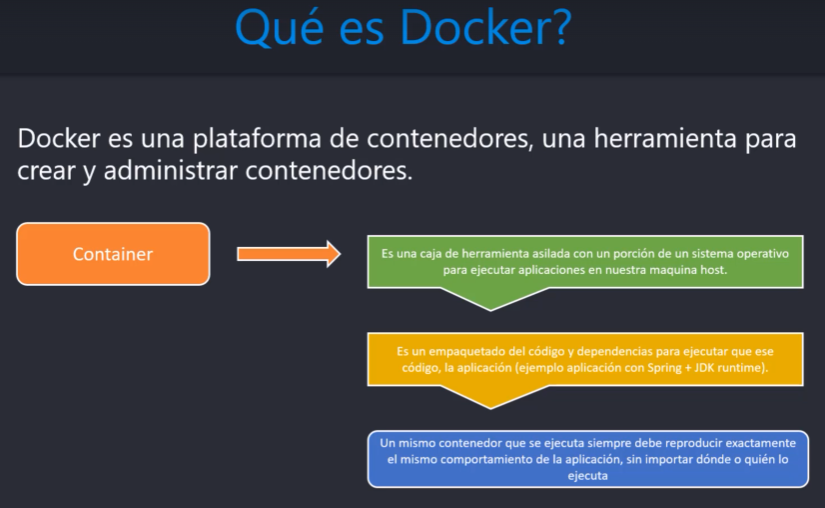
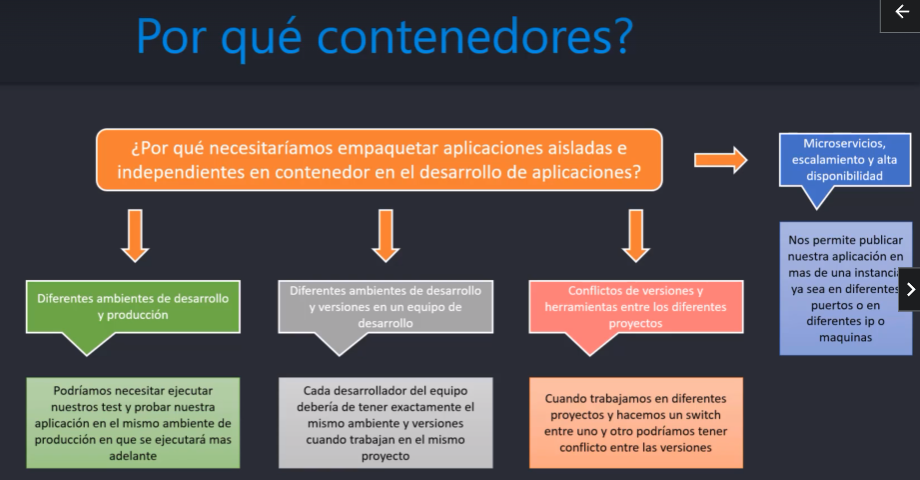
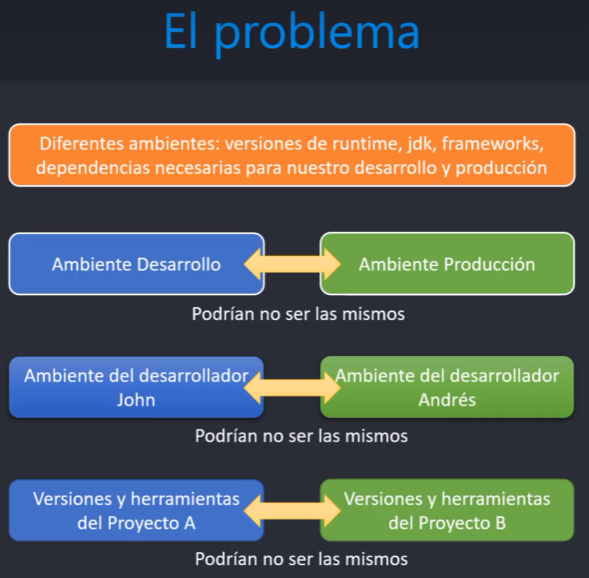
6. Introducción a Docker

Introducción a contenedores Docker

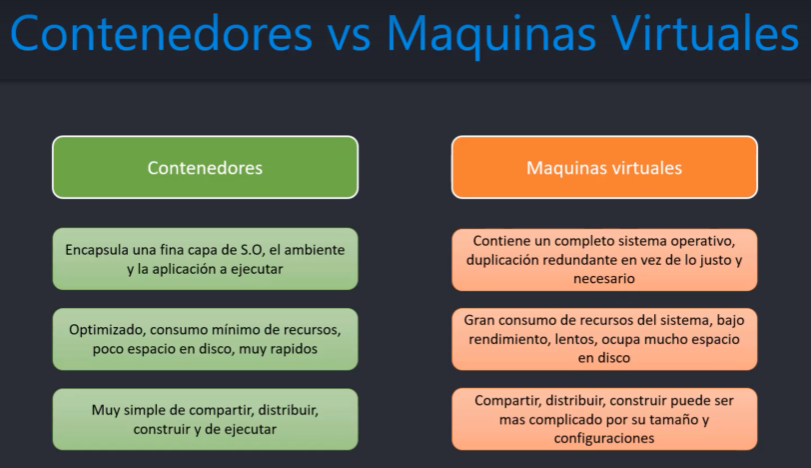
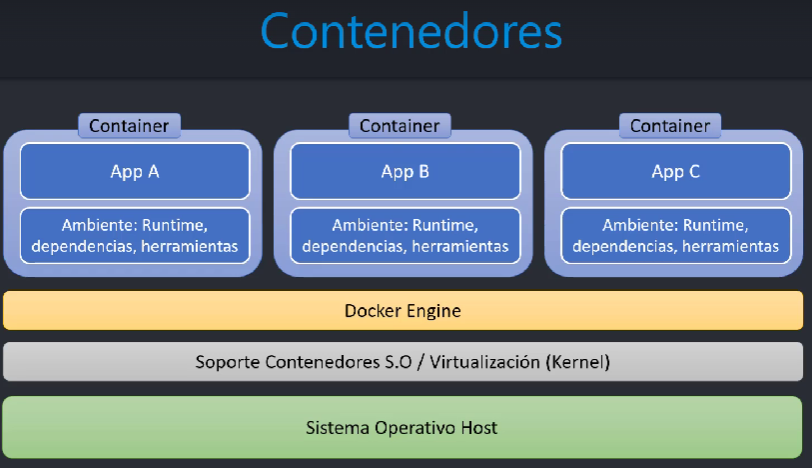


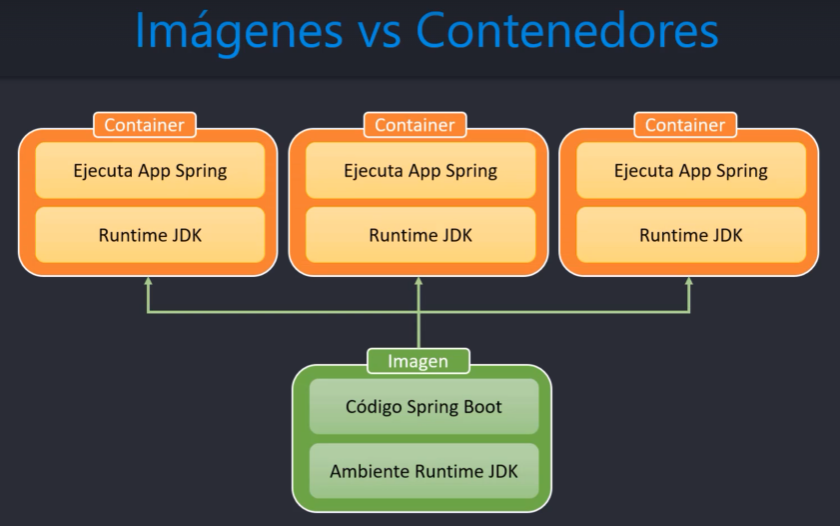
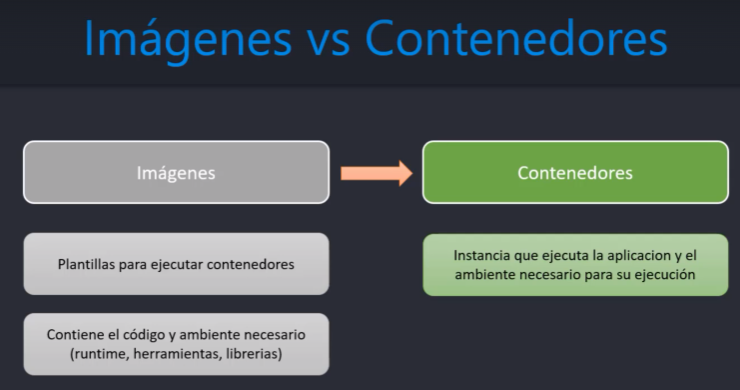
Ventajas de los contenedores en el desarrollo





Arquitectura de Docker: Imágenes & Contenedores





 Instalación Docker Windows

Instalación Docker macOS

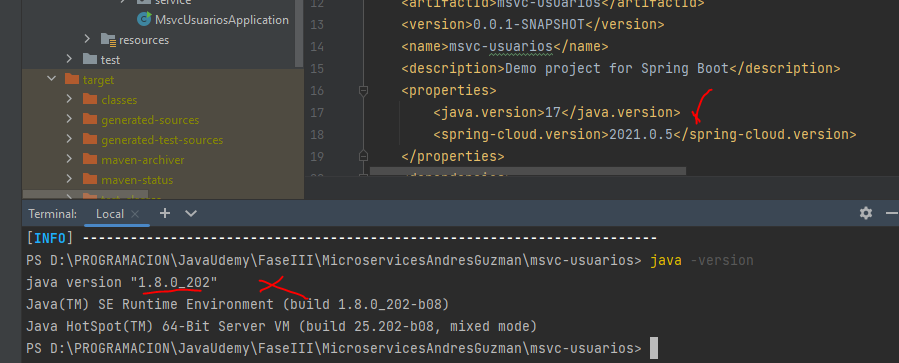
Instalación Docker Linux

Generando archivo jar para dockerizar

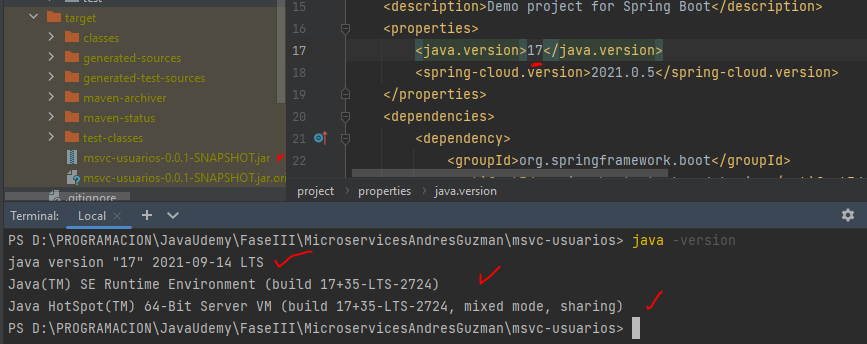
 Generando archivo jar para dockerizar

La idea es dockerizar el microservicio usuario y curso quedara de forma Local

* Primero validar que las versiones tanto en el sistema con el el pom del proyecto tengan la misma versión del jdk

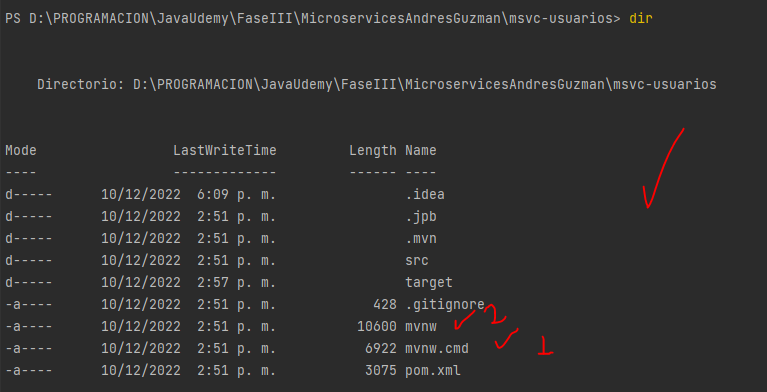


Como vemos las versiones no son iguales, el proyecto ejecuta bien el jar pero al probarlo da errores ya que el sitema esta configurando con jdk 1.8, para poder validarlo debo realizar el cambio en el sistema

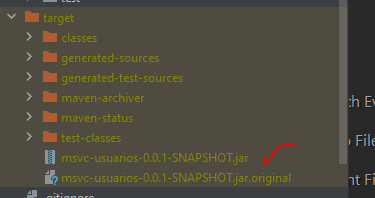


* Generando Jar Maven desde el terminal

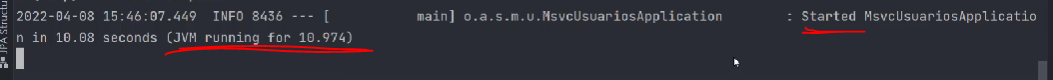
1. Para Windows
2. Para Cualquier SO

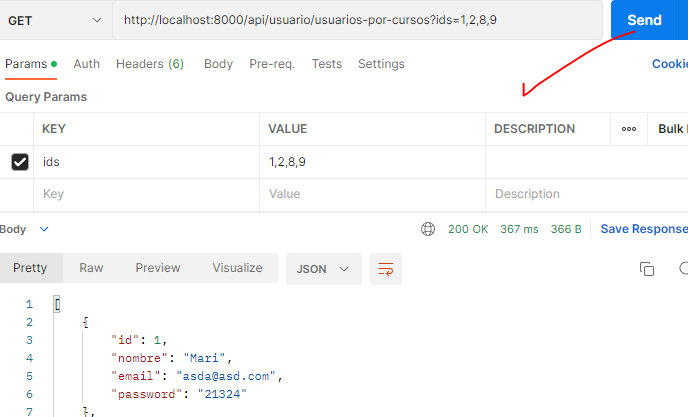


* Ejecutar el comando para limpiar si ahí Jar y crear uno nuevo ./mvnw.cmd clean package si da error aplico ./mvnw.cmd clean package -DskipTests



* Ejecuto el Jar para probar si esta Ok y realizo un prueba java -jar .\target\msvc-usuarios-0.0.1-SNAPSHOT.jar





Para detener el Jar lo puedo realizar desde el Administrador de tareas -> Detalles 🡪 Java.exe o darle varias veces ctrl+c en la terminal que se ejecutó.

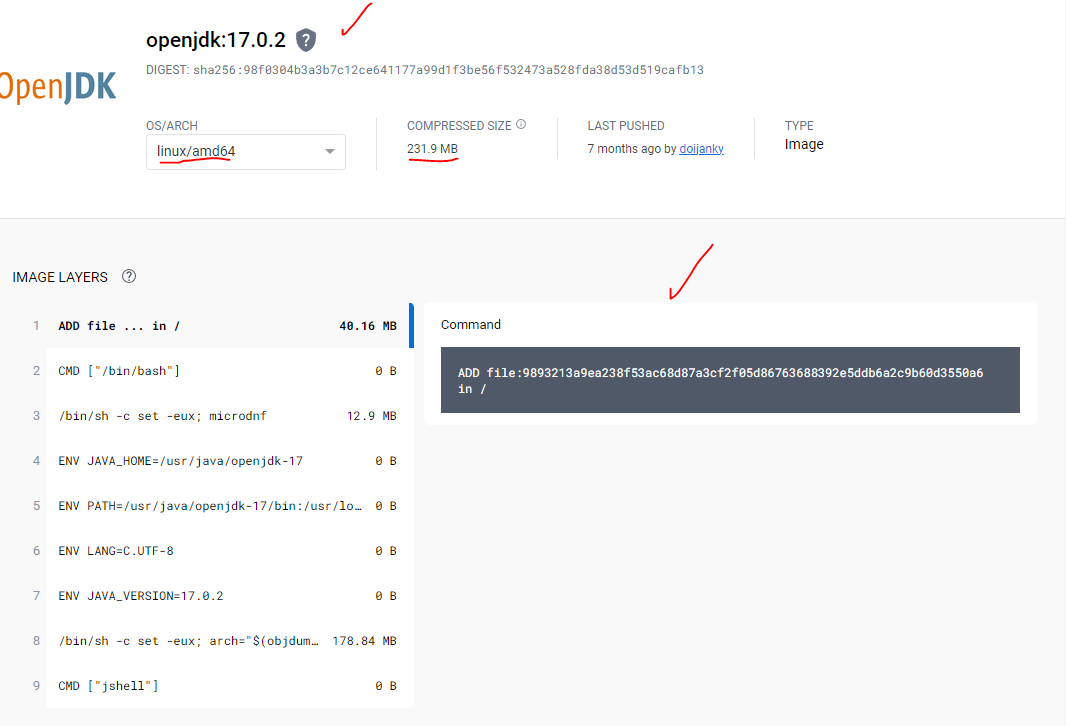
Creando archivo Dockerfile usando imagen OpenJDK

Como la idea es dockerizar cada uno de los microservicios existen dos formas de crear la imagen de estos y son

* Forma existente(Proyecto ya existente), mediante DockerHub
* Creación de un archivo Docker File pero igualmente se debe basar en una imagen en lo posible oficial ya existente en dockerHub.

Procedo a realizar los pasos para dockerizar los microservicios de forma DockerFIle

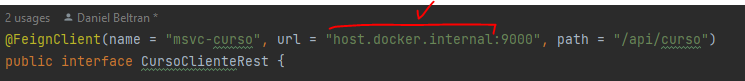
* Procedo a descargar el OpenJdk en la versión 17, OJO EN ARQUITECTURA LINUX que esta mi proyecto <https://hub.docker.com/_/openjdk/tags>



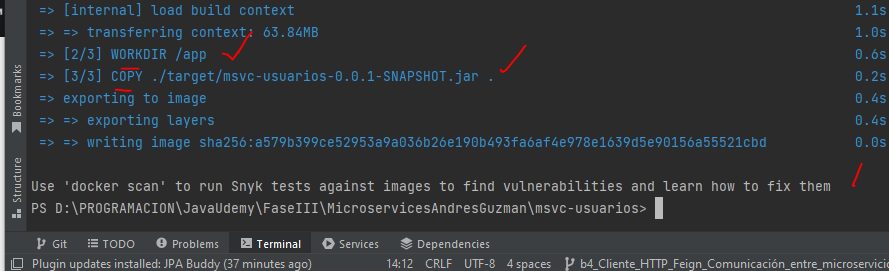
* Creando el archivo msvc-usuarios.DockerFile
* Instalo el plugin de Docker en Intellij
* Configurando el archivo Dockerfile

Construyendo nuestra primera imagen con Dockerfile y corriendo contenedor

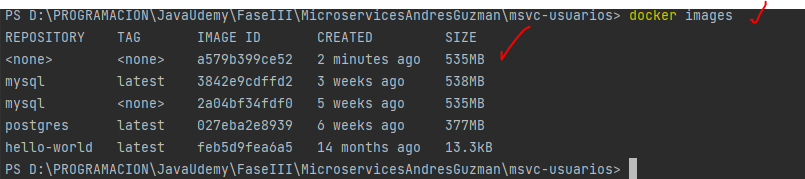
* Listo el archivo Dockerfile
* Construyendo la imagen en la ruta raíz del proyecto y en la terminal con el siguiente comando docker build . no le agrego la ruta porque estoy dentro de ella con la terminal que ejecuta el comando
* Importante unificar los endpoints o url de conexión entre los microservicios para este caso la conexión de usuario y curso en la clase CursoClienteRest la url sera host.docker.internal



Cada una de las líneas que tenga el archivo DockerFile sera una layer o una capa para la creación de la imagen.

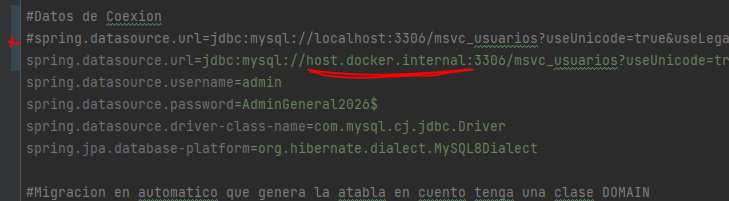


* Listando las imágenes Docker, valido que se haya creado la del microservicio usuario



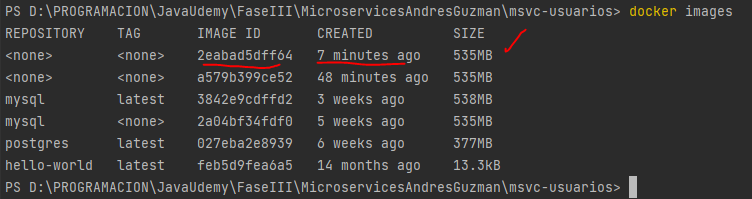
1. IMPORTANTE: Como el ing esta realizando las pruebas de DB por medio de app instaladas como postgres y mysql cuando levante el contenedor de microservicio usuario debo ajustarle la ruta del dataSource para que apunte a estas apps instalas, yo debo realizar lo mismo por que las DB(Mysql - Postgre) se encuentran dockerizadas, OJO si se ajusta el DataSource Nuevamente se tiene que realizar el jar y crear la imagen.

Cambia la ruta de localhost por

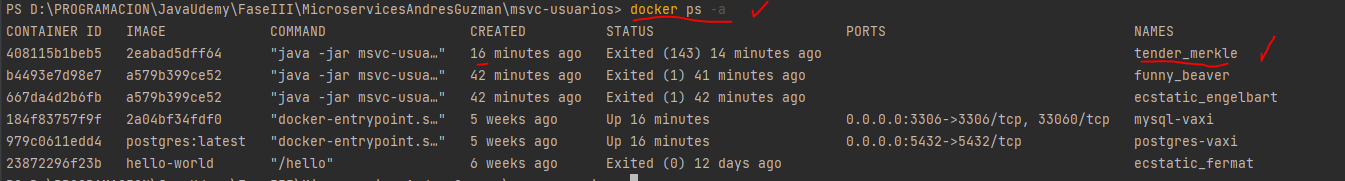


IMPORTANTE

* Listo nuevamente los contenedores, para tener el id del con el nuevo Jar



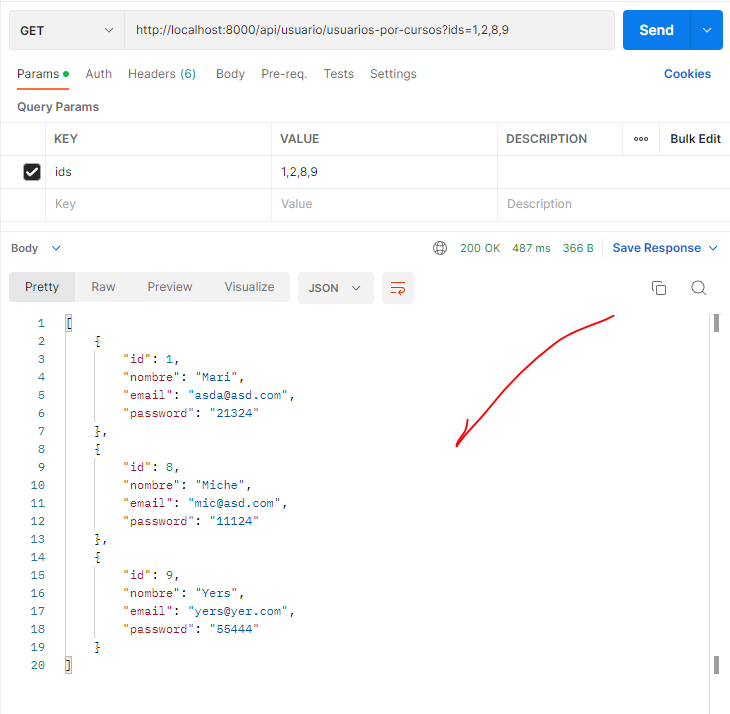
* Ver el detalle de los contendores creados Parva validar el que tiene nuestro Jar y es tender\_merkelet



* Corriendo el el contenedor microservicio-usuario por medio del id y le doy el puerto interno y asgino el puerto externo (Ya levantado los contenedores de DB)

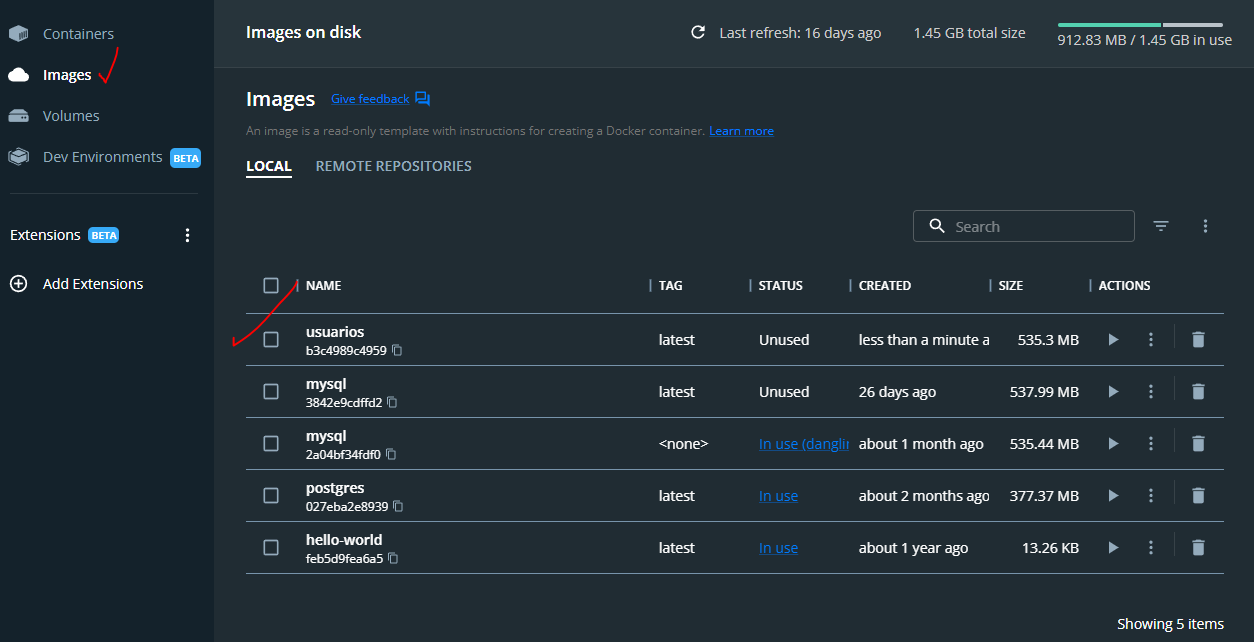
docker run -p 8000:8000 2eabad5dff64

* Pruebas(Todo Ok)

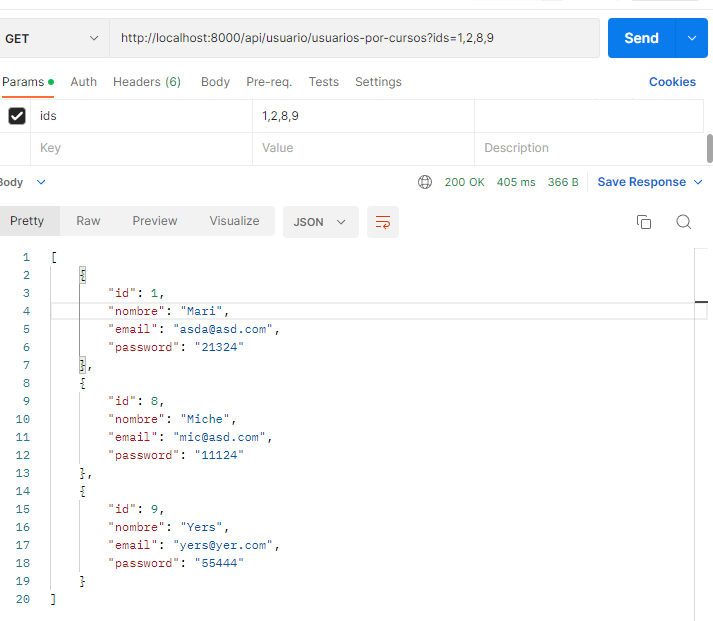


Comunicación entre aplicación en contenedor y aplicación local

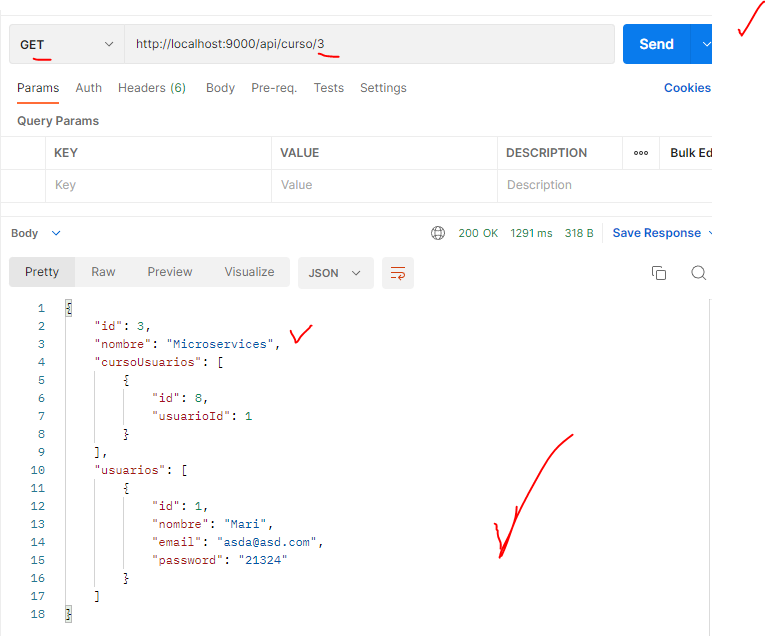
* Ya actualizada la url o enpoint del feign de usuarios procedo a generar nuevamente el Jar y el contenedor del servicio usuarios.
  + Crear nuevamente la imagen que contiene el microserivicio docker build -t usuarios .



* Listo ahora procedo a crear el contenedor a base de la imagen usuarios docker run --name msvc-usuarios -d -p 8000:8000 usuarios:latest
* Prueba



* Generar el jar para el microservicio de curso, este se manejara de forma LOCAL y no dockerizado
  + Generar el Jar
    - ./mvnw.cmd clean package -DskipTest
  + Levanto el jar
    - java -jar .\target\msvc-curso-0.0.1-SNAPSHOT.jar
* Pruebas consultando Curso(local) y usuarios(Docker)
* Consultando el curso con id=3 Microservices y muestra los usuarios que se encuentran en este curso



 Resumiendo un poco

