

Tenemos 2 aplicaciones desarrolladas con Sprint Boot y necesitamos crear las imágenes Docker para cada uno. Entonces cada uno en localhost debemos de ejecutarlas en puertos diferentes

## **\_1. Debemos de tener configurado el Dockerfile con el puerto indicado**

\_1a. flujox.jar = **8080**

```
FROM openjdk:8-jdk-alpine
MAINTAINER IBM
COPY flujox.jar /home/flujox.jar
CMD ["java", "-jar", "-Dserver.port=8080", "/home/flujox.jar"]
```

\_1b. flujoy.jar = **8085**

```
FROM openjdk:8-jdk-alpine
MAINTAINER IBM
COPY flujoy.jar /home/flujoy.jar
CMD ["java", "-jar", "-Dserver.port=8085", "/home/flujoy.jar"]
```

## **\_2. Ejecutar los comandos para generar las imágenes en docker localhost**

Nota: Tener en consideración el comando «run»

*docker run -d -p puertopublico:puertointerno nombredeimagedocker*

### **\_2a. Pasos para ejecutar la aplicación flujox:8080**

*step1:*

*docker build C:\IBMjupari\flujox -tag flujox:latest*

*step2:*

*docker run -d -p 8080:8080 flujox:latest*

### **\_2b. Pasos para ejecutar la aplicación flujoy**

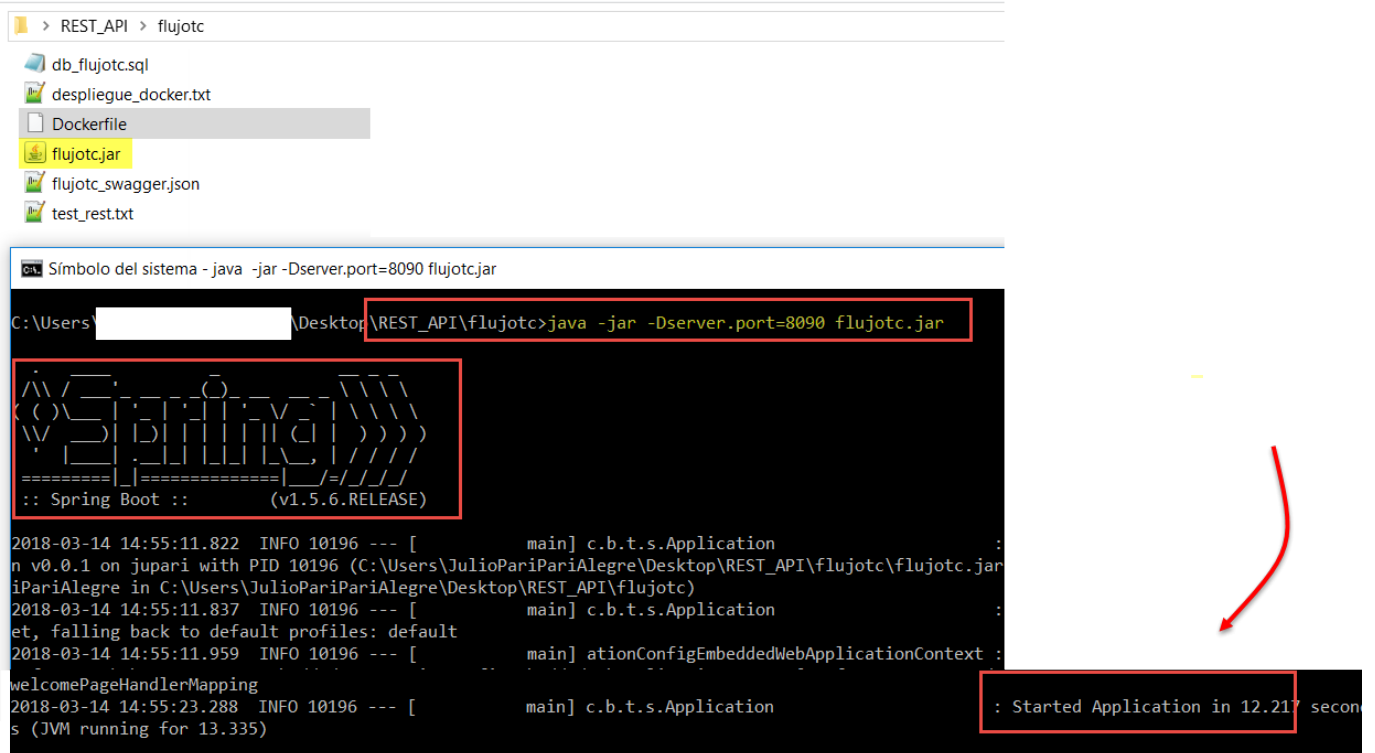
*step1:*

*docker build C:\IBMjupari\flujoy -tag flujoy:latest*

*step2:*

*docker run -d -p 8085:8085 flujoy:latest*

NOTA: A nivel de local tambien podemos ejecutar nuestras aplicaciones sprint boot desde la consola de comandos



### Ejecución de comando

```
java -jar -Dserver.port=1234 flujotc.jar
```

### Julio Pari (IT Architect IBM)



Si te ha interesado este artículo y deseas un apoyo o asesoría en algún requerimiento, envíame un mensaje a: (info@juliopari.com) o sino a través de LinkedIn: <https://www.linkedin.com/in/juliopari/>