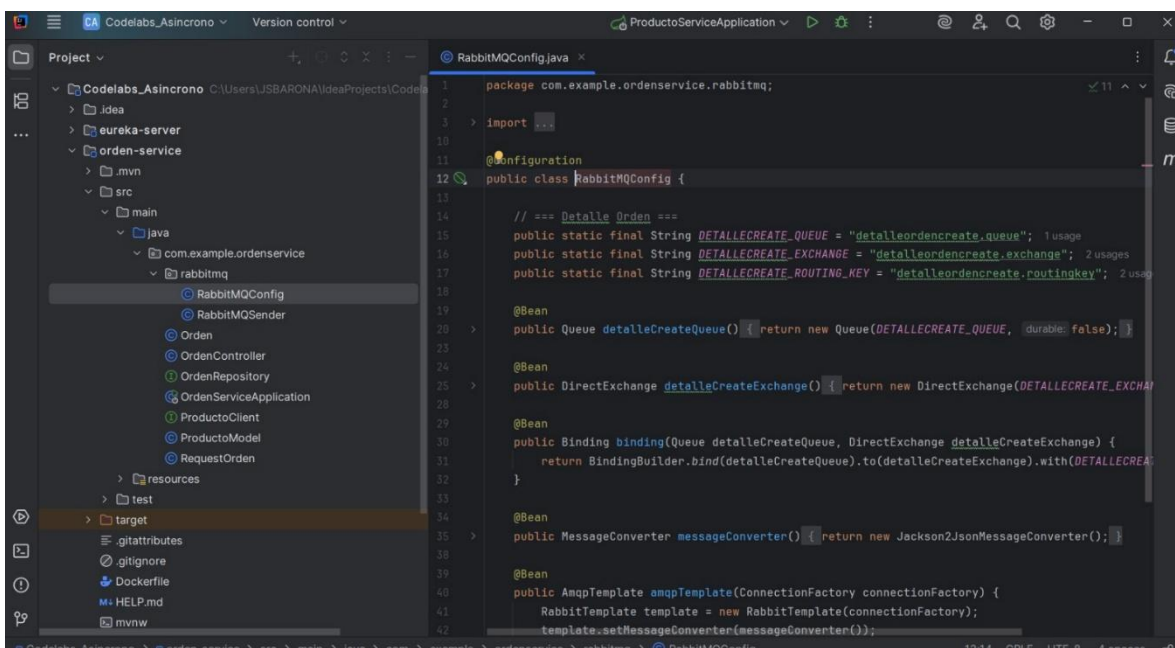


CODELAB 5

Cuando se crea una orden de compra, nuestro sistema manda automáticamente un mensaje a través de RabbitMQ al servicio de productos. Ese mensaje le dice al otro microservicio: ‘oye, ¿hay suficiente inventario de estos productos?’.

Así se hace la verificación sin tener que esperar una respuesta inmediata o hacer llamadas directas. Esto hace que todo sea más rápido, más escalable y que los servicios estén menos acoplados entre sí.



```
1 package com.example.orderservice.rabbitmq;
2
3 import ...
4
5 @Configuration
6 public class RabbitMQConfig {
7
8     // === Detalle Orden ===
9     public static final String DETALLECREATE_QUEUE = "detalleordencreate.queue"; 1 usage
10     public static final String DETALLECREATE_EXCHANGE = "detalleordencreate.exchange"; 2 usages
11     public static final String DETALLECREATE_ROUTING_KEY = "detalleordencreate.routingkey"; 2 usages
12
13     @Bean
14     public Queue detalleCreateQueue() { return new Queue(DETALLECREATE_QUEUE, durable: false); }
15
16     @Bean
17     public DirectExchange detalleCreateExchange() { return new DirectExchange(DETALLECREATE_EXCHANGE); }
18
19     @Bean
20     public Binding binding(Queue detalleCreateQueue, DirectExchange detalleCreateExchange) {
21         return BindingBuilder.bind(detalleCreateQueue).to(detalleCreateExchange).with(DETALLECREATE_ROUTING_KEY);
22     }
23
24     @Bean
25     public MessageConverter messageConverter() { return new Jackson2JsonMessageConverter(); }
26
27     @Bean
28     public AmqpTemplate amqpTemplate(ConnectionFactory connectionFactory) {
29         RabbitTemplate template = new RabbitTemplate(connectionFactory);
30         template.setMessageConverter(messageConverter());
31     }
32 }
```

