Carlos Puyana Paraíso, 14/06/2022

Manual de Instalación y Despliegue

Proyecto Desarrollo de Aplicaciones Web

Índice

1. Requerimientos previos de Instalación

Para llevar a cabo la instalación debemos de contar con una serie de requisitos previos básicos de los que poder comenzar sin problema alguno. Se tendrá en cuenta que el usuario que realizara la instalación tiene unos conocimientos básicos de informática.

Los requisitos de hardware son:

- Al menos 10Gb de espacio de almacenamiento libre.
- Al menos 2GB de memoria RAM
- Procesador 300Mhz x86 como mínimo.
- Conexión a internet

Los requisitos para el software son:

- Sistema operativo instalado, en el documento se va a realizar la instalación sobre Windows 10, aunque también es válido para Windows 11 y Ubuntu.
- Navegador Web, preferiblemente navegador basado en Chromium.

2. Instalación del sistema

Basándonos en un sistema operativo Windows 10.

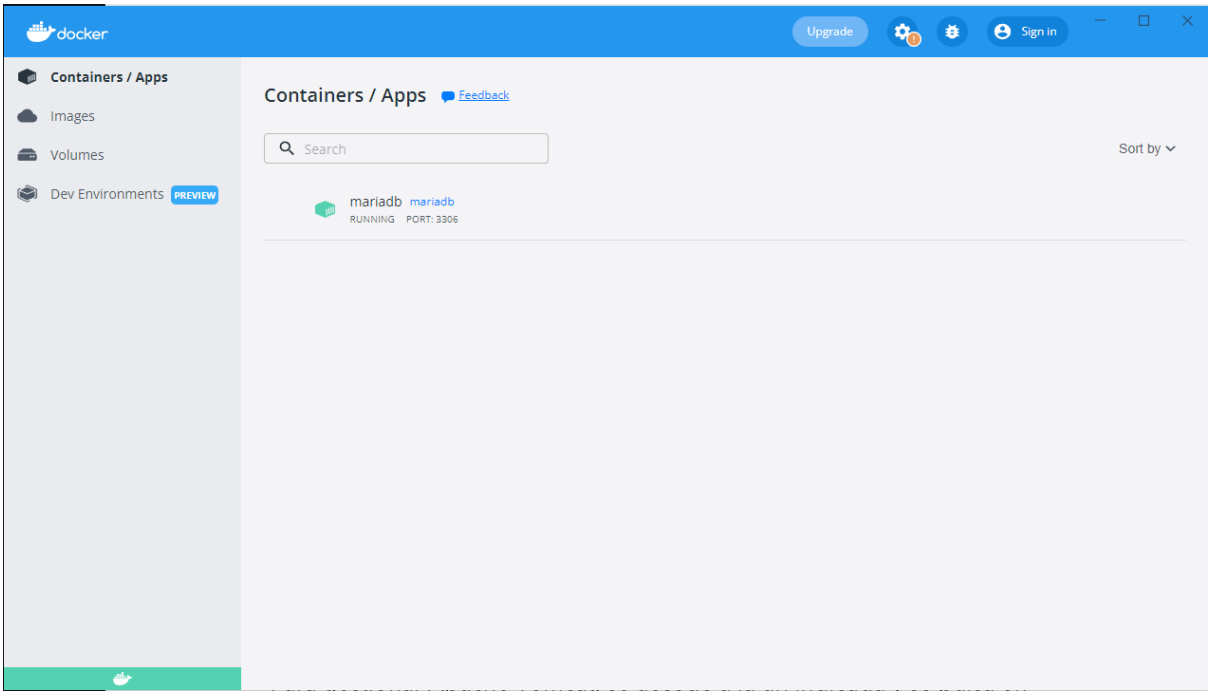
- Instalación de persistencia: Al disponer del framework de Spring, este ya tiene implementado un sistema de persistencia mediante anotaciones en los modelos para definir las entidades, el cual nos facilita todo.

- Instalación de contenedor de aplicaciones: Deberemos tener un contenedor Docker con la imagen de MariaDB.
  Deberemos de tener Docker instalado en nuestro equipo. Siguendo el siguiente tutorial se podrá instalar en Windows.
  Una vez instalado deberemos descargar la imagen de MariaDB, poniendo el siguiente comando en nuestro cmd/powershell `docker pull mariadb`

- Instalación servidor local: Con Spring Boot trae embebido un contenedor Tomcat el cual escucha todas las peticiones de manera local y la despliega automáticamente. Con el cual podremos acceder de forma predeterminada desde un navegador con localhost seguido del puerto, 8080 de manera predeterminada, si se desea cambiar el puerto se deberá de poner la anotación server.port en el archivo application.properties

3. Despliegue del sistema en local

Lo primero que tendremos que tener desplegado sería nuestro contenedor Docker donde tengamos MariaDB.



Una vez ejecutada, podremos seguir con el despliegue del Back-End. Abrimos nuestro Spring Tools Suite Eclipse IDE y le daremos a Open Project from File System… y elegiremos la carpeta del Backend.

Ya teniendo nuestro proyecto Back-End abierto en nuestro Eclipse, deberemos actualizar las dependencias Maven para que así se instalen, le damos click derecho sobre el proyecto – Maven — Update Project

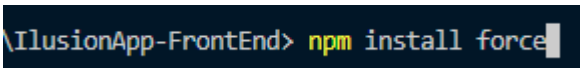
Cuando las dependencias se hayan instalados tendremos que dirigirnos a la carpeta resources y en el archivo application.properties verificar los datos de nuestra base de datos, tanto el usuario como contraseña. Además de como es la primera vez que ejecutamos nuestro proyecto tendremos que cambiar el ddl-auto de ‘update’ a ‘create’.

Una vez hayamos cambiado la configuración ya podemos ejecutar nuestro proyecto Back-End dandole click derecho sobre el — Run As — Spring Boot App . Y este empezará a ejecutarse y crear las tablas e inserciones necesarias en la base de datos.

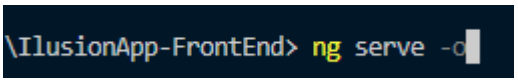
Habiendo hecho todo esto, ya tendremos nuestra parte del Back-End y Base de Datos desplegada.

Lo siguiente que tendremos que hacer, será desplegar nuestro Front-End. Para ello desde el Visual Studio Code abriremos la carpeta donde esta nuestro código del Front-End.

Cuando lo tengamos ya abierto en nuestro explorador tendremos que abrir un terminal y descargar todas las dependencias necesarias para el despliegue, con el comando:
npm install –force



Con este comando se descargarán todas las dependencias necesarias y podremos desplegarlo con éxito con el comando ‘ng serve -o’



Y por último se te abriría automáticamente una nueva ventana en su navegador con el Front-End desplegado.

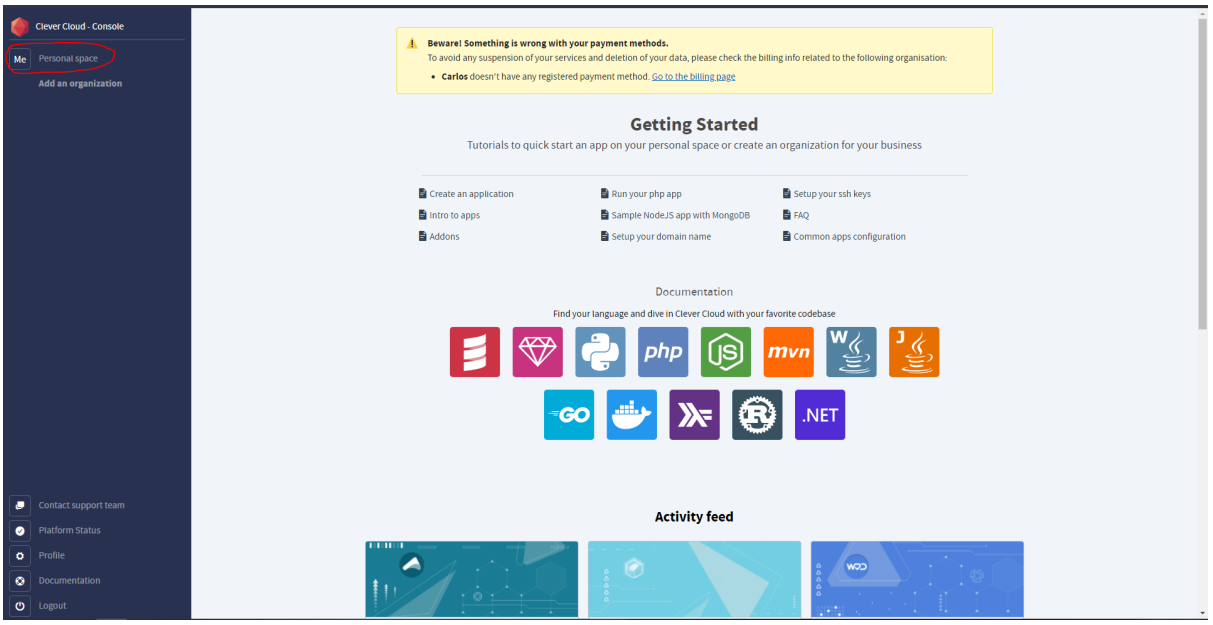
Y ya podrías usar la aplicación

4. Despliegue del sistema en la nube

Para desplegarlo en la nube, tendremos que subir cada parte de nuestro proyecto a una plataforma.

Empezamos con nuestra base de datos. Se puede usar Amazon Web Services o CleverCloud, aunque nosotros utilizaremos CleverCloud ya que es gratis y AWS necesitaras depositar 1$ para un año.

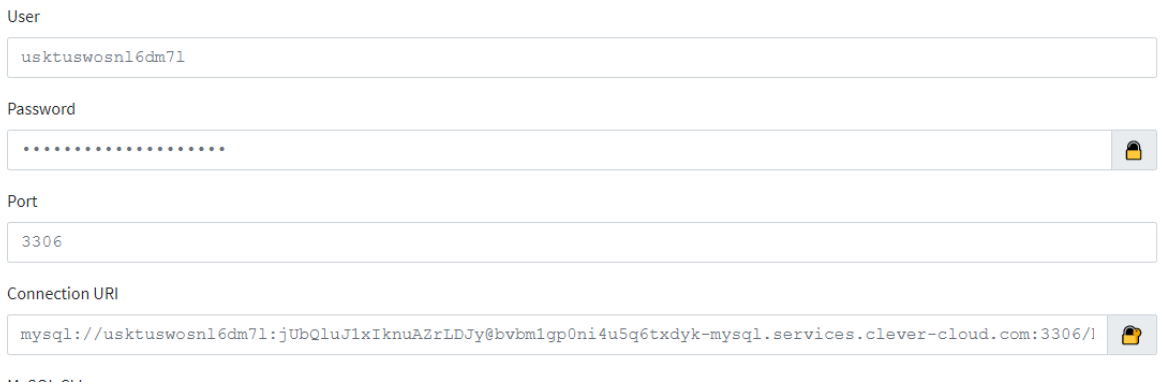
Una vez creada nuestra cuenta en Clever Cloud e iniciado sesión. Nos dirigiremos a `Personal space` en la barra de navegación lateral



Se te abrirá nuevos componentes y tendremos que clickar en `Create`, que se abrirá justo debajo del botón anterior y crearemos un Add-on. Y añadiremos el Add-on de MySQL, y pulsaremos en el primer plan ‘DEV’ que es gratuito. Le añadiremos un nombre a nuestro addon y elegiremos la ubicación de Francia y se creará y nos mandarán a nuestro Dashboard.

Una vez que tengamos nuestra base de datos alojada en la nube, tendremos que ir a nuestro Back-End y hacer algunos cambios antes de desplegarlo en la nube.

Nos dirigimos a nuestro application.properties y cambiaremos los datos que nos proporcione nuestro Dashboard de la base de datos. El uri, username, password



Al ser nuestra primera vez ejecutando el back-end en la nube, deberemos de cambiar de update a create (después del primer despliegue deberemos de cambiarlo nuevamente a update y realizar un despliegue).

Luego, deberemos de cambiar el driver de MariaDB a MySQL, ya que nuestra base de datos en la nube es de MySQL (no había posibilidad de MariaDB).

```
spring.datasource.driver-class-name= com.mysql.cj.jdbc.Driver
```

Por último tendremos que cambiar la dependencia de MariaDB a MySQL en nuestro pom.xml

```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <scope>runtime</scope>
</dependency>
```

<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<scope>runtime</scope>
</dependency>

Una vez hecha todas las configuraciones previas, ya podemos seguir con el despliegue del Back-End.

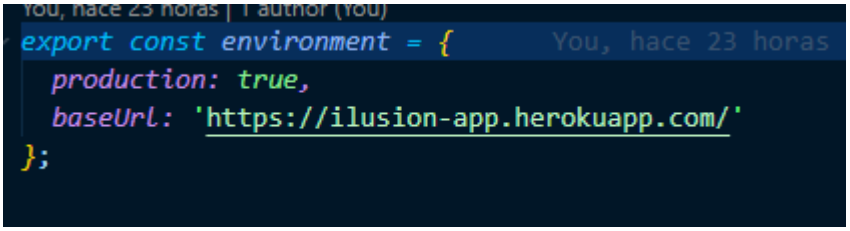
Nosotros lo desplegamos en la plataforma de Heroku. Tendremos que crearnos una cuenta e iniciar sesión en ella.
Estando en nuestro dashboard, pulsaremos en el botón `New` y crearemos una nueva app.

Una vez creada nuestra App nos iremos a la pestaña de Deploy, allí elegiremos el método de despliegue por GitHub, nos conectaremos con nuestra cuenta de GitHub y elegiremos el repositorio donde alojamos nuestro servidor back-end.

Cuando tengamos vinculado con nuestro repositorio GitHub, en la pestaña de Deploy al final del todo nos dirigimos a Manual deploy, y tendremos que hacer un deploy del branch donde está nuestra aplicación. Y ya tendríamos nuestro servidor Back-End desplegado en la nube y con ello se empezará a crear en la base de datos las tablas necesarias.

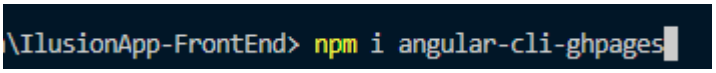
Ya por último nos faltaría desplegar en la nube nuestro servidor Front-End. Lo desplegamos de manera fácil a través de GitHub Pages, deberemos tener nuestro Front subido en un repositorio GitHub.

Antes de nada, tocaremos algo de la configuración. Nos iremos a la carpeta environments desde la carpeta src y en el archivo environment.prod.ts cambiaremos la url al del servidor heroku



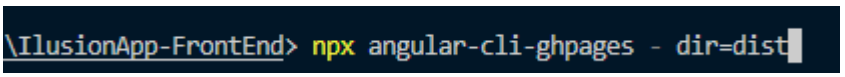
Tendremos que añadir esta url en varios sitios más donde hace llamada a la api directamente como es en la obtención de las imágenes de los usuarios.
Nos iremos al archivo html del perfil (perfil.component.html) y cambiaremos la línea 17 y 42, cambiando el localhost por la URL de heroku. Esto lo tendremos que hacer en el archivo perfil.component.ts también en la línea 73. Y ya por último en header.component.ts, la linea 62 y header.component.html en la línea 29.

Tendremos que instalar Angular CLI gh-pages, que es una dependencia que se encarga de todo el solo.
Iremos a la terminal de nuestro visual studio en el proyecto y usaremos el siguiente comando `npm i angular-cli-ghpages`



Después deberemos de hacer un build añadiendo el base href a index.html con el siguiente comando ` ng build –prod –base-href=”./” `

Y ya realizamos el deploy a GitHub Pages con el comando
` npx angular-cli-ghpages – dir=dist `



Y ya tendríamos desplegado nuestro servidor Front-End

Una vez desplegado el Front deberemos hacer cambios en el Back, tales como cambiar en application.properties de create a update para que así se actualicen los cambios en la base de datos. Y tendremos que añadir en el Cors la url de nuestro servidor Front.

En la carpeta de security, nos iremos al archivo de CorsConfig.java

Y cambiaremos la URL de localhost a la de nuestro github.

```
private String url = "https://carlospuyana.github.io";
```

Además deberemos de ir a nuestros controladores y añadir esta anotación en ellos con la url de nuestro github

```
@CrossOrigin(origins = "https://carlospuyana.github.io")
public class DuenoController {
```

Ya por último subimos nuestro nuevo código del Back-End al repositorio GitHub y nos dirigimos a Heroku y lo volvemos a desplegar desde el Branch donde lo subimos.

Y ya tendríamos nuestra aplicación desplegada en la nube.