

# Manual de instalación y operación GDevLink

Deltasoft

## Contenido

PARTE 1: INSTALACIÓN Y DESPLIEGE .....	1
1. Instalación de Python .....	1
2. Creación y activación entorno virtual .....	2
3. Clonar repositorio de github .....	2
4. Instalación de dependencias .....	2
5. Instalación y creación base de datos Postgresql .....	3
6. Iniciación de la base de datos .....	5
7. Ejecutar localmente .....	5
8. Despliegue .....	5
PARTE 2: OPERACIÓN .....	7
1. Super usuario: .....	7
2. Ventana de administración: .....	7
3. Cambios posteriores: .....	8

## PARTE 1: INSTALACIÓN Y DESPLIEGE

### 1. Instalación de Python

Para instalar Python en Windows, dirigirse a la página <https://www.python.org/downloads/> y descargar Python 3.8.6. Posteriormente se realiza la instalación de Python, dándole a la casilla de agregarlo al Path de las variables de entorno.

En Linux, algunas distribuciones ya vienen con Python preinstalado, para verificar su instalación, desde la terminal, ejecutar `python3 --versión`. En caso de no estar instalado, ejecutar los siguientes comandos:

```
sudo apt-get update
sudo apt-get install python3.8
```

## 2. Creación y activación entorno virtual

Desde la terminal de Linux, o cmd de Windows, ejecutar el siguiente comando para instalar el entorno virtual:

```
pip install virtualenv
```

Luego, para crear el entorno virtual, se debe especificar el nombre del directorio, donde se debe crear:

```
virtualenv "directorio"
```

Por último, se debe activar el entorno virtual. Esto se debe hacer cada vez que se deba ejecutar la aplicación de manera local. Se hace desde la terminal:

En Linux:

```
source mypython/bin/activate
```

En Windows:

```
"directorio"\Scripts\activate
```

## 3. Clonar repositorio de github

Es necesario tener git instalado. En Windows puede ser instalado mediante el link <https://git-scm.com/download/win>. En el caso de Ubuntu, ejecutar desde la consola:

```
sudo apt-get update $ sudo apt-get install git
```

Con git instalado, se ejecuta el siguiente comando:

```
git clone https://github.com/IngSw2030/GDevLink.git
```

## 4. Instalación de dependencias

Estando en la consola de comandos, se debe estar en la carpeta GDevLink, que es la carpeta donde están los archivos propios de la aplicación. Esta se encuentra dentro del propio repositorio GDevLink. Se identifica porque contiene los siguientes archivos:

📁 __pycache__	3/10/2020 11:12 a. m.	Carpeta de archivos	
📁 comunicacion	27/11/2020 2:27 p. m.	Carpeta de archivos	
📁 GDevLink	27/11/2020 2:27 p. m.	Carpeta de archivos	
📁 main	27/11/2020 2:27 p. m.	Carpeta de archivos	
📁 media	27/11/2020 2:27 p. m.	Carpeta de archivos	
📁 posicionVacante	27/11/2020 2:27 p. m.	Carpeta de archivos	
📁 preguntas	27/11/2020 2:27 p. m.	Carpeta de archivos	
📁 proyectos	27/11/2020 2:27 p. m.	Carpeta de archivos	
📁 staticfiles	26/11/2020 10:19 p. m.	Carpeta de archivos	
📁 usuarios	27/11/2020 2:27 p. m.	Carpeta de archivos	
📄 .gitignore	27/11/2020 2:27 p. m.	Documento de tex...	1 KB
📄 manage	27/11/2020 2:27 p. m.	Archivo PY	1 KB
📄 Procfile	27/11/2020 2:27 p. m.	Archivo	1 KB
📄 requirements	27/11/2020 2:27 p. m.	Documento de tex...	1 KB
📄 requirementsWindows	29/11/2020 6:28 p. m.	Documento de tex...	2 KB
📄 runtime	27/11/2020 2:27 p. m.	Documento de tex...	1 KB

Para que la consola de comandos este en la carpeta, ejecutar el siguiente comando:

```
cd "directorio del repositorio" \GDevLink\GDevLink
```

Como en la siguiente imagen:

```
C:\Users\santi\demo\Scripts>cd C:\Users\santi\Documents\GitHub\GDevLink\GDevLink
```

Posteriormente se deben instalar todas las dependencias de la aplicación. Sí se desear ejecutar la aplicación desde un sistema Unix, ejecutar el siguiente comando:

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

Si es el caso de Windows, entonces ejecutar el siguiente comando:

```
Pip install -r requirementsWindows.txt
```

Lo anterior, se encarga de instalar todas las dependencias de la aplicación, como Django y asgiref.

## 5. Instalación y creación base de datos Postgresql

### Instalación

Para Windows, se puede descargar e instalar desde el siguiente enlace: <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>. Para

el caso de Linux, la instalación depende de la distribución. Para Ubuntu, se debe ejecutar el comando:

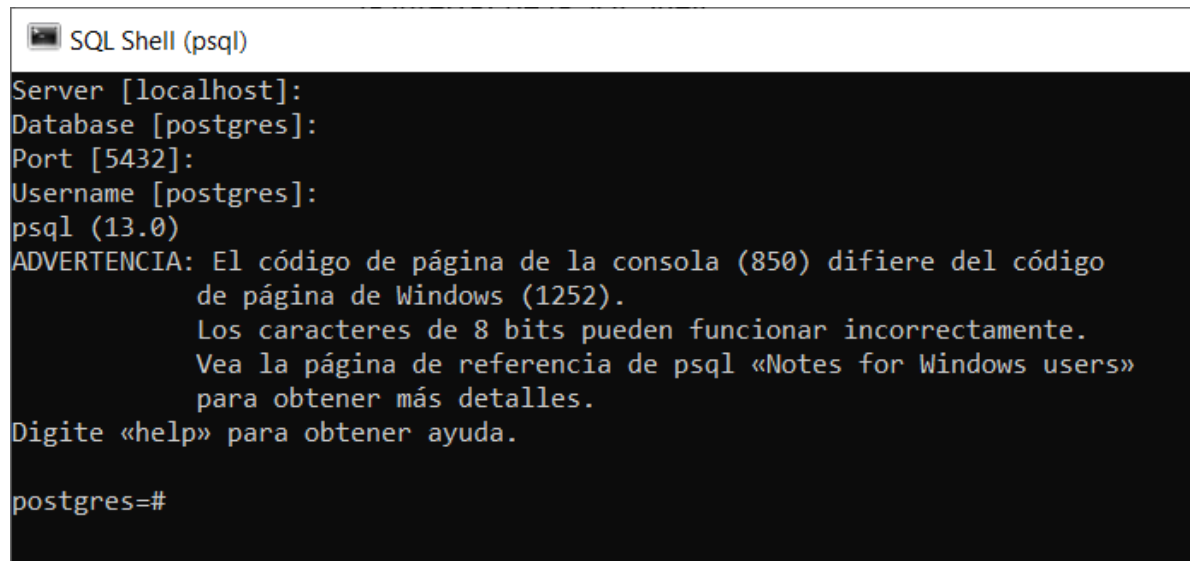
```
sudo apt-get install postgresql postgresql-contrib
```

Si se está usando Windows, primero se debe ejecutar el siguiente comando:

```
setx PATH "%PATH%;C:\Program Files\PostgreSQL\9.3\bin
```

### Creación base de datos

Se debe entrar a la consola de Postgresql escribiendo el comando **psql**, o alternativamente en Windows, usar el SQL Shell (Ingresar en la barra de búsqueda) de PostgreSQL. En SQL Shell, se debe dar presionar la tecla enter en los campos solicitados (server, database, port, y username). En la siguiente imagen se muestra la interfaz de la SQL Shell:



```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
psql (13.0)
ADVERTENCIA: El código de página de la consola (850) difiere del código
de página de Windows (1252).
Los caracteres de 8 bits pueden funcionar incorrectamente.
Vea la página de referencia de psql «Notes for Windows users»
para obtener más detalles.
Digite «help» para obtener ayuda.
postgres=#
```

Se ingresa el siguiente comando para crear un usuario. Todos los comandos de postgresql, deben terminar en ";".

```
CREATE user deltasoft;
```

Para la creación de la base de datos, se ejecuta el siguiente comando:

```
CREATE DATABASE gdevlink OWNER deltasoft;
```

Es importante que la base de datos tenga el nombre de gdevlink, y que el usuario dueño, se llame deltasoft, porque son los datos que están configurados por defecto en la aplicación.

## 6. Iniciación de la base de datos

Desde la consola de comandos, ubicados en el directorio de la aplicación, se deben ejecutar los siguientes comandos:

```
python manage.py migrate
python manage.py makemigrations
```

## 7. Ejecutar localmente

Para ejecutar de manera local la aplicación, simplemente se debe ejecutar el siguiente comando:

```
python manage.py runserver
```

posteriormente, se debe abrir un navegador web, y pegar la siguiente dirección en el buscador: 127.0.0.1:8000/

## 8. Despliegue

Para este tutorial, se muestra como desplegar la aplicación en Heroku. Primero se debe crear una cuenta en el siguiente enlace: <https://signup.heroku.com/login>.

Luego se debe instalar heroku-cli. En Windows se puede descargar desde <https://devcenter.heroku.com/articles/heroku-cli#download-and-install>. En el caso de Ubuntu, se ejecuta el siguiente comando:

```
sudo snap install --classic heroku
```

Posteriormente desde la consola (Estando en el directorio de la aplicación) ejecutar los siguientes comandos:

```
git init
git add .
git commit -m "deploy"
```

Luego es necesario autenticarse con la cuenta creada en Heroku. Para lo cual se ejecuta el comando:

```
Heroku login
```

El cual pedirá que se introduzca una tecla diferente de q, al hacerlo se abre una pestaña en su navegador predeterminado, en la página de inicio de sesión de heroku.

Una vez se ha hecho el inicio de sesión, se debe crear la aplicación, mediante el siguiente comando

```
heroku create nombre
```

Donde *nombre*, especifica el nombre de la aplicación. Luego se debe añadir la base de datos Postgres, para lo cual se usa el comando:

```
heroku addons:create heroku-postgresql:hobby-dev
```

Luego se debe aplicar el migrate y el migrations de nuevo, de la siguiente forma:

```
heroku run python manage.py migrate  
heroku run python manage.py makemigration
```

Solo falta guardar los cambios realizados para hacer el push al repositorio remoto de heroku, primero ejecutar los siguientes comandos:

```
git add .  
git commit -m "deploy"
```

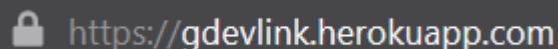
Para enviar la aplicación a Heroku, luego se ejecuta el comando:


```
git push heroku master
```

Este proceso puede tardar incluso minutos la primera vez que se hace, o si se hacen cambios en el archivo "requirements.txt", ya que Heroku primero debe instalar todas las dependencias de la aplicación, y además recibir todos los archivos de está. Cuando halla terminado, solo falta abrir la aplicación:

```
heroku open.
```

En la consola, aparecerá el enlace que se debe usar para ingresar a la aplicación en la web, que debe ser similar a la siguiente imagen:

A dark rectangular box containing a lock icon and the URL <https://gdevlink.herokuapp.com>.

 <https://gdevlink.herokuapp.com>

## PARTE 2: OPERACIÓN

### 1. Super usuario:

Para la creación de un super usuario, se debe estar en la consola, en el directorio de la aplicación, y con el entorno virtual activado. Luego se debe estar autenticado en heroku, usando el comando ya visto, *heroku login*. Posteriormente, se ejecutan los comandos:

```
heroku run python manage.py createsuperuser  
heroku run python manage.py migrate
```

El comando de *createsuperuser*, le pedirá los datos del usuario.

### 2. Ventana de administración:

La página de administración permite administrar toda la información de la aplicación, Puede ser leída la información que no está encriptada. Para acceder a ella, se debe usar el siguiente link: <https://<nombre de la app>.herokuapp.com/admin/>

Una vez allí, se debe ingresar con los datos del super usuario creado anteriormente. Una vez logeado, aparecerá esta página:

## Django administration

Site administration

ACCOUNTS		
Email addresses	+ Add	Change
AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
COMUNICACION		
Conversacions	+ Add	Change
Mensajes	+ Add	Change
POSICIONVACANTE		
Posicion vacantes	+ Add	Change
PREGUNTAS		
Preguntas	+ Add	Change
Respuestas	+ Add	Change
PROYECTOS		
Actualizacions	+ Add	Change
Participacions	+ Add	Change

Esta, está organizada por cada modelo, y en cada uno de estos, están las clases del modelo. Al hacer click sobre una clase, aparecerá una pestaña, en donde se muestran todas las instancias de dicha clase, almacenadas en la base de datos.

Los objetos pueden ser eliminados, o ver sus atributos modificados. En el caso de los usuarios, estos pueden, además, ser dados el rango de super usuario.

### 3. Cambios posteriores:

La aplicación no es muy amigable a los cambios, debido a la forma que se implemento el almacenamiento de las imágenes, si se introducen nuevos cambios, las imágenes subidas por los usuarios, serán eliminadas.

Sin embargo, realizar nuevos cambios al producto desplegado, es muy fácil con heroku. Una vez se tengan todos los cambios deseados en el repositorio local de uno, se deben ejecutar los siguientes comandos:

```
git add .
git commit -m "deployment ready"
git push heroku master
```



Es importante tener en cuenta, que, si se realizan cambios a la base de datos, es necesario ejecutar los siguientes comandos:

```
heroku python manage.py migrate
```

```
heroku python manage.py makemigrations
```