

Proyecto DC19	Versión: 1.0
Manual de despliegue	Fecha: 03/11/2020

## **Manual de despliegue**

### **Proyecto DC19**

### **Versión 1.0**

Proyecto DC19	Versión: 1.0
Manual de despliegue	Fecha: 03/11/2020

## Tabla de contenidos

<b>Paso 1</b>	<b>3</b>
<b>Paso 2</b>	<b>3</b>
<b>Paso 3</b>	<b>3</b>
<b>Paso 4</b>	<b>3</b>
<b>Paso 5</b>	<b>4</b>
<b>Paso 6</b>	<b>4</b>
<b>Paso 7</b>	<b>4</b>
<b>Paso 8</b>	<b>4</b>
<b>Paso 9</b>	<b>4</b>
<b>Paso 10</b>	<b>5</b>
<b>Paso 11</b>	<b>5</b>
<b>Paso 12</b>	<b>5</b>
<b>Paso alternativo al 6</b>	<b>6</b>
<b>Paso extra</b>	<b>6</b>
Comandos Heroku útiles	6

Proyecto DC19	Versión: 1.0
Manual de despliegue	Fecha: 03/11/2020

## Paso 1

Instalar waitress

```
pip install waitress
```

## Paso 2

Dentro del directorio de la aplicación principal (en este caso dc19.py), añadir las siguientes líneas de código luego de crear el objeto app:

```
if __name__ == "__main__":  
    app.debug = False  
    port = int(os.environ.get('PORT', 33507))  
    waitress.serve(app, port=port)
```

## Paso 3

Descargar e instalar Heroku CLI. ([The Heroku CLI | Heroku Dev Center](#))

## Paso 4

Abrir una terminal/línea de comando y ejecutar el siguiente comando:

```
heroku login
```

Proyecto DC19	Versión: 1.0
Manual de despliegue	Fecha: 03/11/2020

## Paso 5

Cambiar al directorio del repositorio del código fuente:

```
cd mi-directorio
```

## Paso 6

Inicializar git en el directorio (si no cuenta con una cuenta git, [Git - Downloads \(git-scm.com\)](https://git-scm.com/downloads))

```
git init
```

## Paso 7

Crear una app:

```
heroku create your_app_name
```

## Paso 8

Generar un archivo requirements.txt

```
pip freeze > requirements.txt
```

## Paso 9

Crear un Procfile (sin extensión) en el directorio raíz:

Proyecto DC19	Versión: 1.0
Manual de despliegue	Fecha: 03/11/2020

```
web: gunicorn your_app_name:app
```

## Paso 10

Instalar gunicorn server

```
pip install gunicorn
```

## Paso 11

Crear runtime.txt y añadir la siguiente línea:

```
python-3.6.12
```

## Paso 12

Finalmente añadimos vía git

```
git add .  
git commit -m "Mi mensaje de commit"  
git push heroku master
```

**Nota:** En lugar de “git push heroku master” si se desea publicar desde un directorio, se puede ejecutar:

```
git subtree push --prefix Desarrollo/DC19/Fuentes/DC19-Flask heroku master
```

Proyecto DC19	Versión: 1.0
Manual de despliegue	Fecha: 03/11/2020

## Paso alternativo al 6

Podemos clonar un repositorio, por ejemplo

```
git clone https://github.com/warevil/DC19-Flask
cd DC19-Flask
git checkout

heroku apps:create mi-site

git remote -v
```

## Paso extra

```
git remote rename dc19-gcs heroku
```

## Comandos Heroku útiles

El siguiente comando nos permite establecer la configuración de un output:

```
heroku config:set LOG_TO_STDOUT=1
```

El siguiente comando nos permite establecer la aplicación principal:

```
heroku config:set FLASK_APP=dc19.py
```

El siguiente comando nos permite revisar los logs (muy útil para debugging):

```
heroku logs
```

Proyecto DC19	Versión: 1.0
Manual de despliegue	Fecha: 03/11/2020