# ****📌 Roteiro de Testes Manuais****

### ****Introdução****

Este roteiro documenta os passos necessários para testar manualmente a aplicação, garantindo que os serviços essenciais como MongoDB, Redis e RabbitMQ estejam funcionando corretamente.

## ****1️⃣ Verificar se Todos os Containers Estão Rodando****

bash

CopiarEditar

docker ps

✅ **Verifique se os seguintes serviços estão rodando:**  
✔ order-api (porta 8080)  
✔ rabbitmq (porta 5672)  
✔ mongodb (porta 27017)  
✔ redis (porta 6379)

❌ **Se algum serviço não estiver rodando, reinicie todos os containers:**

bash

CopiarEditar

docker-compose down && docker-compose up -d --build

## ****2️⃣ Testar Conexão com o MongoDB****

bash

CopiarEditar

docker exec -it mongodb mongosh --eval "use orderdb; db.orders.find().pretty()"

## ****3️⃣ Testar Conexão com o Redis****

bash

CopiarEditar

docker exec -it redis redis-cli keys "\*"

## ****4️⃣ Testar Publicação de Mensagem no RabbitMQ (Simulando Produto-Externo-A)****

bash

CopiarEditar

docker exec -it rabbitmq rabbitmqadmin publish exchange=orders.generated.exchange routing\_key=orders.generated.key payload='{"orderId":"1234", "products": [{"name":"Produto Externo A","price":15.0,"quantity":2}] }'

## ****5️⃣ Testar o Consumo da Mensagem pela**** order-api

bash

CopiarEditar

docker logs order-api --tail 50

## ****6️⃣ Testar o Consumo da Mensagem na Fila**** orders.processed.queue

bash

CopiarEditar

docker exec -it rabbitmq rabbitmqadmin get queue=orders.processed.queue

## ****7️⃣ Testar a API Diretamente****

bash

CopiarEditar

curl -X POST http://localhost:8080/orders \

-H "Content-Type: application/json" \

-d '{"orderId":"999", "products": [{"name":"Produto Teste","price":50.0,"quantity":2}] }'