



Proyecto DAW

Manual de despliegue

Fco.Javier Carande Cabeza
DAW 2020/21

Índice

- Requisitos previos
- Despliegue del sistema
 - Despliegue del back
 - Despliegue del front

Requisitos previos

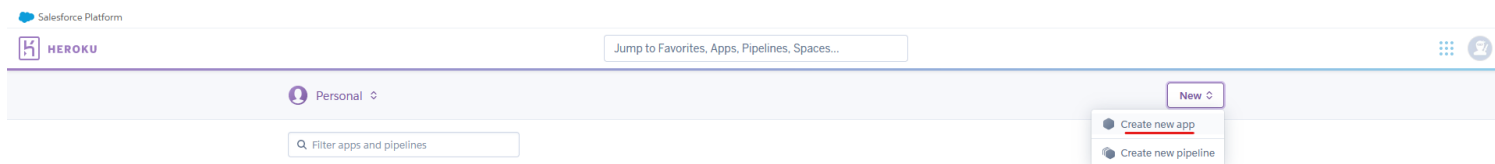
Para desplegar el proyecto en la plataforma Heroku deberemos primero registrarnos en dicha plataforma y vincular una tarjeta bancaria. Usaremos la versión gratuita de Heroku, sin embargo este nos pide una tarjeta vinculada para poder añadir el Add-on de nuestra base de datos.

También, de forma opcional y por comodidad, necesitaremos una cuenta en GitHub. Una vez tengamos nuestra cuenta GitHub deberemos subir en dos repositorios distintos nuestro código de producción back y front.

Despliegue del sistema

Despliegue del back


Para desplegar nuestro back tendremos que crear en Heroku una nueva app.



Habiendo hecho click en 'Create new app' aparecerá una pantalla que nos permitirá dar nombre a nuestra aplicación y elegir la región sobre la que se desplegará.



Create New App

App name



nombre-de-app is available

Choose a region

 Europe 

Add to pipeline...

Create app

Una vez creado, nos aparecerá un menú que nos permitirá realizar diversas acciones. Para empezar debemos incluir nuestra BD, por lo que iremos a la pestaña 'Resources' y buscaremos el Add-on correspondiente para nuestra base de datos. Para este manual usaremos JawsDB Maria que nos da un plan gratuito con 10 conexiones.

Personal > nombre-de-app

Overview Resources Deploy Metrics Activity Access Settings

Dynos

This app has no process types yet
Add a Procfile to your app in order to define its process types. [Learn more](#)

Add-ons [Find more add-ons](#)

The add-on `jawsdb-maria` has been installed. Check out the documentation in its [Dev Center](#) article to get started.

Quickly add add-ons from Elements

JawsDB Maria Attached as JAWSDB_MARIA Kitefin Shared Free

Estimated Monthly Cost \$0.00

Ahora accederemos al dashboard de nuestra BD que nos proporcionará los datos de conexión a la base de datos. Con estos datos procederemos a cambiar nuestro `application.properties` que lleva los datos de conexión a la base de datos.

Settings for nombre-de-app

Connection Info


Connection String

```
mysql://edlqng36sobwihjq:mfxnood6ax2wbzoe@j5zntocs2dn6c3fj.chr7pe7iynqr.eu-west-1.rds.amazonaws.com:3306/arb5t13j982bx1k
```

You can use your connection information to connect manually through a client such as [HeidiSQL](#) to administer your database.

Property	Value	Action
Host	j5zntocs2dn6c3fj.chr7pe7iynqr.eu-west-1.rds.amazonaws.com	
Username	edlqng36sobwihjq	
Password	mfxnood6ax2wbzoe	Reset
Port	3306	
Database	arb5t13j982bx1k	

```
1 spring.datasource.url=jdbc:mysql://yyglm7fnnbel doe3:h6rvoy3hj8p9q0nv@ulsq0qqx999wqz84.chr7pe7iynqr.eu-west-1.rds.amazonaws.com:3306/xbbf3uvnlkominn1
2 spring.datasource.username=yyglm7fnnbel doe3
3 spring.datasource.password=h6rvoy3hj8p9q0nv
4 hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
5 spring.jpa.hibernate.ddl-auto=create
```



Ahora procederemos a conectar nuestro repositorio GitHub del back con Heroku, esto nos permitirá que en siguientes updates de nuestro app con una simple subida del repositorio Heroku despliegue automáticamente nuestro back.

En la sección Deploy encontraremos los métodos de despliegue, seleccionamos GitHub. Una vez asociemos nuestra cuenta GitHub tendremos que seleccionar qué repositorio queremos desplegar. Al seleccionar el repositorio deseado obtendremos dos opciones, despliegue automático o despliegue manual, en ambos casos seleccionamos sobre qué rama queremos que Heroku actúe. Si hemos seleccionado el despliegue automático, por cada push a esa rama en GitHub Heroku buildeará nuestro repositorio y lo desplegará.

Cuando heroku haya construido nuestro app sin problema procederá a desplegarlo, proporcionandonos una url.

Despliegue del front

Este despliegue no dista mucho del back, sin embargo como se trata de Angular este necesitará una serie de directivas para poder ser desplegado. Cabe recordar que al estar el back subido a Heroku, debemos cambiar la dirección del back en nuestro **global.ts** para que los servicios de Angular tengan acceso al back y poder consumir sus recursos.

Primero necesitaremos instalar una dependencia de node: express.

npm install --save express path

Una vez instalada esta dependencia tendremos que definir cómo queremos que Heroku despliegue nuestro proyecto, para ello incluiremos estas líneas en el *package.json* del proyecto Angular.

```
"scripts": {  
  ...  
  "postinstall": "ng build --prod",  
  "preinstall": "npm install -g @angular/cli @angular/compiler-cli typescript"  
},  
"engines": {  
  "node": "8.11.3",  
  "npm": "6.1.0"  
},
```

El siguiente paso será crear un archivo *server.js*, este nos permite dar una serie de directrices de cómo se iniciara nuestro proyecto.

```
const path = require('path');  
const express = require('express');  
const app = express();  
  
// Serve static files  
app.use(express.static(__dirname + '/dist/MI_APP'));  
  
// Send all requests to index.html  
app.get('/*', function(req, res) {
```

```
res.sendFile(path.join(__dirname + '/dist/MI_APP/index.html'));  
});
```

```
// default Heroku port
```

```
app.listen(process.env.PORT || 5000);
```

En el apartado de direcciones, donde pone *MI_APP* nosotros colocaremos el nombre que tenga nuestra app.

Lo siguiente será crear un archivo Procfile, este archivo le indica a Heroku como iniciar la aplicación.

web: node server.js

Una vez hecho todos estos pasos tendremos que buildear nuestra aplicación para crear la estancia estática de nuestra app.

ng build --prod

Ya podemos desplegar nuestra aplicación, ya sea de forma local haciendo uso del comando **node server.js**. Para levantarlo en Heroku esta vez usaremos el CLI de Heroku.

Una vez hecho login en el CLI con nuestra cuenta Heroku realizaremos los siguientes comandos. De esta manera trabajaremos al igual que con GitHub, como si fuese un repositorio.

heroku create

```
git add .
```

```
git commit -m "initial heroku deploy"
```

```
git push heroku master
```

Al hacer push veremos como Heroku realiza todos los pasos necesarios para construir y desplegar la app. Una vez hecho podemos ir a Heroku y veremos como nuestro front está en funcionamiento, accedemos al link que nos proporciona y podremos empezar a usar nuestra app.