

Manual de Despliegue

Paso 1: Localizar los archivos de la aplicación

Para comenzar con el despliegue es necesario ubicarse en la carpeta donde se encuentra las siguientes carpetas (descomprimidas) y archivos:

- reservas-laboratorios-api
- reservas-laboratorios-ui
- docker-compose.yml

En caso de no disponer de los archivos, se puede descargar desde el siguiente enlace:

<https://github.com/tomgreef/reservas-laboratorios>

Paso 2: Generar el Archivo JAR de Spring Boot

1. Usando la Terminal:

- Abre una terminal.
- Navega a la carpeta del proyecto **reservas-laboratorios-api**.
- Ejecuta el comando:

```
./mvnw clean package
```

- El archivo .jar se generará en el directorio **target**.

2. Usando Eclipse:

- Abre Eclipse.
- Importa el proyecto **reservas-laboratorios-api**.
- Haz clic derecho en el proyecto en el Explorador de Proyectos.
- Selecciona "Run As" > "Maven build...".
- En el campo "Goals", ingresa: **clean package**.
- Haz clic en "Run".
- El archivo .jar se generará en el directorio **target**.

3. Usando IntelliJ IDEA:

- Abre IntelliJ IDEA.
- Importa el proyecto **reservas-laboratorios-api**.
- En la ventana de Herramientas de Maven, busca tu proyecto.
- Expande el árbol de carpetas de tu proyecto.
- Selecciona "Lifecycle" > "clean".
- Luego, selecciona "Lifecycle" > "package".
- El archivo .jar se generará en el directorio **target**.

Paso 3: Ejecutar la Solución con Docker

1. Asegúrate de tener Docker Desktop instalado en tu máquina y arrancado.
2. Abre una terminal y navega al directorio que contiene tu archivo `docker-compose.yml`.
3. Ejecuta el siguiente comando para construir y arrancar los contenedores Docker:

```
docker-compose up -d --build
```

La configuración de Docker Compose creará y arrancará los un conjunto de contenedores llamado "reservas-laboratorios", que contendrá los siguientes contenedores: "api", "ui", "postgres".

Acceso a la Aplicación:

- La interfaz de usuario (UI) estará accesible en `http://localhost`. El contenedor "ui" mapea su puerto 80 al host.
- La API del back-end estará accesible en `http://localhost:8080`. El contenedor "api" mapea su puerto 8080 al host. También dispone de una interfaz de swagger accesible en `http://localhost:8080/swagger-ui/index.html`.

Detener los Contenedores:

- Para detener los contenedores Docker, acceda a Docker Desktop y pare el contenedor `reservas-laboratorios`

Actualizar la Aplicación:

- Para actualizar la aplicación, modifica el código fuente en los directorios de proyectos respectivos (reservas-laboratorios-api o reservas-laboratorios-ui). En el caso de modificar el código fuente de la API, es necesario generar nuevamente el archivo `.jar` de Spring Boot.
- Reconstruye el contenedor Docker correspondiente ejecutando nuevamente `docker-compose up -d --build`.

Limpieza:

- Para eliminar los contenedores y recursos asociados, ejecuta desde el directorio que contiene tu archivo `docker-compose.yml`:

```
docker-compose down
```

Nota: Este manual proporciona instrucciones para ejecutar la solución Docker y generar el archivo `.jar` necesario para el proyecto Java usando varias herramientas. Asegúrate de ajustar cualquier detalle específico a tu entorno y flujo de trabajo.