

**2020**

Services & Processes Solutions

Contenido

[**MICROSERVICIOS y Contenedores** 2](#_Toc30759670)

[**Alcance** 2](#_Toc30759671)

[**Creación de recursos necesarios para el microservicio** 2](#_Toc30759672)

[**Pruebas de consumo del microservicio** 5](#_Toc30759673)

[**Contenerización del microservicio** 6](#_Toc30759674)

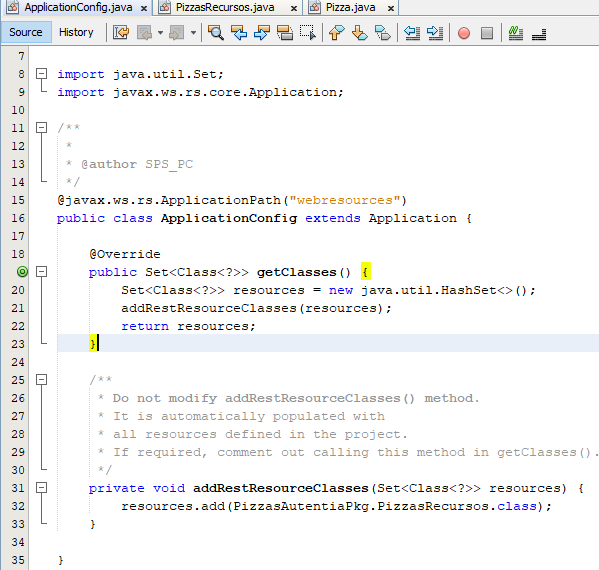
# **MICROSERVICIOS y Contenedores**

## **Alcance**

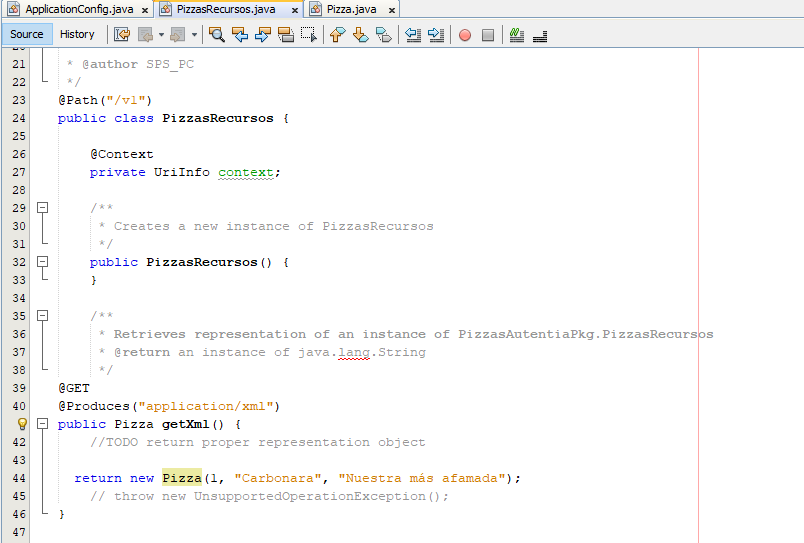
El presente documento describe el proceso de creación, configuración, despliegue y pruebas de un microservicio dentro de un contenedor en este caso Docker.

## **Creación de recursos necesarios para el microservicio**

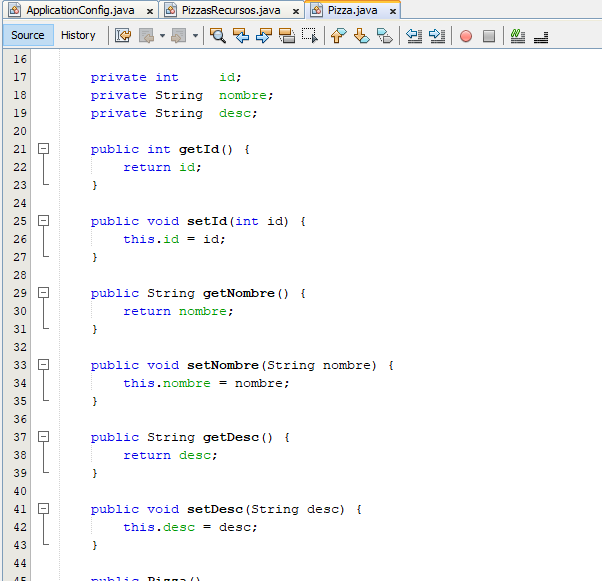
1.- Se creo la clase java ApplicationConfig.java con la siguiente estructura:



2.- Se creo la siguiente clase con el nombre PizzaRecursos.java, en el cual se definieron los métodos para invocar al servicio.



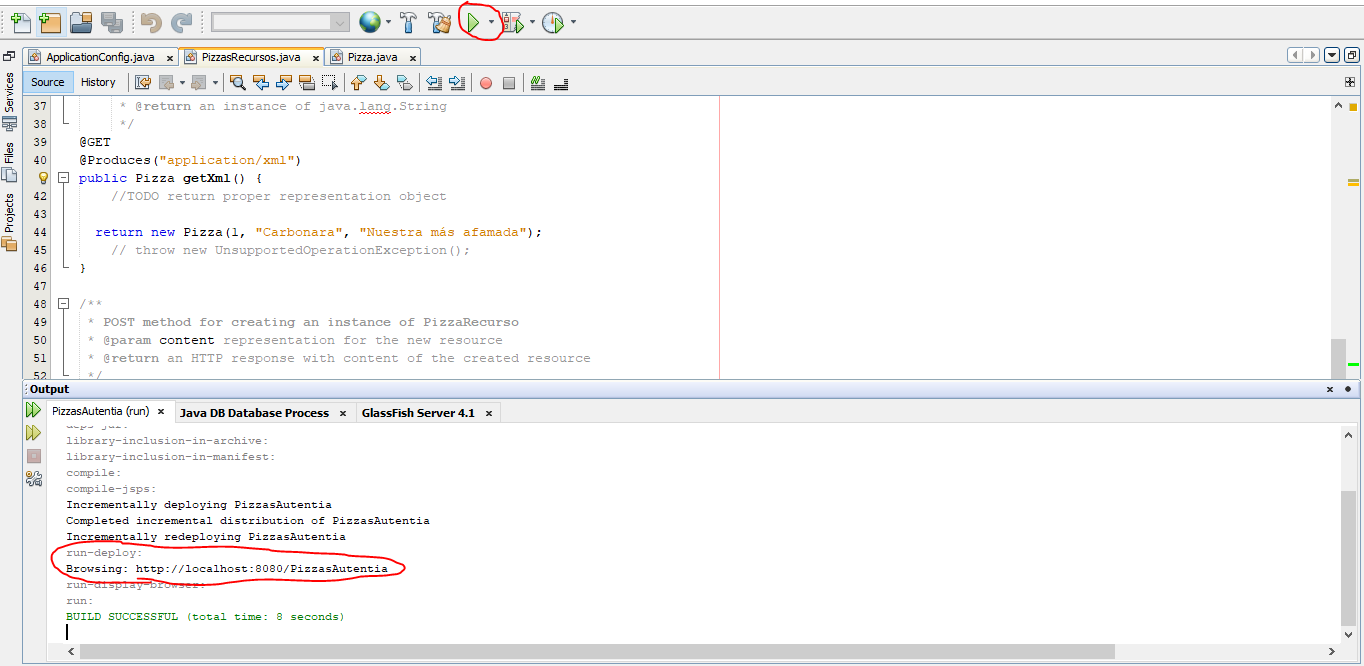
3.- Se creo la siguiente clase Pizza.java con las propiedades de la pizza, en este caso id, nombre y desc, así como sus respectivos setters y getters:



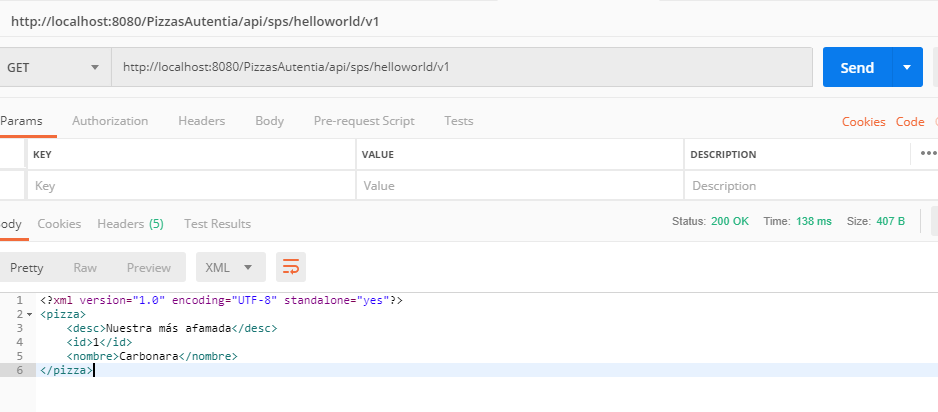
## **Pruebas de consumo del microservicio**

Después de compilar el servicio se realiza el despliegue en el servidor local, en este caso al ser Netbeans se desplego con glassfish.

Al terminar de desplegarse veremos el siguiente mensaje:



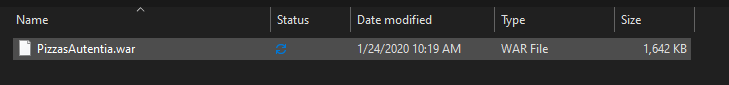
Con esto ya podemos realizar el consumo de nuestro servicio, en mi caso será con Postman como se muestra a continuación:



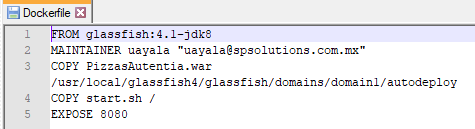
## **Contenerización del microservicio**

1.- Para este ejemplo crearemos una carpeta en la unidad C con el nombre de Microservicio y copiamos el archivo .war:

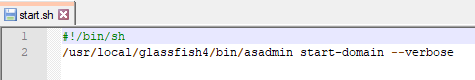
C:\Microservicio



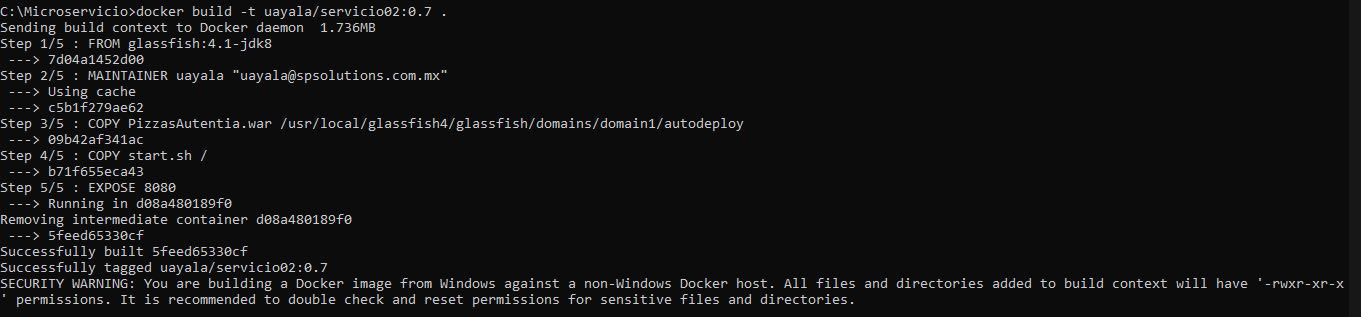
2.- Se debe crear el archivo DockerFile dentro de la carpeta Microservicio, con la siguiente estructura:



3.- Se debe crear el archivo start.sh dentro de la carpeta Microservicio, con la siguiente estructura:



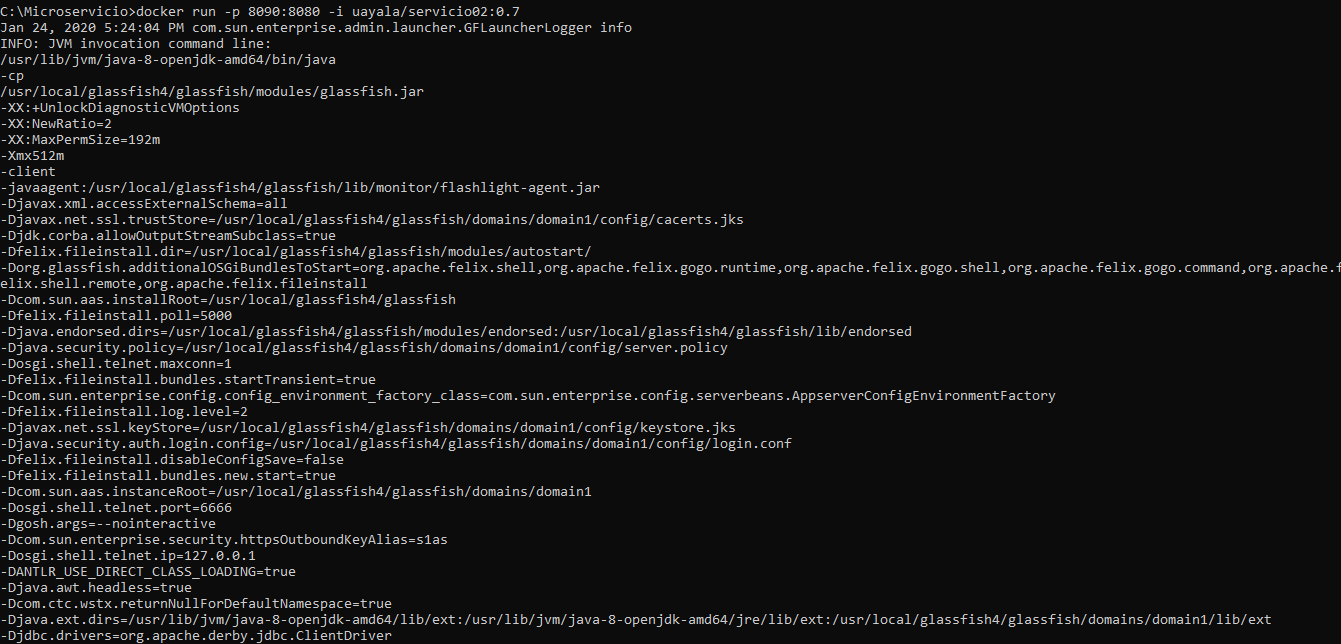
4.- Con los archivos anteriores ya podemos realizar la imagen del docker, en una ventana de línea de comandos ejecutando las siguientes sentencias:



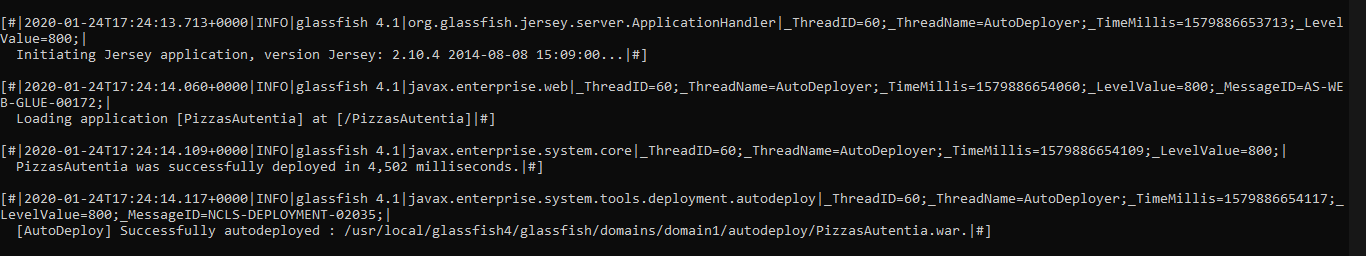
5.- Para validar que se creó correctamente ejecutamos el siguiente comando:



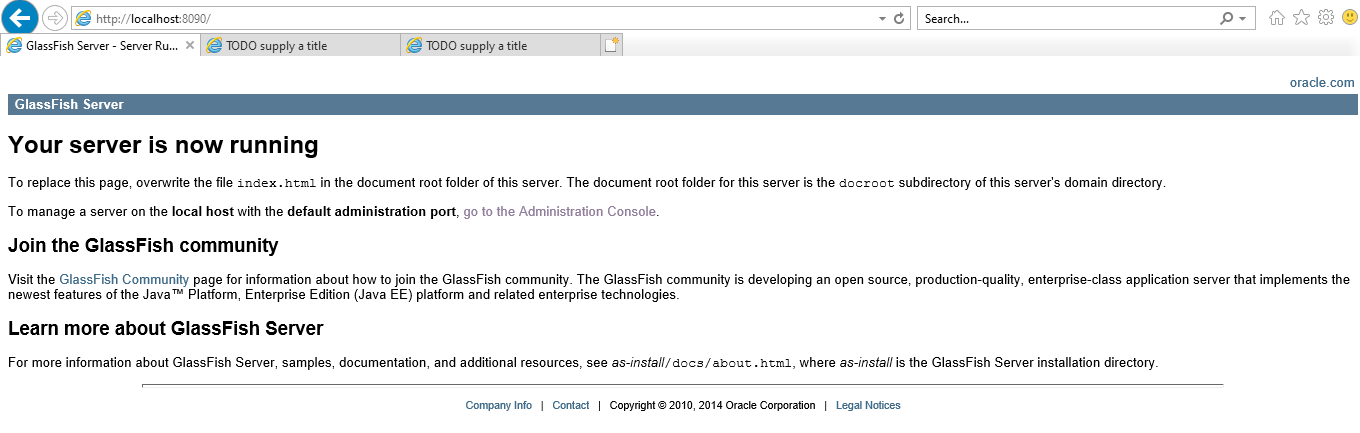
6.- En este paso ya se puede ejecutar la imagen del Docker creada previamente con el siguiente comando:



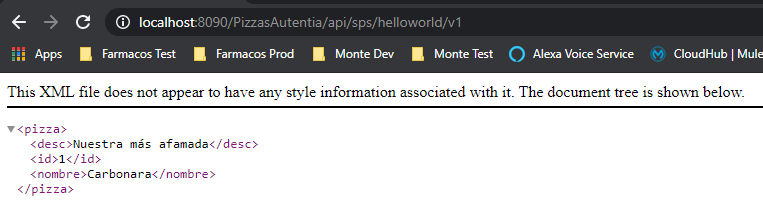
7.- Al final del log podemos ver que la aplicación fue desplegada correctamente:



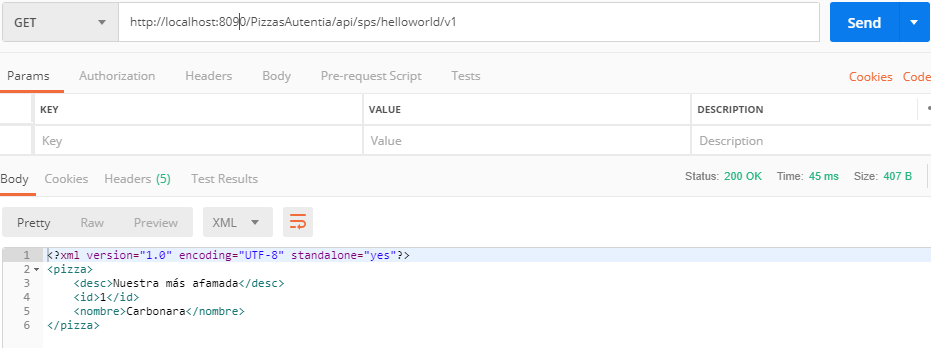
8.- Abrimos el navegador para validar el inicio correctamente:



9.- Es hora de validar nuestro microservicio ejecutándose dentro de Docker con el puerto 8090, en mi caso queda de esta manera: http://localhost:8090/PizzasAutentia/api/sps/helloworld/v1



Ya tenemos nuestro microservicio en java ejecutándose dentro de Docker, aquí una prueba mediante postman:



**S&PSolutions**