

# Manual de Instalación

## Requerimientos:

Python 3.8 o 3.9  
Gunicorn 20+  
Python-virtualenv

Una vez instalado previamente los requerimientos procedemos a copiar todos los archivos de la aplicación a la carpeta `/usr/share/nginx/quality_system` verificamos que todos los archivos se encuentren en la nueva ubicación y procedemos a crear y activar el entorno virtual

## Creación y activación del entorno virtual

Ejecutamos los siguientes comandos:

```
virtualenv env
```

```
source env/bin/activate
```

## Instalar dependencias

Una vez el entorno virtual se encuentre activo procedemos a realizar la instalación de las librerías necesarias para el funcionamiento de la aplicación con el siguiente comando:

```
pip install -r requirements.txt
```

## Configuración Cron

Debido que la aplicación ejecuta algunas tareas periódicas es necesario configurar el cron para que ejecute algunos scripts periódicamente, para hacerlo añadimos las siguientes líneas en el cron

```
# ejecuta las alertas todos los días a las 8 am
0 8 * * * /usr/share/nginx/quality_system/server_configurations/quality.sh
# Backup base de datos postgres
0 21 * * * /usr/share/nginx/quality_system/server_configurations/backupDB.sh
```

## Creación del servicio para la aplicación

Para crear el servicio con distribuciones que usan systemd creamos un archivo llamado **gunicorn\_quality.service** en la siguiente ruta **/etc/systemd/system** este archivo debe tener el siguiente contenido:

```
[Unit]
Description=Run Gunicorn for quality system app
After=default.target

[Service]
Type=simple
RemainAfterExit=yes
ExecStart=/usr/share/nginx/quality_system/gunicorn_start.sh
TimeoutStartSec=0

[Install]
WantedBy=default.target
```

## Instalación Base de Datos

Una vez tengamos el motor de base de datos postgres previamente funcionando procedemos a crear la base de datos y el usuario

```
create database quality_system;
CREATE USER quality WITH PASSWORD 'xxxxxxxxxx';
GRANT ALL PRIVILEGES ON DATABASE quality_system to quality;

psql -U quality -d quality_system < backup.sql
```