

#

# DESARROLLO WEB FULLSTACK

con Python y JavaScript

polotic  
misiones



## Clase 11 Anexo 2

polotic  
misiones





# Desplegando Django a Producción



Ya hemos visto como desarrollar una aplicación desde cero en nuestro entorno de desarrollo local, con un servidor en nuestro ordenador.

Si vamos a utilizar nuestra web, ya sea en internet o en una intranet, necesitamos realizar algunas modificaciones en el código de las configuraciones para agregar seguridad a nuestra aplicación y para mejorar su desempeño en la infraestructura que estamos por desplegar.

# ¿Qué es un entorno de Producción?

El entorno de producción es el servidor (o conjunto de servidores) que alojarán el aplicativo Django para que sea utilizado para el fin que se la ha designado. Éste uso suele ser de accesos internos (usuarios dentro de la red, una intranet) o también accesos externos (desde la web, internet).

El entorno tiene en cuenta lo siguiente:

- La arquitectura de los equipos donde se despliega la aplicación (Si es un servidor de alojamiento compartido, si es un servidor en la nube, una VPS, una VPS en la nube, etc.)
- Que sistema operativo correrá (Linux, Windows, etc.)
- Que servidor web se encargará de servir las páginas de la aplicación (Apache, Nginx, etc)
- Cual es el motor de bases de datos
- Cuales son las limitaciones de CPU, Memoria RAM, espacio en disco, etc.



# Proveedor de Hosting



El proveedor de hosting es el que ofrece el servicio de alojamiento de tu aplicación en la web. Cada proveedor ofrece diferentes arquitecturas y tecnologías a diferente precio. Puedes encontrar un listado de los servicios de hosting para Django (clasificados por categorías como Hobistas, Expertos, Especialistas, etc.) en <https://djangofriendly.com/>



# Proveedor de Hosting



Al elegir un hosting debemos tener en cuenta no solo el precio sino también lo siguiente:

- Cuales son los recursos que se van a necesitar para el despliegue (Consumo de CPU, consumo de memoria RAM, cantidad de espacio en disco). Con una holgura del 20%.
- Donde están ubicados los datacenters donde están alojados los servidores. Cuanto mas lejos puede que tengamos problemas de latencia.
- Uptime y Downtime. Cuanto tiempo esta disponible el servidor, se espera una disponibilidad (Uptime) lo mas cercana al 100%.
- Que herramientas de gestión ofrece. SFTP para subir los archivos permite mayor seguridad. Un acceso SSH permite gestión remota segura, etc.
- Que herramientas de monitoreo y backup ofrecen. Backups automáticos periódicos son esenciales.
- El soporte tiene que ser 24/7 y por canales rapidos de comunicación. Chat online o teléfono.
- Cuales son los beneficios adicionales que puedo obtener por el mismo precio o un precio accesible. Estos beneficios pueden ser un nombre de dominio, SSL, etc.
- Cuales son las limitaciones y que sucede cuando se llegan a éstas.



# Preparaciones



Es bastante común disponer de un archivo **settings.py** separado en producción, e importar los ajustes sensibles desde un archivo aparte o desde una variable de entorno. Este archivo debería, por tanto, estar protegido, aún cuando el resto del código fuente esté disponible en un repositorio público.

**DEBUG:** Es necesario mantener éste parámetro en False, de ésta manera no se visualizarán los Traceback para evitar divulgar información confidencial o sensible.

**SECRET\_KEY:** Es el valor que se utiliza para los token en CSRF. No debería ser expuesta directamente en el archivo de configuración sino con un archivo externo (ver Python-decouple)



# Preparaciones



Internet es un entorno hostil. Antes de implementar tu proyecto Django, debes tomarte un tiempo para revisar su configuración, teniendo en cuenta la seguridad, el rendimiento y las operaciones.

La siguiente lista de verificación incluye configuraciones que:

- debe configurarse correctamente para que Django proporcione el nivel de seguridad esperado;
- se espera que sean diferentes en cada entorno;
- habilitar características de seguridad opcionales;
- habilitar optimizaciones de rendimiento;
- proporcionar informes de errores.

Muchos de estos entornos son sensibles y deben tratarse como confidenciales. Si va a publicar el código fuente de su proyecto, una práctica común es publicar la configuración adecuada para el desarrollo y utilizar un módulo de configuración privada para la producción.

Puedes usar el comando `manage.py check --deploy` para el chequeo automático (sobre el settings.py de producción)





# Requerimientos



Al desarrollar un proyecto seguramente vas incorporando módulos que no están por defecto en Python o Django, como el motor de bases de datos. Debes registrar en un archivo de texto todos los módulos que incorporas para luego hacerlo disponible a otros con los que compartes el código.

En tu entorno virtual visualizas que paquetes tienes instalados con el comando:

**pip list**

o

**pip freeze**

Luego los agregas a un archivo de texto, como por ejemplo requirements.txt concatenando los comandos de la siguiente manera:

**pip freeze > requirements.txt**

# Requerimientos

Tendrás un archivo de texto txt con todos los módulos requeridos para el proyecto:

```
≡ requirements.txt X  
PROJECTCLASE12 > ≡ requirements.txt  
1 asgiref==3.3.4  
2 Django==3.2.3  
3 PyMySQL==1.0.2  
4 PyQt5==5.15.4  
5 PyQt5-Qt5==5.15.2  
6 PyQt5-sip==12.8.1  
7 python-decouple==3.4  
8 pytz==2021.1  
9 sqlparse==0.4.1  
10
```

Si estás en un entorno nuevo y virgen ejecutas lo siguiente para poder instalar estos módulos:

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



# Desglosar archivos de configuración



Puedes tener varios archivos `settings.py` para la configuración general, la configuración en local y la configuración en producción.

Para hacerlo primeramente creas dentro del directorio de tu proyecto un archivo `settings_base.py` que contendrá la información general, un archivo `settings_local.py` para la información de configuración localhost de desarrollo y un archivo `settings_production.py` con la información necesaria para desplegar a producción.

- `settings_base.py` contendrá toda la info original de `settings.py` pero sin la info de base de datos
- `set_local.py` contendrá, por ejemplo, la conexión a SQLite por defecto, Debug en True, ALLOWED\_HOST vacío. Podemos importar todo lo que tenemos en settings base con `from .base import *`
- `set_produ.py` contendrá, por ejemplo, la conexión a una base MySQL, Debug en False, ALLOWED\_HOST en la URL de nuestro host de aplicaciones como por ej. Heroku. Podemos importar todo lo que tenemos en settings base con `from .base import *`

# Desglosar archivos de configuración



Luego, accedemos al archivo `manage.py` y dentro de la función `main()`: agregamos ambas llamadas a los settings:

```
os.environ.setdefault('DJANGO_SETTINGS_MODULE', ' MIPROYECTO.settings.set_local')
```

```
os.environ.setdefault('DJANGO_SETTINGS_MODULE', ' MIPROYECTO.settings.set_produ')
```

Demás está decir que debes comentar aquel que no vayas a usar.



# Recolectar Contenido Estático

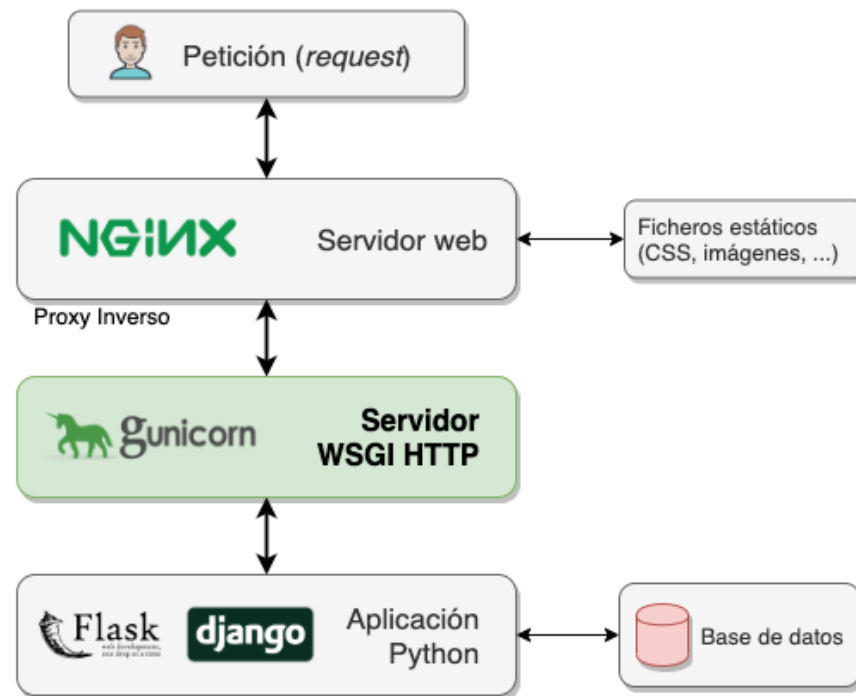


Podemos recolectar todo el contenido estático en la ubicación del directorio que configuramos escribiendo lo siguiente:

```
python manage.py collectstatic
```

Deberá confirmar la operación. Luego, los archivos estáticos se ubicarán en un directorio llamado static, dentro del directorio de su proyecto.

# Estructura de un entorno en Produccion



[Mas info aquí](#)



# HEROKU



# HEROKU



Uno de los servicios plataformas de aplicaciones mas populares y conocidos. No solo por su versatilidad sino también porque permite desplegar aplicaciones gratuitas y probarlas en un entorno en la nube. Se basa en una estructura de gestión de contenedores.

Heroku limita ciertas cuestiones en su nivel gratuito como:

- ❖ Almacenamiento efímero: Los archivos subidos por el usuario no pueden almacenarse de forma segura.
- ❖ Si la aplicación esta inactiva por un período de tiempo determinado Heroku la duerme automáticamente.
- ❖ Heroku limita la cantidad de tiempo que la aplicación puede correr al mes.

Sin embargo nos sirve para realizar pruebas y evaluar los servicios.

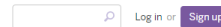
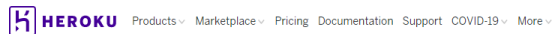


# HEROKU



Pasos para desplegar nuestro proyecto Django:

1. Registrarse en heroku.com (Sign up for free)



## SPOTLIGHT

### Data on Heroku

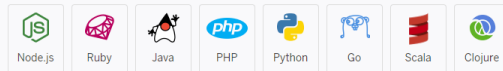
Build data-driven apps with fully managed data services.

[SIGN UP FOR FREE](#)

[Explore Managed Data Services](#)



## OFFICIALLY SUPPORTED LANGUAGES



In addition to our officially supported languages, you can use any language that runs on Linux with Heroku via a third-party buildpack. [View all buildpacks](#) →

# HEROKU



Pasos para desplegar nuestro proyecto Django:

1. Registrarse en heroku.com (Sign up for free)

A screenshot of the Heroku sign-up page. The page has a dark purple background. At the top, there's a navigation bar with the Heroku logo, a link to 'Already have an account?', and a 'Log in' button. The main heading is 'Sign up for free and experience Heroku today'. Below this, there are three sections: 'Free account' (with a plus icon), 'Your app platform' (with a plus icon), and 'Deploy now' (with a plus icon). To the right of these sections is a form with fields for 'First name', 'Last name', 'Email address', 'Company name', 'Role', 'Country', and 'Primary development language'. There is also a checkbox for 'I'm not a robot' and a 'CAPTCHA' image. At the bottom of the form is a blue button labeled 'CREATE FREE ACCOUNT'. Below the button, there is a small disclaimer: 'Signing up signifies that you have read and agree to the Terms of Service and our Privacy Policy. Cookie Preferences'.

# HEROKU



Pasos para desplegar nuestro proyecto Django:

2. Darle un nombre a la aplicación (un nombre único dentro de todo heroku), además elegir la región del datacenter

Create New App

App name

nombre-de-tu-aplicacion



curso-polo-aerolinea is available

Choose a region



United States



Add to pipeline...

Create app

# HEROKU



Pasos para desplegar nuestro proyecto Django:

3. Para hacer el deploy de tu aplicación (tienes que tener instalado git) instalas el CLI de Heroku (lo descargas desde la oagina), lo agregas al path de tu sistema operativo y ya tienes disponible en la línea de comandos

#### Deploy using Heroku Git

Use git in the command line or a GUI tool to deploy this app.

#### Install the Heroku CLI

Download and install the [Heroku CLI](#).

If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.

```
$ heroku login
```

#### Create a new Git repository

Initialize a git repository in a new or existing directory

```
$ cd my-project/  
$ git init  
$ heroku git:remote -a curso-polo-aerolinea
```

#### Deploy your application

Commit your code to the repository and deploy it to Heroku using Git.

```
$ git add .  
$ git commit -am "make it better"  
$ git push heroku master
```



You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions [here](#).

#### Existing Git repository

For existing repositories, simply add the [heroku](#) remote

```
$ heroku git:remote -a curso-polo-aerolinea
```

# HEROKU

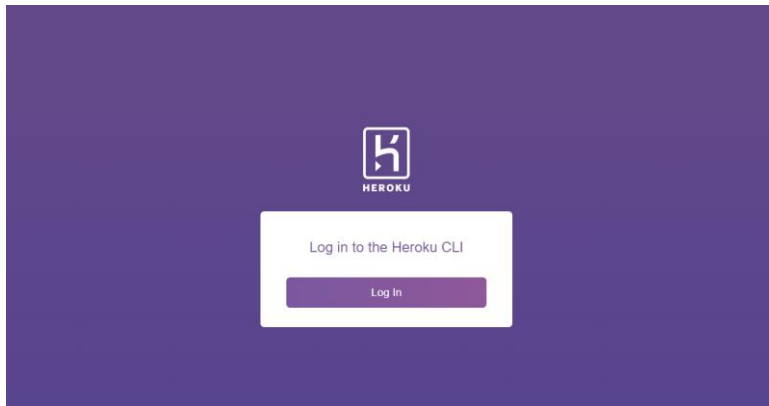


Pasos para desplegar nuestro proyecto Django:

## 4. Debes autenticarte en heroku con **heroku login**

```
Windows PowerShell
PS C:\Users\rfede\Documents\Cursos PoloTIC\CursoPythonconJavaScript2021\Clase 12\PROJECTCLASE12> heroku login
```

Te permitirá acceder al navegador web y autenticar la sesión:



```
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/ee86117d-f28e-4ff2-83a1-4xNzAuODMuMTI0LjExNm4GAe4Nm-t5AWIAAVGA.XWKnRtsJ6lpW_
Logging in... done
Logged in as [email address]
PS C:\Users\rfede\Documents\Cursos PoloTIC\CursoPythonconJavaScript2021\Clase 12\PROJECTCLASE12>
```

# HEROKU



Pasos para desplegar nuestro proyecto Django:

5. Recuerda que dentro de tu aplicación, en la raíz del directorio debes tener un archivo **requirements.txt** con todos los modulos requeridos para que la aplicación funcione. Incluido el modulo gunicorn. Gunicorn permite proveer de un servidor Python WSGI HTTP a UNIX (en lugar de nuestro servidor local)

A screenshot of a code editor window with a dark theme. The editor has three tabs at the top: 'Procfile', 'settings.py', and 'requirements.txt'. The 'requirements.txt' tab is active. The file content is as follows:

```
1 asgiref==3.3.4
2 Django==3.2.3
3 PyMySQL==1.0.2
4 PyQt5==5.15.4
5 PyQt5-Qt5==5.15.2
6 PyQt5-sip==12.8.1
7 python-decouple==3.4
8 pytz==2021.1
9 sqlparse==0.4.1
10 gunicorn==19.0.0
11 |
```

# HEROKU



Pasos para desplegar nuestro proyecto Django:

6. En la raíz de tu proyecto crea un archivo Procfile (asi sin extensión) que definira donde se encuentra el [WSGI](#) de tu aplicación Django:

```
Procfile  X
Procfile
1 web: gunicorn PROJECTCLASE12.wsgi --log-file -
```



# HEROKU



Pasos para desplegar nuestro proyecto Django:

7. En settings.py define ALLOWED\_HOSTS igual a la URL de tu aplicación Heroku

```
ALLOWED_HOSTS = ['tu-aplicacion.herokuapp.com']
```



# HEROKU



Pasos para desplegar nuestro proyecto Django:

## 8. Inicializar GIT

```
PS C:\Users\rfede\Documents\Cursos PolotIC\CursoPythonconJavaScript2021\Clase 12\PROJECTCLASE12> git init
Initialized empty Git repository in C:/Users/rfede/Documents/Cursos PolotIC\CursoPythonconJavaScript2021\Clase 12\PROJECTCLASE12/.git/
PS C:\Users\rfede\Documents\Cursos PolotIC\CursoPythonconJavaScript2021\Clase 12\PROJECTCLASE12> heroku git:remote -a curso-polo
set git remote heroku to https://git.heroku.com/curso-polo
```

## Agregar los archivos y hacer el commit

```
PS C:\Users\rfede\Documents\Cursos PolotIC\CursoPythonconJavaScript2021\Clase 12\PROJECTCLASE12> git add .
PS C:\Users\rfede\Documents\Cursos PolotIC\CursoPythonconJavaScript2021\Clase 12\PROJECTCLASE12> git commit -am "Inital Commit"
[master (root-commit) 76e8727] Inital Commit
 57 files changed, 767 insertions(+)
```

## Hacer el push al repositorio

```
PS C:\Users\rfede\Documents\Cursos PolotIC\CursoPythonconJavaScript2021\Clase 12\PROJECTCLASE12> git push heroku master
Enumerating objects: 68, done.
Counting objects: 100% (68/68), done.
Delta compression using up to 12 threads
Compressing objects: 100% (64/64), done.
Writing objects: 100% (68/68), 29.80 KiB | 4.26 MiB/s, done.
Total 68 (delta 13), reused 0 (delta 0), pack-reused 0
```

# HEROKU



```
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-20 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Python app detected
remote: -----> No Python version was specified. Using the buildpack default: python-3.9.5
remote: -----> To use a different version, see: https://devcenter.heroku.com/articles/python-runtimes
remote: -----> Installing python-3.9.5
remote: -----> Installing pip 20.2.4, setuptools 47.1.1 and wheel 0.36.2
remote: -----> Installing SQLite3
remote: -----> Installing requirements with pip
remote: Collecting asgiref==3.3.4
remote:   Downloading asgiref-3.3.4-py3-none-any.whl (22 kB)
remote: Collecting Django==3.2.3
remote:   Downloading Django-3.2.3-py3-none-any.whl (7.9 MB)
remote: Collecting PyMySQL==1.0.2
remote:   Downloading PyMySQL-1.0.2-py3-none-any.whl (43 kB)
remote: Collecting PyQt5==5.15.4
remote:   Downloading PyQt5-5.15.4-cp36.cp37.cp38-cp39-abi3-manylinux2014_x86_64.whl (8.3 MB)
remote: Collecting PyQt5-Qt5==5.15.2
remote:   Downloading PyQt5-Qt5-5.15.2-py3-none-manylinux2014_x86_64.whl (59.9 MB)
remote: Collecting PyQt5-sip==12.8.1
remote:   Downloading PyQt5-sip-12.8.1-cp39-cp39-manylinux1_x86_64.whl (286 kB)
remote: Collecting python-decouple==3.4
remote:   Downloading python-decouple-3.4-py3-none-any.whl (9.5 kB)
remote: Collecting pytz==2021.1
remote:   Downloading pytz-2021.1-py2.py3-none-any.whl (510 kB)
remote: Collecting sqlparse==0.4.1
remote:   Downloading sqlparse-0.4.1-py3-none-any.whl (42 kB)
remote: Installing collected packages: asgiref, pytz, sqlparse, Django, PyMySQL, PyQt5-Qt5, PyQt5-sip, PyQt5, python-decouple
remote: Successfully installed Django-3.2.3 PyMySQL-1.0.2 PyQt5-5.15.4 PyQt5-Qt5-5.15.2 PyQt5-sip-12.8.1 asgiref-3.3.4 python-decouple-3.4 pytz-2021.1 sqlparse-0.4.1
remote: -----> Skipping Django collectstatic since the env var DISABLE_COLLECTSTATIC is set.
remote: -----> Discovering process types
remote: Procfile declares types -> (none)
remote:
remote: -----> Compressing...
remote: Done: 122.3M
remote: -----> Launching...
remote: Released v6
```

# HEROKU



Pasos para desplegar nuestro proyecto Django:

9. Inicializar el dyno web con **heroku ps:scale web=1**

```
PS C:\Users\rfede\Documents\Cursos PolotIC\CursoPythonconJavaScript2021\Clase 12\PROJECTCLASE12> heroku ps:scale web=1
Scaling dynos... done, now running web at 1:Free
```

# HEROKU



Pasos para desplegar nuestro proyecto Django:

10. Ya puedes acceder a tu aplicación mediante la URL que te da heroku:

← → ↻ 🏠 🔒 tu-aplicacion.herokuapp.com/%5Eaccounts/reset/done/\$login/?next=/AEROLINEA/

Inicie sesión para ver esta página.

Username:

Password:

[¿Olvidó su contraseña?](#)