



## Desafio Backend BTG Pactual - Build & Run

Status

In progress

Assign

Empty

Link para o vídeo completo:

[www.youtube.com](https://www.youtube.com/watch?v=e_WgAB0Th_I)

[https://www.youtube.com/watch?v=e\\_WgAB0Th\\_I](https://www.youtube.com/watch?v=e_WgAB0Th_I)

Repositorio com o código do vídeo:

<https://github.com/buildrun-tech/buildrun-desafio-backend-btg-pactual>

Desafio: [problem.md](#) buildrun-tech/buildrun-desafio-backend-btg-pactual

### Desafio Backend BTG Pactual

**Escopo do desafio backend**

Construir um microserviço que seja capaz de:

- Processar pedidos a partir de uma fila RabbitMQ
- Criar uma API Rest que permita consultar:
  - Valor total de um pedido
  - Quantidade de pedidos por cliente
  - Lista de pedidos realizados por cliente

### Desafio Backend BTG Pactual

**Solução técnica:**

- A API deverá ser **Restfull**
- Podemos utilizar qualquer tecnologia para o microserviço (Java, Javascript, .Net, etc)
- Podemos utilizar qualquer estrutura de armazenamento (MongoDB, PostgreSQL, MySQL, etc)

API -> order-ms -> Banco de Dados

btg-pactual-order-created (RabbitMQ)

### Desafio Backend BTG Pactual

**Nossa escolha:**

- Java 21 com Spring Boot
- Banco de dados MongoDB

API -> order-ms -> Banco de Dados

btg-pactual-order-created (RabbitMQ)

spring boot

mongoDB

### Desafio Backend BTG Pactual

**Requisito técnico:**

- Segue um exemplo de mensagem que deverá ser consumida da fila RabbitMQ
- A partir dessa mensagem, estruture como será a modelagem de banco de dados e sua API;
- Lembre-se, a sua API deverá informar os seguintes itens:
  - Valor total de um pedido
  - Quantidade de pedidos por cliente
  - Lista de pedidos realizados por cliente

```
queue: btg-pactual-order-created
content-type: application/json

{
  "codigoPedido": 1001,
  "codigoCliente": 1,
  "items": [
    {
      "produto": "lápis",
      "quantidade": 100,
      "preco": 1.10
    },
    {
      "produto": "caderno",
      "quantidade": 10,
      "preco": 1.00
    }
  ]
}
```

# Bora codar!

**BUILD & RUN**

Resolveremos o Desafio Backend do BTG Pactual e aprenderemos:

- Como criar um microserviço com Spring Boot
- Como consumir uma fila do RabbitMQ
- Como comunicar com o banco de dados MongoDB via Docker
- Como mapear uma collection do MongoDB dentro do Spring
- Como fazer aggregations no MongoDB com Spring
- Como efetuar logs com o SLF4J

**Lista de tarefas:**

Introdução:

- ☐
- Qual é o desafio que vamos resolver?

Comunicação com o banco e dados:

- ☐ Iniciando o projeto Java (Web, Data MongoDB, RabbitMQ)
- ☐ Configurando o RabbitMQ e MongoDB no Docker
- ☐ Configurando a comunicacao do Spring Boot com o MongoDB
- ☐ Configurando a criacao da fila no RabbitMQ

Funcionalidade de Consumo de Pedidos:

- ☐ Mapear as entidades (Order, OrderItem)
- ☐ Criar o listener da fila do RabbitMQ
- ☐ Criar a implementação para salvar o pedido no MongoDB
- ☐ Testar o fluxo (rabbitmq → spring → mongodb)

Funcionalidade de disponibilizar as informações via API:

- ☐ Lembre-se, a API deverá informar:
  - Lista de pedidos realizados por cliente
  - Valor total de um pedido
  - Quantidade de pedidos por cliente
- ☐ Criar endpoint (estruturar DTO de retorno)
- ☐ Criar serviço de listagem de pedidos do cliente
- ☐ Criar serviço que calcula o valor total de todos os pedidos do cliente
- ☐ Testar a API