

### 1. Prerrequisitos

A person is running away from the camera on a dirt path. The path is lined with trees and has some fallen leaves. The entire image is overlaid with a blue tint. The person is wearing a dark long-sleeved shirt and dark leggings. Their hands are on their head, and they are wearing white sneakers. The background shows a dense line of trees and foliage.

### 2. Despliegue del proyecto

A person is running away from the camera on a dirt path. The path is lined with trees and has some fallen leaves. The entire image is overlaid with a blue tint. The person is wearing a dark long-sleeved shirt and dark leggings. Their hands are on their head, and they are wearing white sneakers. The background shows a dense line of trees and foliage.

# GFit Analysis

## Guía de despliegue

|                                |          |
|--------------------------------|----------|
| <b>Prerrequisitos</b>          | <b>2</b> |
| <b>Despliegue del proyecto</b> | <b>2</b> |
| Back-end                       | 2        |
| Front-end                      | 4        |

Este documento se centrará en el despliegue del front-end y el back-end.

## Prerrequisitos

Pasos necesarios para el despliegue del proyecto.

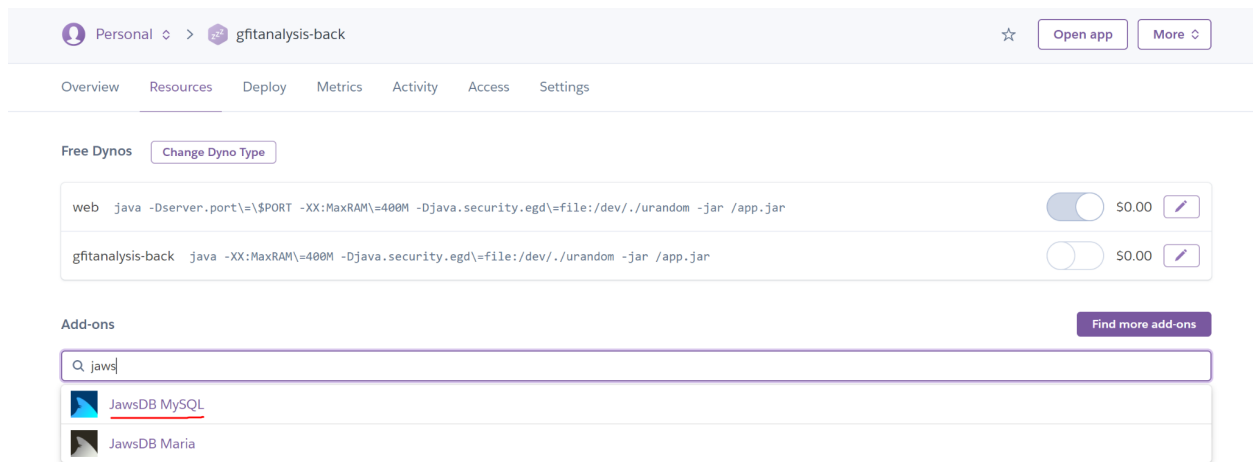
- Haber recreado todos los pasos del manual de instalación.
- Crear una cuenta en <https://www.heroku.com/>
- Acceder a la consola de Firebase <https://console.firebase.google.com/>

## Despliegue del proyecto

### Back-end

Para desplegar el back tendremos que crear un contenedor con la aplicación integrada y subirlo a heroku.

- Antes que nada iniciar sesión en heroku y crear una aplicación.
- Tras crear una aplicación debemos instalar el siguiente add-on:



“JawsDB MySQL”

- Tras instalarlo, al hacer click sobre el nombre del add-on, te dirigirá a una página indicando el string de conexión a la base de datos, el usuario y la contraseña. Estos datos se deben establecer en el application.properties localizado en “**Java Resources > src/main/resources > application.properties**” desde eclipse.
- Después habrá que compilar la aplicación en un war. Primero hacemos un maven clean para asegurarnos que la carpeta target está vacía, y después ejecutamos maven install:
  - Click derecho sobre el proyecto > Run as > **Maven clean**
  - Click derecho sobre el proyecto > Run as > **Maven install**

Tras seguir estos pasos se habrá generado un archivo .war en la carpeta target.

- Creamos un Dockerfile en el directorio raíz del proyecto java con las siguientes instrucciones:

```
FROM openjdk:11-jdk-slim-buster

VOLUME /tmp

ARG JAR_FILE=target/GFit-Analysis-Back-0.0.1-SNAPSHOT.war

ADD ${JAR_FILE} app.jar

ENTRYPOINT ["java", "-Dserver.port=$PORT",
"-XX:MaxRAM=400M", "-Djava.security.egd=file:/dev/./urandom", "-jar",
/app.jar"]
```

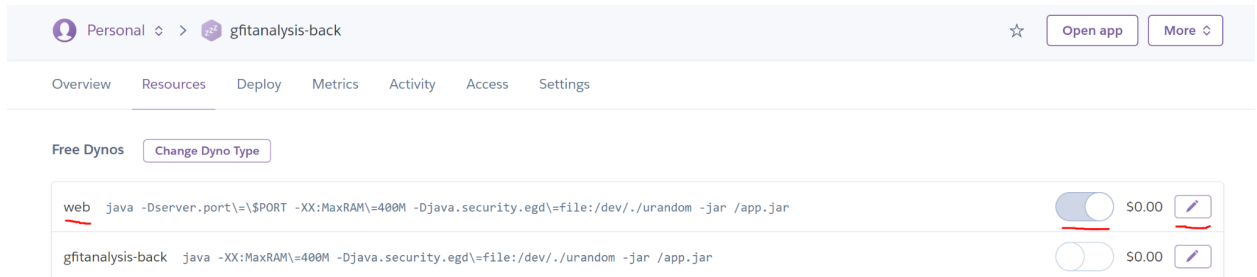
El archivo dockerfile también se puede encontrar en el zip del proyecto completo o en repositorio de github.

- Abrimos el terminal y nos movemos al directorio raíz del proyecto java (**GFit-Analysis\src\GFit-Analysis-Back**), desde ahí ejecutamos el siguiente comando para crear una imagen docker: “**docker build -t web .**”

La imagen debe llamarse web para que heroku lo reconozca como un servicio web.

- Lo siguiente es subir esa imagen a heroku para que este construya y despliegue la aplicación, pero antes se debe instalar el cli de heroku, lo podrá instalar con el comando “**npm install -g heroku**”, tras instalarlo ejecutar los siguientes comandos:
  - **heroku login**
  - **heroku container:login**

- **heroku container:push web -a** [nombre de la aplicación creada en heroku]
- **heroku container:release web -a** [nombre de la aplicación creada en heroku]
- Tras desplegar la release deberá dirigirse a la web de heroku y activar el dyno creado a partir de la imagen:



Una vez activado se ejecutará la aplicación y ya estará disponible el servicio rest del backend.

## Front-end

Antes de hacer el despliegue es **importante cambiar las url del proyecto** que hacen peticiones a la api del backend para que funcione correctamente. Estas URL se encuentran en los métodos de los servicios **"auth.service.ts"** y **"data-api.service.ts"** en la ruta **"GFit-Analysis\src\GFit-Analysis-Front\src\app\services"**

Para desplegar el front hay que crear un proyecto en la consola de firebase.

Tras crear el proyecto dirigirse a la pestaña "Hosting", te indicará cómo puedes desplegar la aplicación en firebase.

- Primero instalar el cli de firebase con el comando **"npm install -g firebase-tools"**
- Desde el directorio raíz del front (**GFit-Analysis\src\GFit-Analysis-Front**) ejecutar los siguientes comandos:
  - **firebase login**
  - **firebase init**

Lo siguiente a tener en cuenta es que en el proyecto la ruta del build de angular está cambiada para que se compile todo en la carpeta **"public"**, que es la carpeta por defecto que utilizará firebase.

- **ng build**
- **firebase deploy**

- Se puede comprobar la dirección donde está la aplicación desplegada desde la sección "Hosting" del proyecto firebase.