



Codelab Comunicación asincrónica entre microservicios

Desarrollo de Software III

Jhojan Stiven Castaño Jejen-2259476

Universidad Del Valle

Sede Tuluá

Septimo Semestre

2025

Evidencias sobre elaboración del codelab

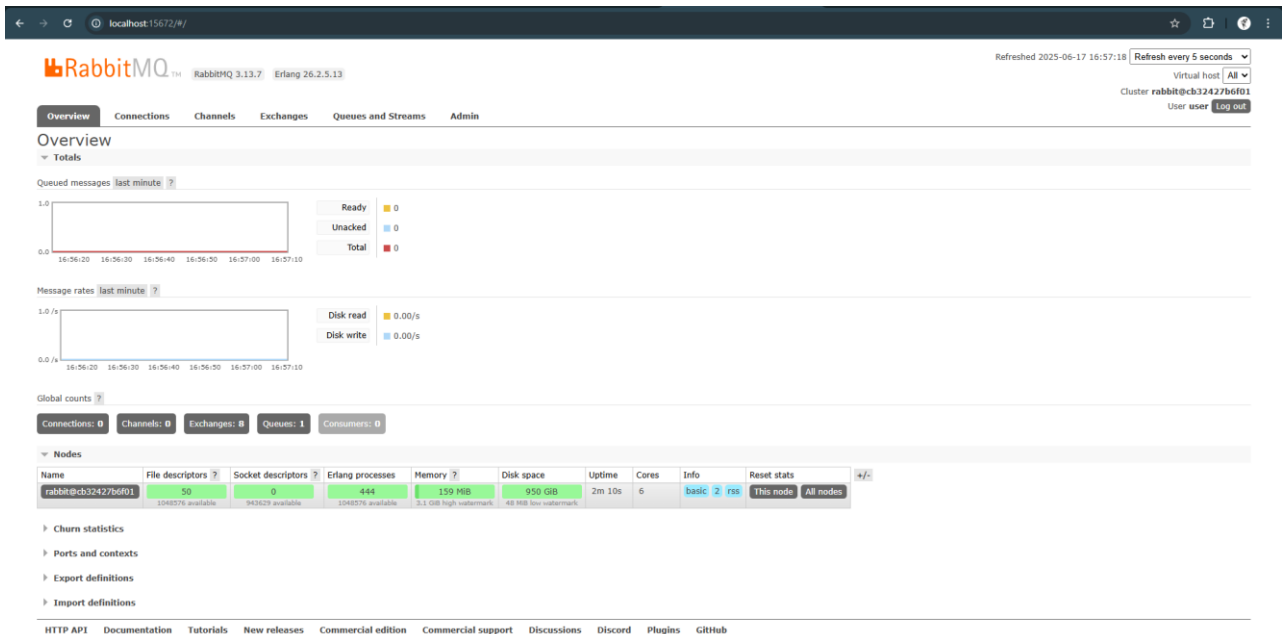
Creacion del Docker-compose.yml

```
docker-compose.yml X
C:\Users\EIRolote\Documents\Codelab\Codelab cinco Comunicacion asincronica > docker-compose.yml
1 services:
2   rabbitmq:
3     image: rabbitmq:3-management
4     container_name: rabbitmq
5     ports:
6       - "5672:5672" # puerto AMQP
7       - "15672:15672" # interfaz de administraci
8     environment:
9       RABBITMQ_DEFAULT_USER: user
10      RABBITMQ_DEFAULT_PASS: password
11
```

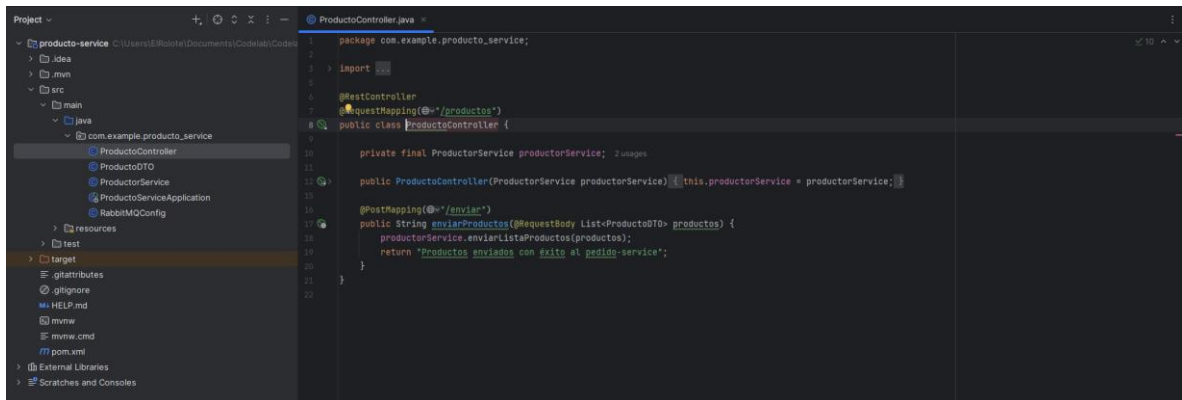
```
MINGW64:/c:/Users/EIRolote/Documents/Codelab/Codelab cinco Comunica...
ElRolote@DESKTOP-HP6I80S MINGW64 ~
$ cd Documents/Codelab/Codelab\ cinco\ Comunicacion\ asincronica/

ElRolote@DESKTOP-HP6I80S MINGW64 ~/Documents/Codelab/Codelab cinco Comunicacion asincronica
$ docker-compose up -d
[+] Running 1/1
✓ Container rabbitmq Started 0.8s
```

Ingresamos al <http://localhost:15672> con el user y password



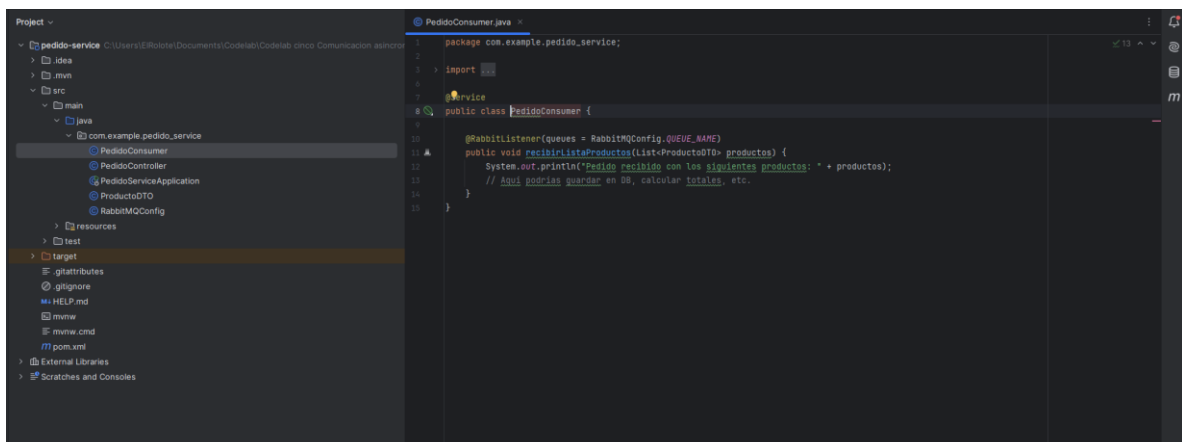
Creacion de producto-service



The screenshot shows an IDE with a project named 'producto-service'. The left sidebar displays the project structure, including 'src/main/java/com.example.producto_service'. The main editor window shows the code for 'ProductoController.java'.

```
1 package com.example.producto_service;
2
3 import org.springframework.web.bind.annotation.*;
4
5 @RestController
6 @RequestMapping("/productos")
7 public class ProductoController {
8
9     private final ProductService productService;
10
11     public ProductoController(ProductService productService) {
12         this.productService = productService;
13     }
14
15     @PostMapping("/enviar")
16     public String enviarProductos(@RequestBody List<ProductoDTO> productos) {
17         productService.enviarListaProductos(productos);
18         return "Productos enviados con éxito al pedido-service";
19     }
20 }
```

Creacion de pedidos-service



The screenshot shows an IDE with a project named 'pedido-service'. The left sidebar displays the project structure, including 'src/main/java/com.example.pedido_service'. The main editor window shows the code for 'PedidoConsumer.java'.

```
1 package com.example.pedido_service;
2
3 import org.springframework.amqp.rabbit.annotation.*;
4
5 @Service
6 public class PedidoConsumer {
7
8     @RabbitListener(queues = RabbitMQConfig.QUEUE_NAME)
9     public void recibirListaProductos(List<ProductoDTO> productos) {
10         System.out.println("Pedido recibido con los siguientes productos: " + productos);
11         // Aquí podrías guardar en DB, calcular totales, etc.
12     }
13 }
```

Accedemos a postman, ingresamos un post con el siguiente url
<http://localhost:8081/productos/enviar>

