://ve.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Des:8 http://ve.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2.370 kB]
37% [8 Packages 1.476 kB/2.370 kB 62%] 18,0 kB/s 6min 1s
```

**sudo apt-get install php7.4 php7.4-mbstring php7.4-gettext php7.4-zip php7.4-fpm
php7.4-curl php7.4-mysql php7.4-gd php7.4-cgi php7.4-xml php7.4-bcmath php7.4-
imagick php7.4-intl -y**

```
micropwe@micropwe:~$ sudo apt-get install php7.4 php7.4-mbstring php7.4-gettext php7.4-zip php7.4-fp
m php7.4-curl php7.4-mysql php7.4-gd php7.4-cgi php7.4-xml php7.4-bcmath php7.4-imagick php7.4-intl
-y_
```

La -y bandera en el comando permitirá automáticamente que el servidor instale PHP 7.4. Si no agrega la -y bandera, simplemente le pedirá confirmación.

INSTALAR COMPOSER EN UBUNTU

Verifique si tiene instalado composer y en su versión 2 o superior con el siguiente comando

Composer --version

En caso de poseer composer pero en su versión 1 ejecute el siguiente comando, caso contrario salte hasta el próximo comando:

sudo apt-get remove composer -y

En caso de no poseer composer ejecute el siguiente comando para instalar composer en el servidor de Ubuntu.

php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"

```
micropwe@micropwe:/$ sudo php -r "copy('https://getcomposer.org/installer','composer-setup.php');"
```

php composer-setup.php --install-dir=/usr/bin --filename=composer

```
micropwe@micropwe:/$ sudo php composer-setup.php --install-dir=/usr/bin --filename=composer
All settings correct for using Composer
Downloading...

Composer (version 2.5.2) successfully installed to: /usr/bin/composer
Use it: php usr/bin/composer
```

INSTALACION DEL MICROSERVICIO GENERAL

Crea una base de datos para la administración de los datos del microservicio general. Al realizar estos pasos recuerde dejar a la mano el nombre de la bdd, la clave del usuario y el nombre del usuario creado a continuación.

Para muestras del manual se creó una base de datos llamada “lu_ms1_products_api”

Para ello primero ingresar a mysql con el siguiente comando:

sudo mysql -u root -p

```
micropwe@micropwe:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.32-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Crear la base de datos con el siguiente comando:

CREATE DATABASE lu_ms1_products_api;

```
mysql> CREATE DATABASE lu_ms1_products_api;
Query OK, 1 row affected (0,34 sec)
```

Crear un usuario para asignarlo a la bdd recién creada con el siguiente comando:

CREATE USER 'pwe'@'localhost' IDENTIFIED BY '123';

```
mysql> CREATE USER 'pwe'@'localhost' IDENTIFIED BY '123'
-> ;
Query OK, 0 rows affected (0,15 sec)
```

Otorgar todos los privilegios de la base de datos para el usuario recién creado, ejecuta el siguiente comando:

GRANT ALL PRIVILEGES ON * . * TO 'pwe'@'localhost';

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'pwe'@'localhost';
Query OK, 0 rows affected (0,07 sec)
```

Para que los cambios tengan efecto inmediatamente, restablece todos los privilegios usando el comando:

FLUSH PRIVILEGES;

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,05 sec)
```

Salir del modo Mysql escribiendo la palabra: **exit**

```
mysql> exit
Bye
```

Ingresa al directorio raíz donde se alojará el microservicio general

cd /var/www/html/mpwe-general.com

```
micropwe@micropwe:/$ cd /var/www/html/mpwe-general.com/_
```

Clona el repositorio <https://github.com/XBasir/micropwe> . en caso de no realizarlo en pasos anteriores.

Ingresa el siguiente comando

sudo -su root

Establezca el permiso y la propiedad adecuados para laravel con el siguiente comando:

chmod 0+w var/www/html/mpwe-general.com

chmod 0+w var/www/html/mpwe-general.com/storage

Seguido de esto, ejecute el comando para establecer permisos y finalmente ejecute

ctrl +d para

Salir del modo root.

Instala las dependencias del proyecto con el siguiente comando

composer install

```
micropwe@micropwe:/var/www/html/mpwe-general.com$ composer install
```

Copie el archivo .env desde .env.example

cp env.example .env

```
micropwe@micropwe:/var/www/html/mpwe-general.com$ cp env.example .env
```

A continuación, edite el archivo .env y defina su base de datos:

sudo nano .env

Cambia o agrega las siguientes líneas al .env con los datos de tu base de datos configurada en pasos anteriores:

**DB_CONNECTION=mysql
DB_HOST=HOST_DE_TU_BASE_DE_DATOS
DB_PORT=3306
DB_DATABASE=NOMBRE_DE_TU_BASE_DE_DATOS
DB_USERNAME=NOMBRE_DE_USUARIO_ASIGNADO_A_TU_BASE_DE_DATOS
DB_PASSWORD=CLAVE_DE_USUARIO_ASIGNADO_A_TU_BASE_DE_DATOS**

```
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1  
DB_PORT=3306  
DB_DATABASE=lu_ms1_products_api  
DB_USERNAME=pwe  
DB_PASSWORD=123
```

Agrega las siguientes líneas del .env con los datos de tu servidor de correos para que funcione el envío de correos correctamente:

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME= CORREO_DEL_SERVIDOR_DE_CORREOS
MAIL_PASSWORD=CLAVE_DEL_CORREO_DEL_SERVIDOR_DE_CORREOS
MAIL_FROM= CORREO_QUE_SALDRA_EN_EL_CORREO_ENVIADO
MAIL_NAME= NOMBRE_QUE_SALDRA_EN_EL_CORREO_ENVIADO
```

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=27e70
MAIL_PASSWORD=13d0
MAIL_ENCRYPTION=tls
MAIL_FROM=prueba@gmail.com
MAIL_NAME=PortalWEB
```

Para efectos de manual, se utilizo la configuración de envío con los datos de mailtrap

Los campos MAIL_FROM y MAIL_NAME son los datos que se mostrara en el correo recibido, específicamente en el campo DE del correo

Cambia los datos de tu entorno

```
APP_DEBUG=false
APP_URL=URL_DEL_MICROSERVICIO_GENERAL
APP_URL_FRONT=URL_DEL_ENTORNO_VISUAL_DONDE_ALOJAS_ANGULA
```

R

```
APP_DEBUG=false
APP_URL=http://mpwe-general.com
APP_URL_FRONT=http://front-pwe.com:4200
```

Después de actualizar su archivo .env , presione CTRL+X , seguido la tecla Y para confirmar los cambios y por último la tecla Intro o Enter para guardar el archivo .env

Ejecutar el comando:

php artisan migrate --seed

```
micropwe@micropwe:/var/www/html/mpwe-general.com$ php artisan migrate --seed
Migration table created successfully.
Migrating: 2020_03_14_083218_create_products_table
Migrated: 2020_03_14_083218_create_products_table (0.23 seconds)
Migrating: 2022_02_26_000454_create_acto_grado_table
Migrated: 2022_02_26_000454_create_acto_grado_table (0.22 seconds)
Migrating: 2022_02_26_014444_create_presentacion_dep_table
Migrated: 2022_02_26_014444_create_presentacion_dep_table (0.18 seconds)
Migrating: 2022_02_26_014454_create_evento_table
Migrated: 2022_02_26_014454_create_evento_table (0.2 seconds)
Migrating: 2022_02_26_014526_create_actividad_extension_table
Migrated: 2022_02_26_014526_create_actividad_extension_table (0.18 seconds)
Migrating: 2022_02_27_190253_create_carrera_table
Migrated: 2022_02_27_190253_create_carrera_table (0.18 seconds)
Migrating: 2022_02_27_191322_create_bolsa_trabajo_table
Migrated: 2022_02_27_191322_create_bolsa_trabajo_table (0.21 seconds)
Migrating: 2022_02_27_194725_create_egresado_table
Migrated: 2022_02_27_194725_create_egresado_table (0.91 seconds)
Migrating: 2022_02_27_200836_create_bolsa_egresado_table
Migrated: 2022_02_27_200836_create_bolsa_egresado_table (0.73 seconds)
Migrating: 2022_02_29_130444_create_banco_table
```

INSTALACION DEL API GATEWAY

Crea una base de datos para la administración de los datos del micro servicio general. Al realizar estos pasos recuerde dejar a la mano el nombre de la bdd, la clave del usuario y el nombre del usuario creado a continuación.

Para muestras del manual se creo una base de datos llamada “lu_ms1_api_gateway”

Para ello primero ingresar a mysql con el siguiente comando:

sudo mysql -u root -p

```
micropwe@micropwe:/var/www/html/api-gateway-pwe.com$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 8.0.32-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Crear la base de datos con el siguiente comando:

CREATE DATABASE lu_ms1_api_gateway;

```
mysql> CREATE DATABASE lu_ms1_api_gateway;
Query OK, 1 row affected (0,08 sec)
```

Crear un usuario para asignarlo a la bdd recién creada con el siguiente comando:

CREATE USER 'pweapi'@'localhost' IDENTIFIED BY '123';

```
mysql> CREATE USER 'pweapi'@'localhost' IDENTIFIED BY '123';
Query OK, 0 rows affected (0,14 sec)
```

Otorgar todos los privilegios de la base de datos para el usuario recién creado, ejecuta el siguiente comando:

GRANT ALL PRIVILEGES ON *.* TO 'pwe'@'localhost';

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'pweapi'@'localhost';
Query OK, 0 rows affected (0,08 sec)
```

Para que los cambios tengan efecto inmediatamente, restablece todos los privilegios usando el comando:

FLUSH PRIVILEGES;

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,05 sec)
```

Salir del modo Mysql escribiendo la palabra: **exit**

```
mysql> exit
Bye
```

Ingresa al directorio raíz donde se alojará el microservicio general

cd /var/www/html/api-gateway-pwe.com

```
micropwe@micropwe:/$ cd /var/www/html/api-gateway-pwe.com/
```

Clona el proyecto con el comando git clone <https://github.com/XBasir/apigateway-pwe>. En caso de no haberlo clonado en pasos anteriores.

Establezca el permiso y la propiedad adecuados para ello ingrese el siguiente comando

sudo -su root

seguidamente establezca permisos con el siguiente comando

chmod 0+w var/www/html/api-gateway-pwe.com

chmod 0+w var/www/html/api-gateway-pwe.com/storage

```
root@micropwe:/var/www/html/api-gateway-pwe.com# chmod o+w /var/www/html/api-gateway-pwe.com/
```

Seguido de esto, ejecute el comando para establecer permisos y finalmente, para salir del modo root ejecute:

ctrl +d para

Instala las dependencias del proyecto con el comando:

composer install

```
micropwe@micropwe:/var/www/html/api-gateway-pwe.com$ composer install
```

Copie el archivo .env desde .env.example

cp .env.example .env

```
micropwe@micropwe:/var/www/html/api-gateway-pwe.com$ cp env.example .env_
```

Genere una clave de cifrado, con el siguiente comando.

php artisan jwt:secret

```
micropwe@micropwe:/var/www/html/api-gateway-pwe.com$ php artisan jwt:secret
jwt-auth secret [dS5I3pbKC38FCB2VJDhzE5XBvYqDuH1VYvNdb467eTFHywWHPKSE0IWC3hw7fzGJC] set successfully.
```

A continuación, edite el .env archivo y defina su base de datos:

sudo nano .env

```
micropwe@micropwe:/var/www/html/api-gateway-pwe.com$ sudo nano .env
[sudo] password for micropwe: _
```

Cambia o agrega las siguientes líneas al .env con los datos de tu base de datos configurada en pasos anteriores:

```
DB_CONNECTION=mysql
DB_HOST=HOST_DE_TU_BASE_DE_DATOS
DB_PORT=3306
DB_DATABASE=NOMBRE_DE_TU_BASE_DE_DATOS
DB_USERNAME=NOMBRE_DE_USUARIO_ASIGNADO_A_TU_BASE_DE_DATOS
DB_PASSWORD=CLAVE_DE_USUARIO_ASIGNADO_A_TU_BASE_DE_DATOS
```

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=lu_ms1_api_gateway
DB_USERNAME=pweapi_
DB_PASSWORD=123
```

Agrega las siguientes líneas del .env con los datos de tu servidor de correos para que funcione el envío de correos correctamente:

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME= CORREO_DEL_SERVIDOR_DE_CORREOS
MAIL_PASSWORD=CLAVE_DEL_CORREO_DEL_SERVIDOR_DE_CORREOS
MAIL_FROM= CORREO_QUE_SALDRA_EN_EL_CORREO_ENVIADO
MAIL_NAME= NOMBRE_QUE_SALDRA_EN_EL_CORREO_ENVIADO
```

```
MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=27e70
MAIL_PASSWORD=13d0
MAIL_ENCRYPTION=tls
MAIL_FROM=prueba@gmail.com
MAIL_NAME=PortalWEB
```

Para efectos de manual, se utilizo la configuración de envío con los datos de mailtrap

Los campos MAIL_FROM y MAIL_NAME son los datos que se mostrara en el correo recibido, específicamente en el campo DE del correo

Cambia los datos de tu entorno

```
APP_DEBUG=false
APP_URL=URL_DEL_API_GATEWAY
PWE_SERVICE_BASE_URI=http://mpwe-general.com
PWE_STATISTICS_SERVICE_BASE_URI=http://mpwe-stats.com
```

```
APP_DEBUG=false
APP_URL=http://api-gateway-pwe.com
APP_TIMEZONE=UTC
PWE_SERVICE_BASE_URI=http://mpwe-general.com
PWE_STATISTICS_SERVICE_BASE_URI=http://mpwe-stats.com_
```

Después de actualizar su archivo .env , presione CTRL+X , seguido la tecla Y para confirmar los cambios y por último la tecla Intro o Enter para guardar el archivo .env

Ejecutar el comando:

```
php artisan migrate --seed
```

```
micropwe@micropwe:/var/www/html/api-gateway-pwe.com$ php artisan migrate --seed
Migration table created successfully.
Migrating: 2022_01_22_164518_add_users_table
Migrated: 2022_01_22_164518_add_users_table (0.12 seconds)
Migrating: 2022_05_17_123579_create_roles_table
Migrated: 2022_05_17_123579_create_roles_table (0.12 seconds)
Migrating: 2022_05_17_145237_create_permissions_table
Migrated: 2022_05_17_145237_create_permissions_table (0.15 seconds)
Migrating: 2022_05_17_146538_create_permission_user_table
Migrated: 2022_05_17_146538_create_permission_user_table (1.06 seconds)
Migrating: 2022_05_17_155759_create_permission_role_table
Migrated: 2022_05_17_155759_create_permission_role_table (0.75 seconds)
Migrating: 2022_05_17_156834_create_role_user_table
Migrated: 2022_05_17_156834_create_role_user_table (0.69 seconds)
Migrating: 2022_07_16_156834_add_cedula_for_user_table
Migrated: 2022_07_16_156834_add_cedula_for_user_table (0.21 seconds)
Seeding: RoleTableSeeder
```

INSTALACION DE LOS SERVICIOS PARA EL MANEJO DE PYTHON EN LOS SERVIDORES DONDE SE ENCONTRARÁ ALOJADO EL MICROSERVICIO DE ESTADISTICAS

INSTALAR PYTHON Y EL PAQUETE DE ENTORNOS VIRTUALES.

sudo apt install python3-pip

```
micropwe@micropwe:/var/www/html/mpwe-stats.com$ sudo apt install python3-pip_
```

sudo apt install python3-venv

```
micropwe@micropwe:/var/www/html/mpwe-stats.com$ sudo apt install python3-venv
```

INSTALAR GUARNICON Y FLASK

Para una interfaz de puerta de enlace del servidor web (Gunicorn), proceda a ejecutar el siguiente comando:

pip install flask

```
micropwe@micropwe:/var/www/html/mpwe-stats.com$ pip install gunicorn flask
```

INSTALAR WHEEL

Para asegurarse de que sus paquetes se instalarán incluso si faltan archivos de ruedas:

pip install wheel

```
micropwe@micropwe:/$ pip install wheel
Requirement already satisfied: wheel in /usr/lib/python3/dist-packages (0.34.2)
micropwe@micropwe:/$ _
```

INSTALACION DEL MICROSERVICIO 2 (DESARROOLADO EN PYTHON)

Debes tener previamente e instalado y configurado el microservicio 1.

Ingresa al directorio raíz donde se alojara el microservicio de estadísticas.

cd /var/www/html/mpwe-stats.com

```
micropwe@micropwe:/$ cd /var/www/html/mpwe-stats.com/_
```

Clona el proyecto con el comando en caso de no clonarlo en pasos anteriores:

git clone <https://github.com/XBasir/micropwe-stats> .

Cree un entorno virtual para almacenar los requisitos de Python de su proyecto Flask escribiendo:

python3 -m venv micropwe-stats

```
micropwe@micropwe:/$ sudo python3 -m venv micropwe-stats
```

Activar el entorno virtual usando este comando:

source venv/bin/activate

```
micropwe@micropwe:/$ source micropwe-stats/bin/activate
(micropwe-stats) micropwe@micropwe:/$
```

pip install --upgrade pip

```
micropwe@micropwe:/var/www/html/mpwe-stats.com$ pip install --upgrade pip
```

pip install -r requirements.txt

```
micropwe@micropwe:/var/www/html/mpwe-stats.com$ pip install -r requeriments.txt_
```

export FLASK_APP=stats-pwe.py

```
micropwe@micropwe:/var/www/html/mpwe-stats.com$ export FLASK_APP=stats-pwe.py
```

flask run

```
micropwe@micropwe:/var/www/html/mpwe-stats.com$ flask run
```

INSTALACION DE LOS SERVICIOS PARA EL MANEJO DE ANGULAR EN LOS SERVIDORES DONDE SE ENCONTRARÁ ALOJADO ENTORNO VISUAL

INSTALAR NODE Y NPM

Habilite el repositorio de NodeSource ejecutando el siguiente curl comando como usuario con privilegios de sudo:

curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -

Una vez que el repositorio de NodeSource esté habilitado, instale Node.js y npm escribiendo:

sudo apt-get install nodejs -y

```
micropwe@micropwe:/var/www/html/front-pwe.com$ sudo apt-get install -y nodejs
```

El paquete nodejs contiene los binarios node y npm

Verifique que Node.js y npm se hayan instalado correctamente imprimiendo sus versiones:

node --version

npm --version

```
micropwe@micropwe:/var/www/html/front-pwe.com$ node --version
v12.22.12
micropwe@micropwe:/var/www/html/front-pwe.com$ npm --version
6.14.16
```

INSTALAR ANGULAR

Para ello debe ejecutar el siguiente comando

npm install -g @angular/cli

```
micropwe@micropwe:/var/www/html/front-pwe.com$ sudo npm install -g @angular/cli@latest  
[sudo] password for micropwe:
```

Para verificar si todo se instalo correctamente ejecute el siguiente comando.

ng version

INSTALACION DEL ENTORNO VISUAL

Para proceder con los pasos siguientes debe tener previamente instalado el Api Gateway y los microservicios 1 y 2.

Ingresa al directorio raíz donde se alojará el entorno visual

cd /var/www/html/front-pwe.com

```
micropwe@micropwe:~$ cd /var/www/html/front-pwe.com/
```

Clona el repositorio solo si aun no lo has clonado

git clone https://github.com/XBasir/front-pwe .

Establezca el permiso y la propiedad adecuados para ello ingrese el siguiente comando

sudo -su root

seguidamente establezca permisos con el siguiente comando

chmod 0+w var/www/html/front -pwe.com

```
root@micropwe:/var/www/html/api-gateway-pwe.com# chmod o+w /var/www/html/api-gateway-pwe.com/  
root@micropwe:/var/www/html/api-gateway-pwe.com# exit
```

Seguido de esto, ejecute el comando para establecer permisos y finalmente, para salir del modo root ejecute:

ctrl +d para

Instala las dependencias del proyecto con el comando de angular ejecutando el comando:

npm install

Obtenga los contenidos distribuibles finales construyendo la aplicación con

ng build --prod

```
micropwe@micropwe:/var/www/html/front-pwe.com$ ng build --prod_
```

Establezca permisos con copiar la carpeta dist en la carpeta www.

sudo -su root

Seguidamente establezca permisos con el siguiente comando

chmod 0+w var/www/html

Seguido de esto, ejecute el comando para establecer permisos y finalmente, para salir del modo root ejecute:

ctrl +d para

Mueva los archivos a su destino final para que los sirva el servidor web

```
mv /var/www/html/front-pwe.com/dist /var/www/html
```

ACTIVACION DEL HOST VIRTUAL CREADO

Habilite el host virtual que se encuentra configurando con el siguiente comando según sea el caso:

```
sudo ln -s /etc/nginx/sites-available/mpwe-general.com.conf /etc/nginx/sites-enabled/
```

```
micropwe@micropwe:/etc/nginx/sites-available$ sudo ln -s /etc/nginx/sites-available/mpwe-general.com.conf /etc/nginx/sites-enabled/
```

```
sudo ln -s /etc/nginx/sites-available/api-gateway-pwe /etc/nginx/sites-enabled/
```

```
micropwe@micropwe:/etc/nginx/sites-available$ sudo ln -s /etc/nginx/sites-available/api-gateway-pwe.com.conf /etc/nginx/sites-enabled/
```

```
sudo ln -s /etc/nginx/sites-available/mpwe-stats.com.conf /etc/nginx/sites-enabled/
```

```
micropwe@micropwe:/etc/nginx/sites-available$ sudo ln -s /etc/nginx/sites-available/mpwe-stats.com.conf /etc/nginx/sites-enabled/
```

```
sudo ln -s /etc/nginx/sites-available/front-pwe.com /etc/nginx/sites-enabled/
```

```
micropwe@micropwe:/etc/nginx/sites-available$ sudo ln -s /etc/nginx/sites-available/front-pwe.com.conf /etc/nginx/sites-enabled/
```

A continuación, compruebe si hay algún error de sintaxis en Nginx con el siguiente comando:

```
nginx-t
```

```
root@micropwe:/etc/nginx/sites-available# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Finalmente, reinicie el servidor Nginx para aplicar los cambios. Ejecute el siguiente comando:

```
sudo systemctl restart nginx
```

```
root@micropwe:/etc/nginx/sites-available# sudo systemctl restart nginx
```

Si su servidor Nginx se reinicia con éxito, podrá acceder a su proyecto en el navegador.

DIAGRAMAS DEL MODELADO

CASOS DE USO

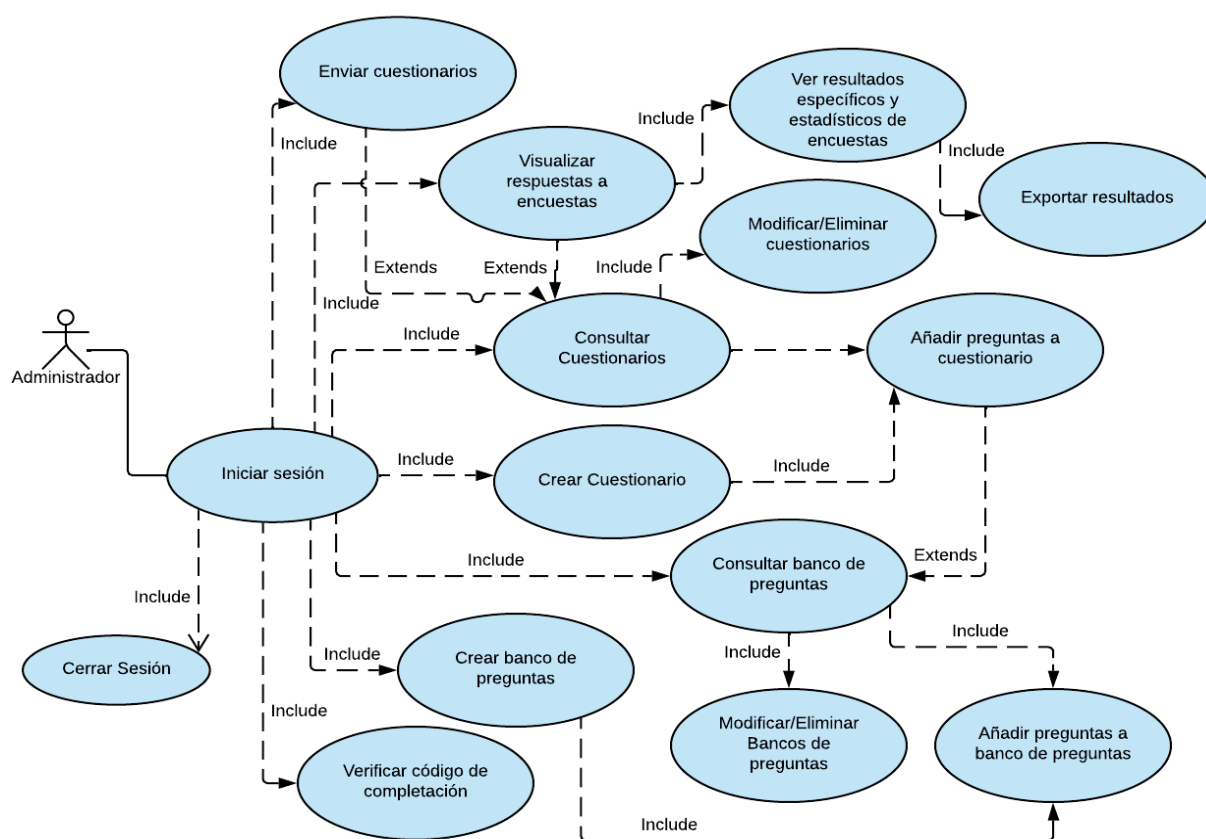
En el diagrama de caso de uso se detalla el papel a desempeñar en relación con el aplicativo por parte de las personas relacionadas.

Cada cuadro a mostrar, describen desde el ID del caso, el cual es un identificador de tipo alfanumérico, compuesto por las letras CU que significan Caso de Uso seguido del número del caso, además, muestran el nombre, la descripción, el actor y el flujo normal de trabajo con las acciones que puede ejecutar dicho actor.

El 1er caso de uso reflejado en la es el caso de uso: “Gestión de cuestionario”, este caso de uso define las posibles acciones que puede realizar el usuario administrador al momento de gestionar un cuestionario.

Estas acciones abarcan tanto el crear, editar, ver y eliminar un cuestionario, así como también, la gestión de preguntas de un cuestionario de forma individual y a través de los bancos de preguntas.

Además, este caso de uso también incluye la acción de validar el código final que se le otorgará a un usuario egresado una vez complete un formulario, esto con el fin de validar si el cuestionario se completó o no de forma exitosa.



Tomando en cuenta lo planteado anteriormente y resumiendo con ello las posibles acciones a efectuar por el usuario administrador, a continuación se describe el flujo normal de trabajo del usuario con respecto a la gestión de cuestionarios, además de una breve descripción del caso de uso y del usuario participe del mismo.

CUADRO 1. CASO DE USO: “GESTIÓN DE CUESTIONARIO”

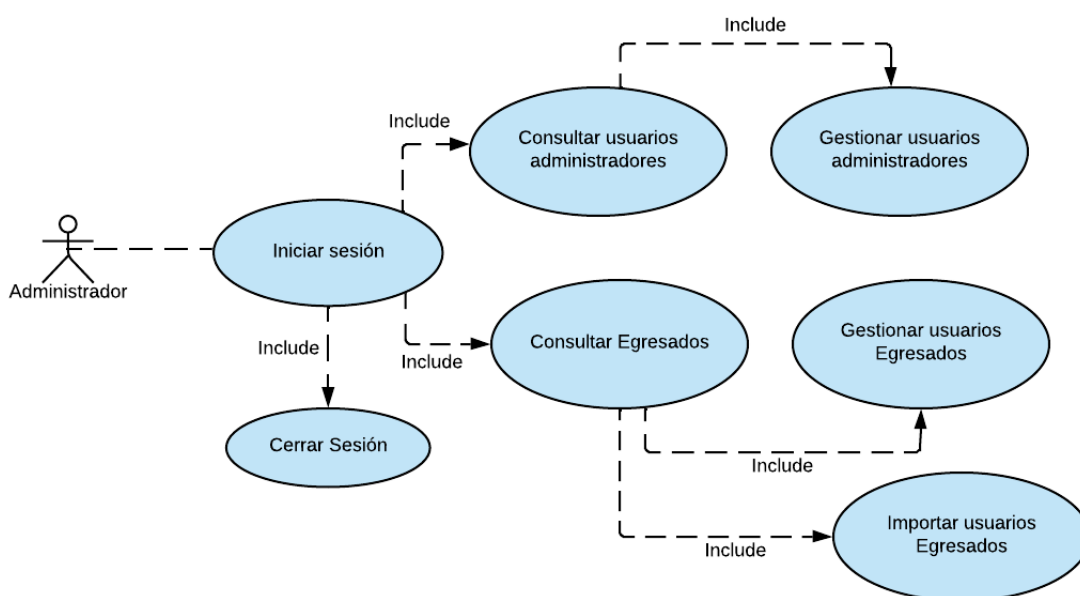
Nombre del caso de uso: Gestión de cuestionarios	ID: CU01
Actores: Administrador	
Descripción: Permite al usuario administrador con permisos crear, consultar, enviar, eliminar y validar cuestionarios, además de gestionar el banco de preguntas, exportar y comparar las respuestas de estos cuestionarios.	

Flujo normal

El usuario administrador puede:

1. Iniciar sesión.
2. Enviar cuestionarios.
3. Visualizar respuestas de los cuestionarios.
4. Ver resultados específicos y estadísticos de los cuestionarios.
5. Exportar resultados de los cuestionarios.
6. Crear cuestionarios.
7. Consultar cuestionarios
8. Modificar/eliminar cuestionarios.
9. Añadir preguntas a los cuestionarios.
10. Consultar banco de preguntas.
11. Crear banco de preguntas
12. Añadir preguntas a banco de preguntas.
13. Modificar/Eliminar banco de preguntas.
14. Verificar código de completación.
15. Cerrar sesión.

Para el caso de uso 2, "Gestión de usuarios", se especifican las funcionalidades que puede realizar el usuario administrador con respecto a la gestión de usuarios en general, esto incluye tanto el ver, editar, crear y eliminar un usuario, ya sea administrador o egresado, así como, el importar información relacionada a estos de forma masiva.

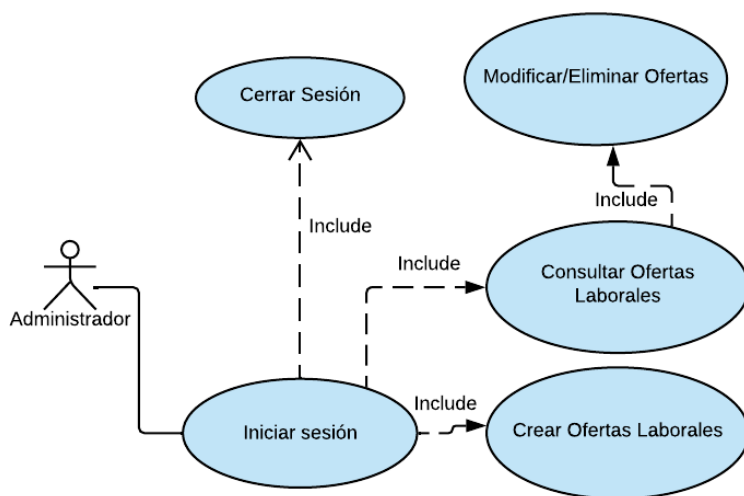


De acuerdo con la información anterior, y en pro a una mejor comprensión del caso de uso, a continuación se detalla de forma simplificada el flujo normal de trabajo de un usuario según el Caso de uso: "Gestión de usuarios" y con ello, la definición del actor responsable de este caso de uso.

Nombre del caso de uso: Gestión de usuarios	ID: CU02
Actores: Administrador	
Descripción: Permite al usuario administrador con permisos, Gestionar usuarios (administradores y egresados).	
<p style="text-align: center;">Flujo normal</p> <p>El usuario administrador puede:</p> <ol style="list-style-type: none"> 1. Iniciar sesión. 2. Consultar usuarios administradores. 3. Gestionar usuarios administradores (visualizar, crear, modificar y eliminar). 4. Consultar usuarios egresados. 5. Gestionar usuarios egresados (visualizar, crear, modificar y eliminar). 6. Importar usuarios egresados 7. Cerrar sesión. 	

Con respecto al tercer caso de uso, se busca previsualizar las posibles funciones del usuario administrador al momento de gestionar las ofertas laborales.

En este caso de uso, el usuario administrador puede eliminar, ver, crear y editar ofertas laborales, estas acciones son realizadas con el fin de otorgarle al usuario egresado el conocimiento necesario para poder optar por alguna propuesta laboral que sea de su interés.



Para este caso, en el se puede observar a detalle el flujo normal de trabajo que trae consigo el caso de uso, además de los actores partícipes de este caso de uso.

Nombre del caso de uso: Gestión de ofertas laborales	ID: CU03
Actores: Administrador	
Descripción: Permite al usuario administrador con permisos, Gestionar las ofertas laborales que posteriormente se le mostraran al usuario egresado	
Flujo normal	

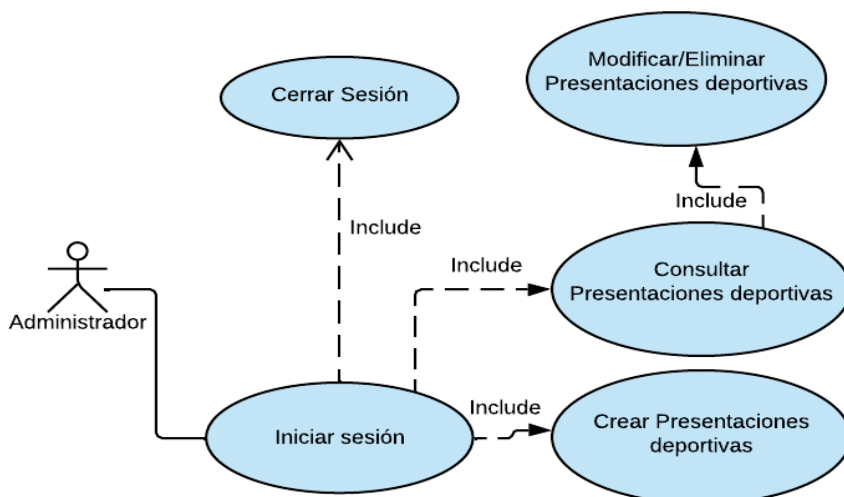
El usuario administrador puede:

1. Iniciar sesión.
2. Crear ofertas laborales.
3. Consultar ofertas laborales.
4. Modificar/Eliminar ofertas.
5. Cerrar sesión.

El cuarto caso de uso mostrado se define la gestión de presentaciones deportivas, realizadas por el usuario administrador.

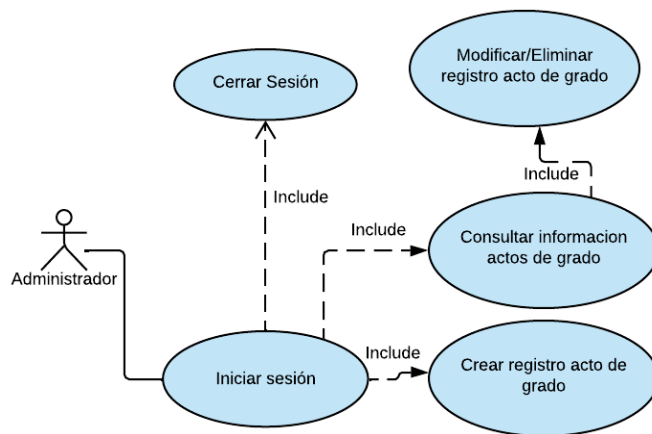
En este caso de uso, se detallan las acciones que puede realizar un usuario administrador con respecto a la gestión de presentaciones deportivas, las cuales, posteriormente serán de carácter informativo para el usuario egresado.

Estas presentaciones deportivas son presentaciones o actividades deportivas relacionadas con la institución educativa y ligadas al deporte. Para su gestión, el usuario administrador puede editar, crear, visualizar o eliminar dichas actividades



Nombre del caso de uso: Gestión de presentaciones deportivas	ID: CU04
Actores: Administrador	
Descripción: Permite al usuario administrador con permisos, Gestionar las presentaciones deportivas que posteriormente se le mostraran al usuario egresado	
Flujo normal	
El usuario administrador puede: <ol style="list-style-type: none"> 1. Iniciar sesión. 2. Crear presentaciones deportivas. 3. Consultar presentaciones deportivas. 4. Modificar/Eliminar presentaciones deportivas. 5. Cerrar sesión. 	

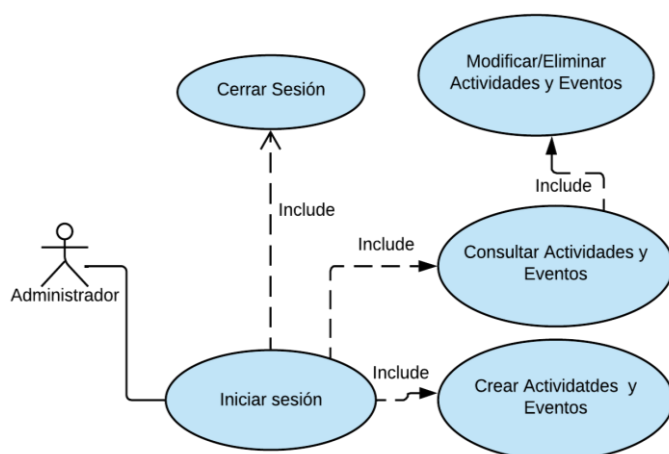
Para el quinto caso de uso, denominado gestión de actos de grado, se define información referente a los actos de grados relacionados a la institución universitaria, con el fin de que el egresado pueda visualizar dicha información al momento de ingresar al sistema.



Este caso de uso está compuesto por diversas acciones que puede ejecutar el usuario administrador al momento de gestionar los actos de grado, estas acciones abarcan el crear, editar, visualizar y/o eliminar un registro relacionado con el acto de grado y puede ser visualizado de forma simplificada en el descrito a continuación.

Nombre del caso de uso: Gestión de actos de grado	ID: CU05
Actores: Administrador	
Descripción: Permite al usuario administrador con permisos, Gestionar los actos de grado que posteriormente se le mostraran al usuario egresado	
<p style="text-align: center;">Flujo normal</p> <p>El usuario administrador puede:</p> <ol style="list-style-type: none"> 1. Iniciar sesión. 2. Crear registro de actos de grado. 3. Consultar información de actos de grado 4. Modificar/Eliminar registro de actos de grado. 5. Cerrar sesión. 	

El sexto caso de uso consiste en la representación de las funcionalidades del usuario administrador en gestión de eventos y actividades de cualquier temática libre organizado en la universidad.

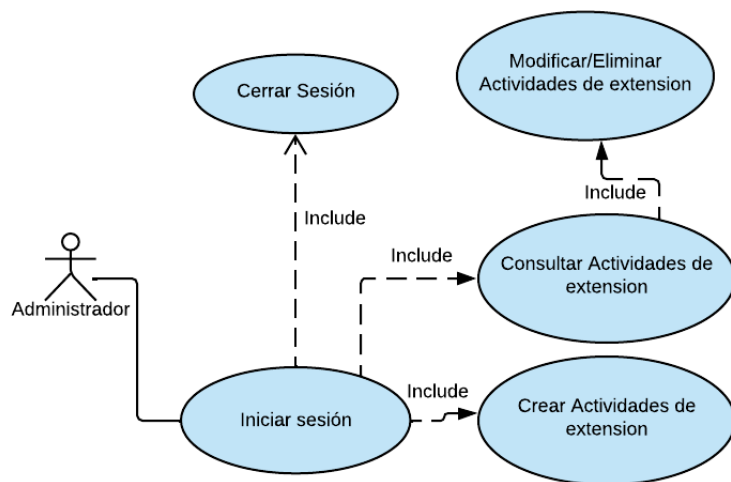


En este caso de uso, simplificado se reflejan las acciones que el usuario administrador puede ejecutar, abarcando con ello el consultar, editar, crear y/o eliminar actividades y eventos, los cuales, posteriormente serán de carácter informativo para el usuario egresado.

Nombre del caso de uso: Gestión de actividades y eventos.	ID: CU06
Actores: Administrador	
Descripción: Permite al usuario administrador con permisos, Gestionar las actividades y eventos que posteriormente se le mostraran al usuario egresado	
<p style="text-align: center;">Flujo normal</p> <p>El usuario administrador puede:</p> <ol style="list-style-type: none"> 1. Iniciar sesión. 2. Crear actividades y eventos. 3. Consultar información de actividades y eventos. 4. Modificar/Eliminar actividades y eventos. 5. Cerrar sesión. 	

El séptimo caso de uso, se detallan las acciones del usuario administrador al momento de gestionar las actividades de extensión. Estas actividades están relacionadas a las actividades de estudio de la carrera como diplomados, cursos de postgrados, entre otros, las cuales, una vez gestionadas por el usuario administrador, podrán ser de carácter informativo para el usuario egresado.

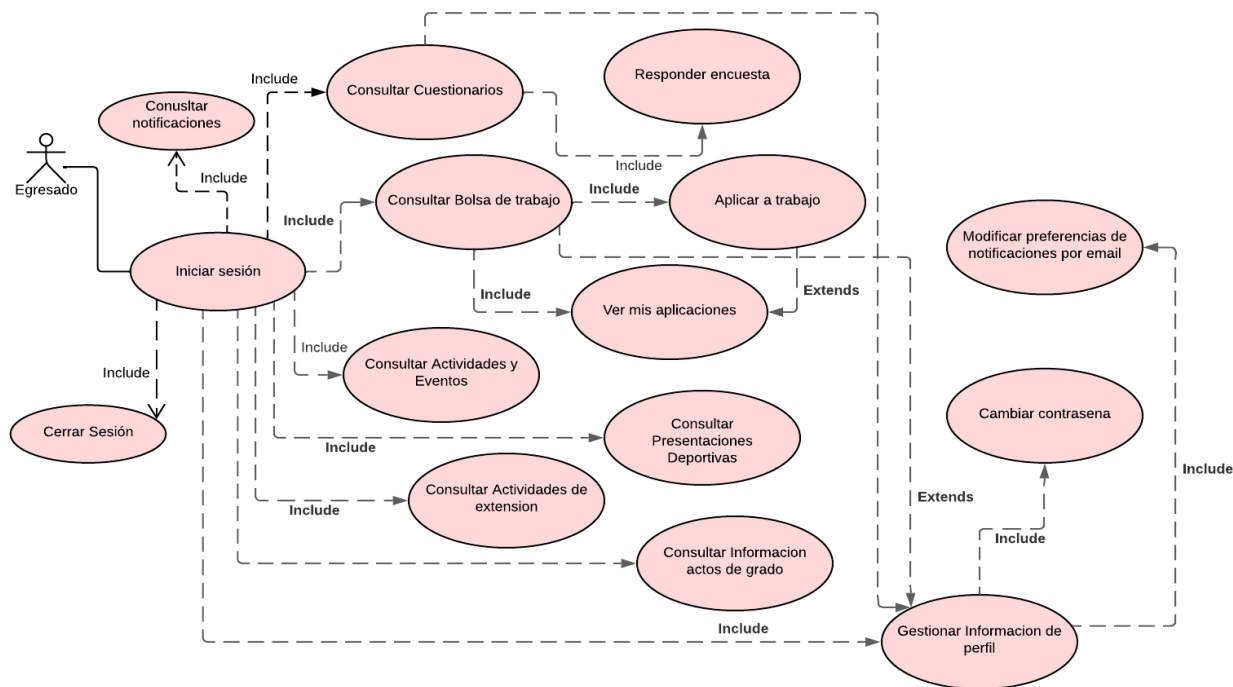
En este caso de uso, se describen las acciones que puede realizar un usuario administrador al momento de gestionar dichas actividades, estas abarcan desde crear, ver, editar y eliminar



Nombre del caso de uso: Gestión de actividades de extensión.	ID: CU07
Actores: Administrador	
Descripción: Permite al usuario administrador con permisos, Gestionar las actividades de extensión, que posteriormente se le mostraran al usuario egresado	
<p style="text-align: center;">Flujo normal</p> <p>El usuario administrador puede:</p> <ol style="list-style-type: none"> 1. Iniciar sesión. 2. Crear actividades de extensión. 3. Consultar información de actividades de extensión. 4. Modificar/Eliminar actividades de extensión. 5. Cerrar sesión. 	

Finalmente, el octavo caso de uso, y se detalla las actividades que puede realizar el usuario egresado al momento de ingresar al sistema.

Para este caso de uso, el usuario egresado puede consultar cuestionarios, así como responder las encuestas previamente creadas por el usuario administrador, además de consultar u optar por una o varias ofertas laborales, gestionar la información de su perfil y consultar las distintas actividades universitarias tales como, actos de grado, actividades de extensión, eventos culturales y presentaciones deportivas.



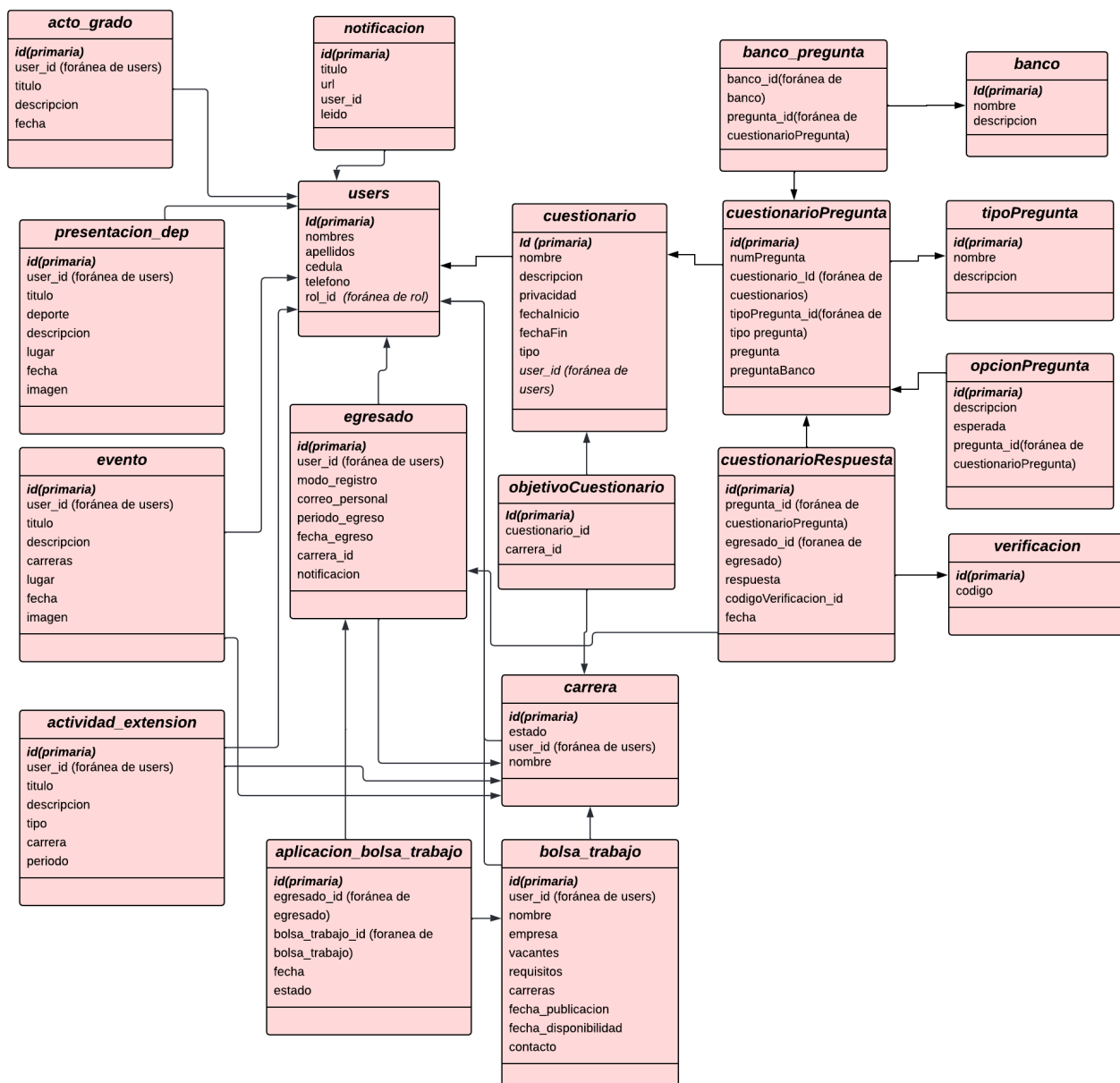
Nombre del caso de uso: Funciones del egresado.	ID: CU08
Actores: Egresado	
Descripción: Permite al usuario egresados, visualizar y gestionar ciertos módulos de interés.	
<p style="text-align: center;">Flujo normal</p> <p>El usuario administrador puede:</p> <ol style="list-style-type: none"> 1. Iniciar sesión. 2. Consultar notificaciones 3. Consultar cuestionarios. 4. Responder cuestionarios. 5. Consultar bolsa de trabajo. 6. Aplicar a trabajo. 7. Ver trabajos aplicados. 8. Consultar actividades y eventos. 9. Consultar presentaciones deportivas. 10. Consultar actividades de extensión. 11. Consultar información de actos de grado. 12. Gestionar información de perfil. 13. Cambiar contraseña. 14. Modificar preferencias de notificaciones por email. 15. Cerrar sesión. 	

DIAGRAMA RELACIONAL

El diagrama relacional está compuesto de las entidades y atributos que se crearon para el almacenamiento de datos del software. A continuación, se describe el objetivo de las tablas de la base de datos general:

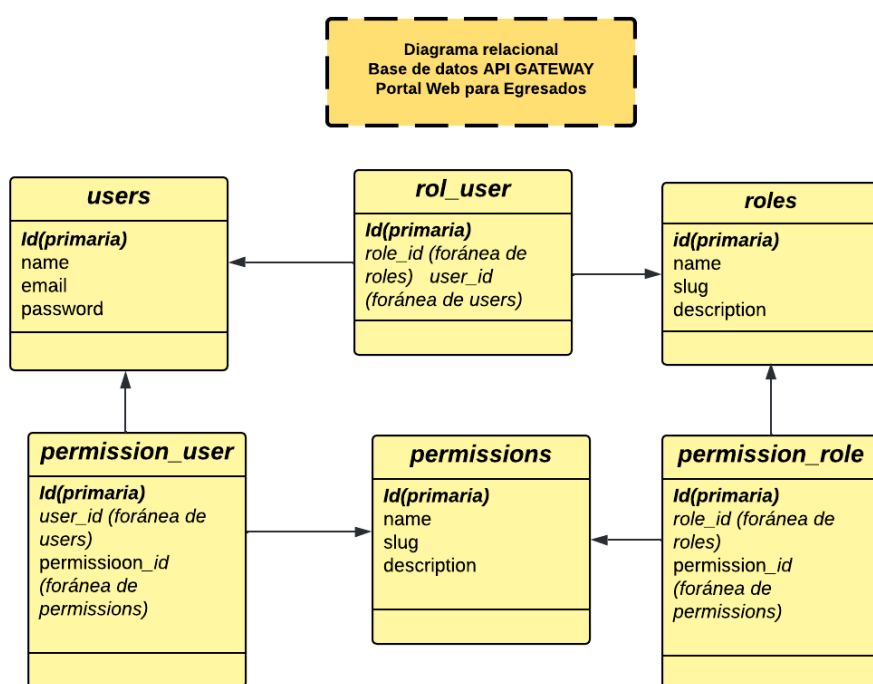
- **acto_grado:** tiene como objetivo almacenar los registros sobre los actos de grado para mantener al egresado informado.
- **presentación_dep:** almacena los registros de las presentaciones deportivas para luego presentarlas a los egresados.
- **evento:** almacena los eventos y actividades a realizar en la entidad con la finalidad de que los egresados puedan estar enterados de ellas.
- **actividad_extensión:** está pensada para las actividades de extensión para las distintas carreras.
- **bolsa_trabajo:** está pensada para almacenar las ofertas laborales y presentarlas a los egresados para que estos puedan optar por ellas.
- **aplicacion_bolsa_trabajo:** guarda la aplicación de un egresado a una oferta laboral de la bolsa de trabajo.
- **users:** almacena la información base de los usuarios, tanto administradores como egresados.
- **egresado:** almacena la información del perfil de los egresados.
- **notificacion:** almacena las notificaciones de los egresados en el sistema y su estatus.
- **carrera:** guarda las distintas carreras existentes o registradas en la Universidad Nacional Experimental de Guayana.
- **cuestionario:** busca tener la información base de los cuestionarios y encuestas para luego agregarle preguntas y publicarlo para ser respondida por los egresados.
- **objetivoCuestionario:** guarda a que carreras va dirigida un cuestionario en específico para que solo los de las carreras objetivo puedan ver y responder el cuestionario.
- **banco:** nos permite tener la información base de un banco de preguntas para luego agregarle las preguntas y utilizar el banco para agilizar la creación de un cuestionario u encuesta.
- **banco_pregunta:** tiene como objetivo relacionar las preguntas con un banco y así saber cuáles preguntas pertenecen a un banco de preguntas.
- **tipoPregunta:** tiene como objetivo almacenar los tipos de pregunta que puede haber en función a la respuesta esperada, siendo estos de selección simple, selección múltiple, respuesta de texto o respuesta numérica.
- **opcionPregunta:** almacena todas las opciones que puede tener una pregunta de selección múltiple.
- **cuestionarioPregunta:** tiene como objetivo almacenar la información base y relaciones de una pregunta para así poder usarla en un cuestionario o en una encuesta.
- **cuestionarioRespuesta:** el objetivo de esta tabla es almacenar las respuestas de los egresados a las preguntas de los cuestionarios o de las encuestas.
- **verificacion:** en la tabla de verificación se tienen los códigos de completación de los egresados sobre los cuestionarios y sobre las encuestas.

Diagrama relacional
Base de datos General
Portal Web para Egresados



37

- **users:** tiene como objetivo almacenar la información de autenticación de los usuarios del sistema, tanto egresados como administradores.
- **permissions:** en esta tabla se guardan todos los permisos a acciones que puede haber en los distintos módulos del sistema.
- **roles:** tiene como objetivo, con ayuda de los permisos, establecer roles que pueden tener los usuarios. Por defecto hay dos roles en el sistema, administrador y egresado.
- **permission_user:** permite añadir un permiso en específico a un usuario en específico.
- **permission_role:** tiene como objetivo almacenar todos los permisos que puede tener un rol en específico. El rol administrador por defecto tiene todos los permisos.
- **rol_user:** busca almacenar el rol perteneciente a cada usuario en el sistema.



DICCIONARIO DE DATOS

Con el fin de obtener una descripción más detallada de los datos procesados en los diagramas relacionales, se procede a crear el diccionario de datos, con la descripción de algunas de las tablas a incluir en la base de datos que será utilizada en el sistema a desarrollar. Entre estas tablas de los diagramas relacionales tenemos:

actividad_extensión: almacena información relacionada con las actividades de extensión. Entre los atributos de esta tabla tenemos la llave foránea `user_id` que hace referencia a la tabla de `users` con la finalidad de poder saber cuál usuario creó el registro de actividad de extensión en específico.

Columna	Tipo	Nulo	Único	Descripción
id (Primary)	int	No	Si	Identificador numérico del registro.
user_id (Foránea de users)	int	Si	No	Identificador numérico correspondiente al usuario que realizo el registro.
titulo	varchar	No	No	Título que se le asigna a la actividad de extensión
descripcion	varchar	Si	No	Descripción relacionada con la actividad de extensión
carrera	varchar	Si	No	Carreras a la que va dirigida esta actividad de extensión
tipo	varchar	No	No	Tipo de activad de extensión
imagen	varchar	Si	No	Imagen de la actividad de extensión
periodo	date	Si	No	Periodo de actividad de extensión
created_at	timestamp	Si	No	Fecha de creación del registro
updated_at	timestamp	Si	No	Fecha de modificación del registro

acto_grado: almacena información sobre los actos de grado. La tabla acto_grado tiene una llave foránea user_id referente a la tabla users , para así, poder saber cuál usuario creó un registro. Esta información se encuentra definida en el mostrado a continuación.

Columna	Tipo	Nulo	Único	Descripción
id (Primary)	int	No	Si	Identificador numérico del registro.
user_id (Foránea de users)	int	Si	No	Identificador numérico correspondiente al usuario que realizo el registro.
titulo	varchar	No	No	Título que se le asigna al acto de grado
descripcion	varchar	No	No	Descripción relacionada al acto de grado
fecha	date	Si	No	Fecha en la que se efectuara el acto de grado
created_at	timestamp	Si	No	Fecha de creación del registro
updated_at	timestamp	Si	No	Fecha de modificación del registro

banco: almacena los datos básicos de los bancos de preguntas, su información se puede observar en el [cuadro 14](#).

Columna	Tipo	Nulo	Único	Descripción
---------	------	------	-------	-------------

id (<i>Primary</i>)	<i>int</i>	No	Si	Identificador numérico del registro.
nombre	<i>varchar</i>	No	No	Nombre que se le asigna al banco de preguntas
descripcion	<i>varchar</i>	No	No	Descripción relacionada al banco de preguntas.
created_at	<i>timestamp</i>	Si	No	Fecha de creación del registro
updated_at	<i>timestamp</i>	Si	No	Fecha de modificación del registro

banco_pregunta: almacena la asociación de las preguntas a su banco de preguntas, esta asociación se logra con sus llaves foráneas banco_id que es llave foránea con respecto a la tabla banco para saber a qué banco pertenece y pregunta_id referente a la tabla cuestionarioPregunta para tener la relación y conocer cuál es la pregunta del banco. Su representación se puede visualizar en el.

Columna	Tipo	Nulo	Único	Descripción
id (<i>Primary</i>)	<i>int</i>	No	Si	Identificador numérico del registro.
banco_id (Foránea de banco)	<i>int</i>	No	No	Identificador numérico del banco de preguntas
pregunta_id (Foránea de cuestionarioPregunta)	<i>int</i>	No	No	Identificador numérico de la pregunta.
created_at	<i>timestamp</i>	Si	No	Fecha de creación del registro
updated_at	<i>timestamp</i>	Si	No	Fecha de modificación del registro

aplicación_bolsa_egresado: almacena información referente a las aplicaciones de ofertas laborales realizadas por el egresado, esta relación la logra con sus llaves foráneas egresado_id que hace referencia a la tabla de egresados y bolsa_trabajo_id que referencia a la tabla bolsa_trabajo para relacionarlo con un trabajo en específico.

Columna	Tipo	Nulo	Único	Descripción
id (<i>Primary</i>)	<i>int</i>	No	Si	Identificador numérico del registro.
egresado_id (Foránea de egresado)	<i>int</i>	No	No	Identificador numérico del egresado
bolsa_trabajo_id	<i>int</i>	No	No	Identificador numérico del trabajo

(Foránea de bolsa_trabajo)				
estado	<i>varchar</i>	No	No	Estado de la solicitud del egresado
fecha	<i>date</i>	No	No	Fecha en la que aplico el egresado
created_at	<i>timestamp</i>	Si	No	Fecha de creación del registro
updated_at	<i>timestamp</i>	Si	No	Fecha de modificación del registro

bolsa_trabajo: almacena informacion referente a las ofertas laborales. La tabla cuenta con la llave foránea user_id como relación con la tabla users para poder saber cuál usuario creó un registro en específico.

Columna	Tipo	Nulo	Único	Descripción
id (<i>Primary</i>)	<i>int</i>	No	Si	Identificador numérico del registro.
user_id (Foránea de users)	<i>int</i>	No	No	Identificador numérico correspondiente al usuario que realizo el registro.
nombre	<i>varchar</i>	No	No	Nombre de la oferta laboral
empresa	<i>varchar</i>	No	No	Nombre de la empresa
vacantes	<i>varchar</i>	Si	No	Cantidad de vacantes
requisitos	<i>varchar</i>	No	No	Requisitos para la oferta laboral
carreras	<i>varchar</i>	No	No	Carreras solicitadas para la oferta laboral
fecha_publicacion	<i>date</i>	Si	No	Fecha de publicación de la oferta laboral
fecha_disponibilidad	<i>date</i>	Si	No	Fecha de disponibilidad de la oferta laboral
contacto	<i>varchar</i>	Si	No	Contacto directo con la persona que solicita un empleado
created_at	<i>timestamp</i>	Si	No	Fecha de creación del registro
updated_at	<i>Timestamp</i>	Si	No	Fecha de modificación del registro

carrera: almacena información relacionada a las carreras disponibles o existentes en la Universidad Nacional Experimental de Guayana. Tiene una llave foránea user_id que referencia a la tabla de usuarios .

Columna	Tipo	Nulo	Único	Descripción
---------	------	------	-------	-------------

id (<i>Primary</i>)	<i>int</i>	No	Si	Identificador numérico del registro.
nombre	<i>varchar</i>	No	No	Nombre de la carrera
user_id (Foránea de users)	<i>int</i>	No	No	Identificador numérico correspondiente al usuario que realizó el registro.
estado	<i>varchar</i>	No	No	Si la carrera esta activa o inactiva
created_at	<i>timestamp</i>	Si	No	Fecha de creación del registro
updated_at	<i>Timestamp</i>	Si	No	Fecha de modificación del registro

cuestionario: almacena información base sobre los cuestionarios. Esta tabla cuenta con la llave foránea objetivo_id que hace referencia a la tabla objetivoCuestionario con el fin de saber a cuáles egresados va dirigido el cuestionario. Adicionalmente, tiene la llave foránea user_id en el que se guarda el id del usuario que creó el registro.

Columna	Tipo	Nulo	Único	Descripción
id (<i>Primary</i>)	<i>int</i>	No	Si	Identificador numérico del registro.
nombre	<i>varchar</i>	No	No	Título que se le asigna al cuestionario
descripcion	<i>varchar</i>	No	No	Descripción relacionada al cuestionario
tipo	<i>varchar</i>	No	No	Tipo de cuestionario
privacidad	<i>varchar</i>	No	No	Estatus de publicación del cuestionario
objetivo_id (Foránea de objetivoCuestionario)	<i>int</i>	No	No	Identificador numérico del objetivo del cuestionario
user_id (Foránea de users)	<i>int</i>	Si	No	Identificador numérico correspondiente al usuario que realizó el registro.
fecha_inicio	<i>date</i>	No	No	Fecha en la que el formulario empezara a ser visible para el egresado
fecha_fin	<i>date</i>	Si	No	Fecha en la que el formulario dejara de ser visible para el egresado
created_at	<i>timestamp</i>	Si	No	Fecha de creación del registro
updated_at	<i>timestamp</i>	Si	No	Fecha de modificación del registro

cuestionario_pregunta: almacena la asociación de cuestionario con sus preguntas. Por medio de sus llaves foráneas cuestionario_id y tipoPregunta_id podemos saber a cuál cuestionario pertenece la pregunta y qué tipo de pregunta es.

Columna	Tipo	Nulo	Único	Descripción
id (<i>Primary</i>)	<i>int</i>	No	Si	Identificador numérico del registro.
numPregunta	<i>int</i>	No	No	Numero de pregunta
cuestionario_id (Foránea de cuestionario)	<i>int</i>	No	No	Identificador numérico del cuestionario
tipoPregunta_id (Foránea de tipoPregunta)	<i>int</i>	No	No	Identificador numérico del tipo de pregunta
pregunta	<i>varchar</i>	No	No	Nombre que se le asignara a la pregunta
preguntaBanco	<i>varchar</i>	Si	No	Estado y referencia a si vino o no de un banco pregunta
created_at	<i>timestamp</i>	Si	No	Fecha de creación del registro
updated_at	<i>Timestamp</i>	Si	No	Fecha de modificación del registro

cuestionario_respuesta: almacenar información referente a la pregunta respondida por el usuario egresado. Tiene tres llaves foráneas pregunta_id que referencia a la tabla cuestionarioPregunta, egresado_id relacionado con la tabla de egresado y codigoVerificacion_id users (detallada en el) que referencia a la tabla verificación. La tabla se puede detallar en el.

Columna	Tipo	Nulo	Único	Descripción
id (<i>Primary</i>)	<i>int</i>	No	Si	Identificador numérico del registro.
pregunta_id (Foránea de cuestionarioPregunta)	<i>int</i>	No	No	Identificador numérico de la pregunta
egresado_id (Foránea de egresado)	<i>int</i>	No	No	Identificador numérico del egresado que responde la pregunta
codigoVerificacion_id (Foránea de verificacion)	<i>int</i>	No	No	Identificador del código de verificación del cuestionario
fecha	<i>date</i>	No	No	Fecha que respondió la pregunta

respuesta	<i>varchar</i>	No	No	Respuesta de la pregunta
created_at	<i>timestamp</i>	Si	No	Fecha de creación del registro
updated_at	<i>timestamp</i>	Si	No	Fecha de modificación del registro

egresado: Almacena información relacionada con el egresado, esta información está representada en el. Esta tabla tiene la llave foránea user_id referente a la tabla users para saber que usuario hizo el registro del egresado y carrera_id para saber cuál es la carrera cursada por el egresado.

Columna	Tipo	Nulo	Único	Descripción
id (<i>Primary</i>)	int	No	Si	Identificador numérico del registro.
user_id (Foránea de users)	int	No	No	Identificador numérico correspondiente al usuario que realizo el registro.
modo_registro	varchar	No	No	Modo en el que se hizo el registro del egresado
correo_personal	varchar	No	No	Correo electrónico del egresado
periodo_egreso	varchar	No	No	Periodo de egreso del egresado
fecha_egreso	date	Si	No	Fecha de egreso del egresado
notificacion	tinyint(1)	No	No	Estado utilizado para recibir o no notificaciones en el correo del egresado
carrera_id (Foránea de carrera)	int	No	No	Identificador numérico de la carrera cursada por el egresado
created_at	timestamp	Si	No	Fecha de creación del registro
updated_at	timestamp	Si	No	Fecha de modificación del registro

evento: Almacena información relacionada a los eventos. Tiene la llave foránea user_id relacionada a la tabla users con el fin de saber cuál usuario manipulo el registro.

Columna	Tipo	Nulo	Único	Descripción
id (<i>Primary</i>)	int	No	Si	Identificador numérico del registro.
user_id (Foránea de users)	int	Si	No	Identificador numérico correspondiente al usuario que realizo el registro.
titulo	<i>varchar</i>	No	No	Título asignado al evento
descripcion	<i>varchar</i>	Si	No	Descripción referente al evento

carreras	<i>varchar</i>	Si	No	Carreras a las que se ofrece el evento
lugar	<i>varchar</i>	No	No	Lugar donde se realizará el evento
imagen	<i>varchar</i>	Si	No	Imagen del evento
fecha	<i>date</i>	Si	No	Fecha en la que se realizara el evento
created_at	<i>timestamp</i>	Si	No	Fecha de creación del registro
updated_at	<i>Timestamp</i>	Si	No	Fecha de modificación del registro

objetivo_cuestionario: almacenar los datos relacionados con los objetivos del cuestionario, estos datos se pueden apreciar en el. La tabla tiene dos llaves foráneas cuestionario_id relacionada con la tabla cuestionario y carrera_id referente a la tabla carreras;

CUADRO 2. DICCIONARIO DE DATOS: "OBJETIVO_CUESTIONARIO"

Columna	Tipo	Nulo	Único	Descripción
id (<i>Primary</i>)	<i>int</i>	No	Si	Identificador numérico del registro.
cuestionario_id (Foránea de cuestionario)	<i>int</i>	No	No	Identificador numérico del cuestionario
carrera_id (Foránea de carrera)	<i>int</i>	No	No	Identificador numérico de la carrera

opcion_pregunta: Almacena las opciones de respuesta a las preguntas creadas. Tiene la llave foránea pregunta_id referente a la tabla cuestionarioPregunta para tener la relación de cuál opción corresponde a cuál pregunta.

Columna	Tipo	Nulo	Único	Descripción
id (<i>Primary</i>)	<i>int</i>	No	Si	Identificador numérico del registro.
nombre	<i>varchar</i>	No	No	Nombre de la opción de pregunta
esperada	<i>varchar</i>	No	No	Opción de pregunta esperada
pregunta_id (Foránea de cuestionarioPregunta)	<i>int</i>	No	No	Identificador numérico de la pregunta
created_at	<i>timestamp</i>	Si	No	Fecha de creación del registro
updated_at	<i>Timestamp</i>	Si	No	Fecha de modificación del registro

presentacion_dep: Almacena información referente a las presentaciones deportivas. La tabla cuenta con la llave foránea user_id relacionada con la tabla users users para saber qué usuario manipuló el registro. La información de esta tabla está disponible en el.

Columna	Tipo	Nulo	Único	Descripción
id (Primary)	int	No	Si	Identificador numérico del registro.
user_id (Foránea de users)	int	Si	No	Identificador numérico correspondiente al usuario que realizo el registro.
titulo	varchar	No	No	Título asignado a la presentación deportiva
descripcion	varchar	Si	No	Descripción de la presentación deportiva
deporte	varchar	No	No	Nombre del deporte de la presentación deportiva
lugar	varchar	No	No	Lugar donde ocurrirá la presentación deportiva
imagen	varchar	Si	No	Imagen de la presentación deportiva
fecha	date	Si	No	Fecha de la presentación deportiva
created_at	timestamp	Si	No	Fecha de creación del registro
updated_at	Timestamp	Si	No	Fecha de modificación del registro

tipo_pregunta: La información que almacena está relacionada con los tipos de preguntas.

Columna	Tipo	Nulo	Único	Descripción
id (Primary)	int	No	Si	Identificador numérico del registro.
nombre	varchar	No	No	Nombre asignado al tipo de pregunta
descripcion	varchar	No	No	Descripción del tipo de pregunta
created_at	timestamp	Si	No	Fecha de creación del registro
updated_at	Timestamp	Si	No	Fecha de modificación del registro

verificacion: almacena los códigos de verificación generados al completar un formulario.

Columna	Tipo	Nulo	Único	Descripción
id (Primary)	int	No	Si	Identificador numérico del registro.
codigo	varchar	No	No	Código generado al completar una encuesta
created_at	timestamp	Si	No	Fecha de creación del registro
updated_at	Timestamp	Si	No	Fecha de modificación del registro

notificación: almacena las notificaciones de los usuarios. Esta tabla cuenta con la llave foránea user_id relacionada con la tabla users para saber a cuál usuario pertenece una notificación.

Columna	Tipo	Nulo	Único	Descripción
id (Primary)	int	No	Si	Identificador numérico del registro.
user_id (Foránea de users)	int	No	No	Identificador numérico correspondiente al usuario al que pertenece la notificación.
url	varchar	Si	No	Ruta asociada a la notificación
titulo	varchar	No	No	Título de la notificación que se le mostrara al usuario
leido	tinyint(1)	No	No	Bandera que muestra si la notificación ya fue vista por el usuario o no.
created_at	timestamp	Si	No	Fecha de creación del registro
updated_at	Timestamp	Si	No	Fecha de modificación del registro

users: almacena los datos básicos de los usuarios del sistema, los detalles de esta tabla los encontramos en el.

Columna	Tipo	Nulo	Único	Descripción
id (Primary)	int	No	Si	Identificador numérico del registro.
nombres	varchar	No	No	Nombres del usuario.
apellidos	varchar	No	No	Apellidos del usuario.
cedula	varchar	No	No	Número de documento de identidad del usuario.
telefono	varchar	Si	No	Número telefónico del usuario.
created_at	timestamp	Si	No	Fecha de creación del registro
updated_at	Timestamp	Si	No	Fecha de modificación del registro

permissions (API Gateway): almacena los distintos permisos u opciones a las que puede acceder un usuario en el sistema.

Columna	Tipo	Nulo	Único	Descripción
id (Primary)	int	No	Si	Identificador numérico del registro.
name	varchar	No	No	Nombre del permiso

slug	<i>varchar</i>	No	Si	Nombre identificador del permiso
description	<i>varchar</i>	Si	No	Descripción del permiso
created_at	<i>timestamp</i>	Si	No	Fecha de creación del registro
updated_at	<i>Timestamp</i>	Si	No	Fecha de modificación del registro

roles (API Gateway): almacena los roles que no son más que un conjunto de permisos que puede tener un usuario.

Columna	Tipo	Nulo	Único	Descripción
id (<i>Primary</i>)	<i>int</i>	No	Si	Identificador numérico del registro.
name	<i>varchar</i>	No	No	Nombre del rol.
slug	<i>varchar</i>	No	Si	Nombre identificador del rol
description	<i>varchar</i>	Si	No	Descripción del rol
created_at	<i>timestamp</i>	Si	No	Fecha de creación del registro
updated_at	<i>Timestamp</i>	Si	No	Fecha de modificación del registro

permission_role (API Gateway): almacena los datos de la relación de cada rol con sus permisos correspondientes, esto lo logra con sus llaves foráneas *role_id* que referencia a la tabla *roles* y *permission_id* relacionada con la tabla *permissions*.

Columna	Tipo	Nulo	Único	Descripción
id (<i>Primary</i>)	<i>int</i>	No	Si	Identificador numérico del registro.
role_id (Foránea – roles)	<i>int</i>	No	No	Identificador del rol.
permission_id (Foránea – permissions)	<i>int</i>	No	No	Identificador del permiso.

users (API Gateway): almacena los datos de autenticación de los usuarios del sistema dentro de la base de datos del API Gateway,.

Columna	Tipo	Nulo	Único	Descripción
id (<i>Primary</i>)	<i>int</i>	No	Si	Identificador numérico del registro.
email	<i>varchar</i>	No	Si	Correo del usuario.
password	<i>varchar</i>	No	No	Contraseña del usuario.

name	<i>varchar</i>	No	No	Nombre del usuario.
created_at	<i>timestamp</i>	Si	No	Fecha de creación del registro
updated_at	<i>Timestamp</i>	Si	No	Fecha de modificación del registro

permission_user (API Gateway): Almacena permisos específicos para un usuario, esto lo logra con sus llaves foráneas user_id relacionada con la tabla users y permission_id relacionada a la tabla permissions.

Columna	Tipo	Nulo	Único	Descripción
id (Primary)	<i>int</i>	No	Si	Identificador numérico del registro.
user_id (Foranea – users)	<i>int</i>	No	No	Identificador del usuario.
permission_id (Foranea – permissions)	<i>int</i>	No	No	Identificador del permiso.

rol_user (API Gateway): Almacena los roles específicos para los usuarios. Cuenta con dos llaves foráneas, user_id referenciando con la tabla users y role_id relacionada a la tabla roles.

Columna	Tipo	Nulo	Único	Descripción
id (Primary)	<i>int</i>	No	Si	Identificador numérico del registro.
user_id (Foranea – users)	<i>int</i>	No	No	Identificador del usuario.
role_id (Foranea – roles)	<i>int</i>	No	No	Identificador del rol.

DEFINICIÓN DE ARQUITECTURA

Como arquitectura del software se optó por una arquitectura basada en microservicios, esto debido a la necesidad construir una aplicación tolerante a las fallas que inhabilite todas las funcionalidades de este, dado que, al trabajar directamente con los microservicios, si hay un error en algún servicio en específico, se puede degradar dicha funcionalidad, sin bloquear toda la aplicación.

Esta arquitectura consta de:

- Una interfaz de usuario a desarrollar bajo HTML, CSS, TypeScript y JavaScript, a través del *framework* de Angular, utilizando como base de diseño las plantillas del tema *fuse admin template* en su versión 14.
- Un *API Gateway* en PHP bajo el micro *framework* de Lumen para la redirección de las peticiones a los microservicios y para la autenticación de los usuarios.
- Un microservicio desarrollado con Python a través del micro *framework* de Flask a beneficio de la arquitectura planteada, el cual, generará las estadísticas y los datos en función a los resultados de los cuestionarios y las encuestas realizadas.
- Un microservicio en PHP bajo el micro *framework* de Lumen para la gestión de los registros de la información en los distintos módulos del sistema.
- Dos bases de datos con MySQL, una con la información de los datos que se gestionaran desde los microservicios y la segunda con información de la autenticación de usuarios gestionada a través del *API Gateway*.

De acuerdo a lo anteriormente mencionado, en la siguiente figura se observa la arquitectura del sistema definida, en conjunto a las herramientas y lenguajes seleccionados para su desarrollo.

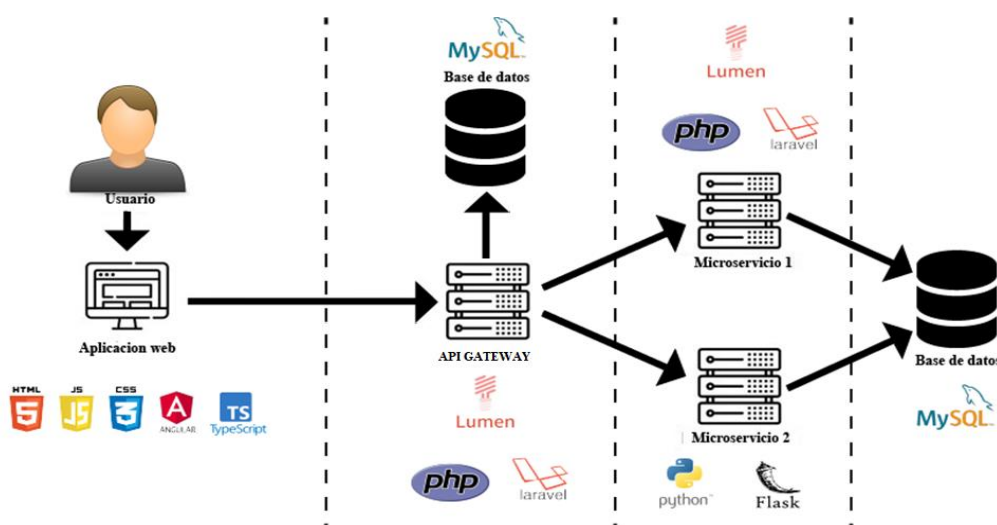


DIAGRAMA NAVEGACIONAL

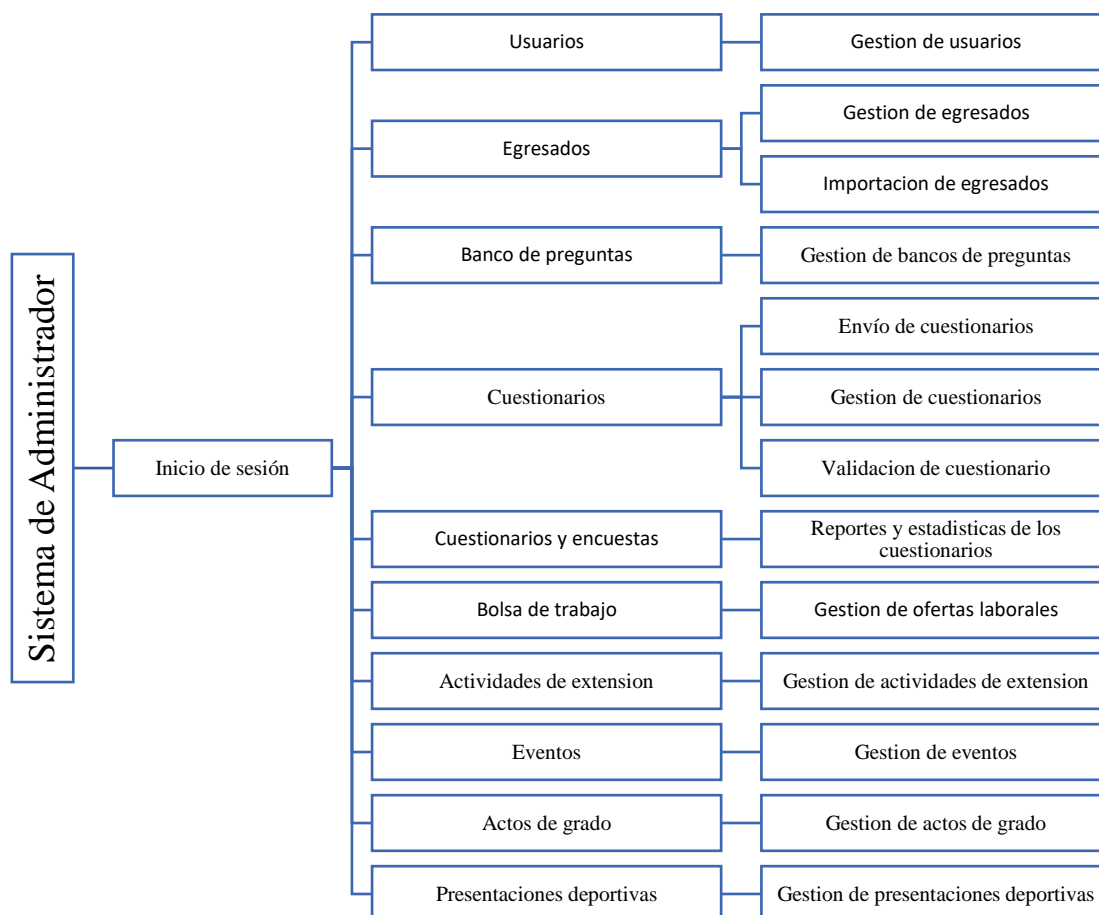
Para obtener una visión más amplia, sencilla y concreta de los módulos y funcionalidades del sistema, se diseñan dos diagramas navegacionales divididos por los actores que intervienen en cada funcionalidad.

Estos diagramas consisten en mostrar los caminos o módulos de navegación en una forma jerárquica, expresando con ello, cómo los usuarios de una perspectiva particular pueden navegar a través de la información disponible.

El primer diagrama navegacional, es del sistema de administrador, en este diagrama, el usuario administrador una vez inicie sesión en la aplicación podrá ingresar a cualquiera de los módulos detallados a continuación:

1. **Módulo de usuarios:** Contiene la información relacionada con los usuarios administradores del sistema. Este módulo está dedicado a gestionar los usuarios administradores, lo cual incluye visualizar, crear, agregar y eliminar un usuario.
2. **Egresados:** Abarca la información relacionada con los usuarios egresados. En este módulo el usuario administrador podrá desde importar de forma masiva los egresados, así como listarlos, editarlos, crearlos o eliminarlos.
3. **Banco de preguntas:** Este módulo está orientado a predefinir preguntas que serán utilizadas posteriormente en la creación o edición de los cuestionarios y encuestas. En este módulo el usuario administrador podrá crear, editar o eliminar preguntas o bancos de preguntas.
4. **Cuestionarios:** Está destinado a la gestión de encuestas y cuestionarios dirigidos a los egresados universitarios, esto abarca desde la creación, edición y eliminación de cuestionarios por parte del usuario administrador, hasta el envío de encuestas vía correo electrónico a uno o varios egresados, y a la validación del código de las encuestas que proporcionan los egresados una vez culminen un cuestionario.
5. **Cuestionarios y Encuestas:** Contiene el procesamiento de datos necesarios para la generación de reportes y estadísticas. En este módulo, el usuario administrador podrá ver, evaluar y comparar mediante diversas gráficas, como de barra, lineal o de dispersión, las respuestas de los cuestionarios completados.
6. **Bolsa de trabajo:** Su funcionalidad está destinada a la gestión de ofertas laborales dirigidas a los egresados universitarios. En este módulo, el usuario administrador podrá crear, visualizar, editar y eliminar las diversas ofertas.
7. **Actividades de extensión:** En este módulo, el usuario administrador podrá crear, editar o eliminar actividades relacionadas con la carrera del egresado, como cursos, doctorados, postgrados, entre otros.
8. **Eventos:** Abarca la creación, edición, visualización y eliminación de eventos con temáticas libres realizadas en la universidad, como ferias de libro, presentaciones de baile, ente otros eventos u actividades.
9. **Actos de grados:** Permite al usuario administrador, crear, editar o eliminar información relacionada con los próximos actos de grado de la institución universitaria.

10. Presentaciones deportivas: En este módulo el usuario administrador podrá crear, editar o eliminar información relacionada con eventos que se realizaran en la universidad.



Como segundo diagrama navegacional, representado en la se puede apreciar el sistema de egresado, en este diagrama, el usuario con rol de egresado una vez inicie sesión en la aplicación podrá navegar entre los distintos módulos detallados a continuación:

1. Cuestionarios: En este módulo el usuario egresado podrá visualizar los cuestionarios disponibles para ser contestados, así como también, podrá obtener el código de completación de estos.
2. Bolsa laboral: Abarca información sobre las propuestas laborales disponibles, en este módulo el usuario egresado podrá visualizar y aplicar a las ofertas disponibles de su interés.
3. Eventos: Es un módulo de carácter informativo que contiene información relacionada con los eventos de carácter libre organizados en la Universidad.
4. Actividades de extensión: En este módulo el usuario egresado podrá visualizar las actividades relacionadas con la carrera.
5. Presentaciones deportivas: Incluye información relacionada a las actividades deportivas realizadas en la universidad.
6. Actos de grado: Es un módulo de carácter informativo el cual contiene información relacionada con los próximos actos de grado.
7. Perfil: Este módulo permitirá al usuario egresado, ver o editar la información disponible en su perfil, bien sea cambiar la clave del usuario o modificar el envío de notificaciones a su correo electrónico.

