

FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA

Guilherme Janunzzi - RM: 558461

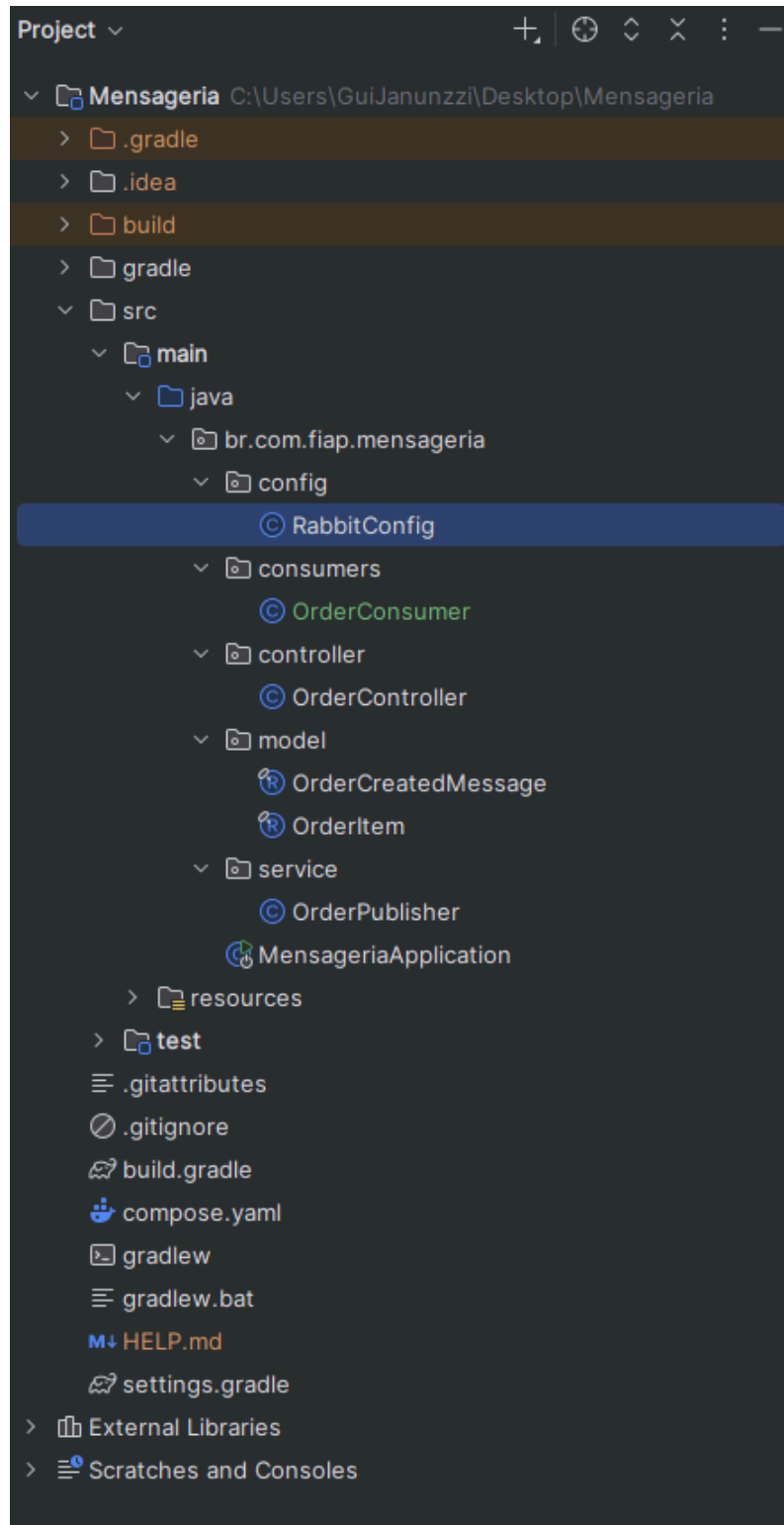
Mensageria

São Paulo/SP

2025

evidências / prints / registros

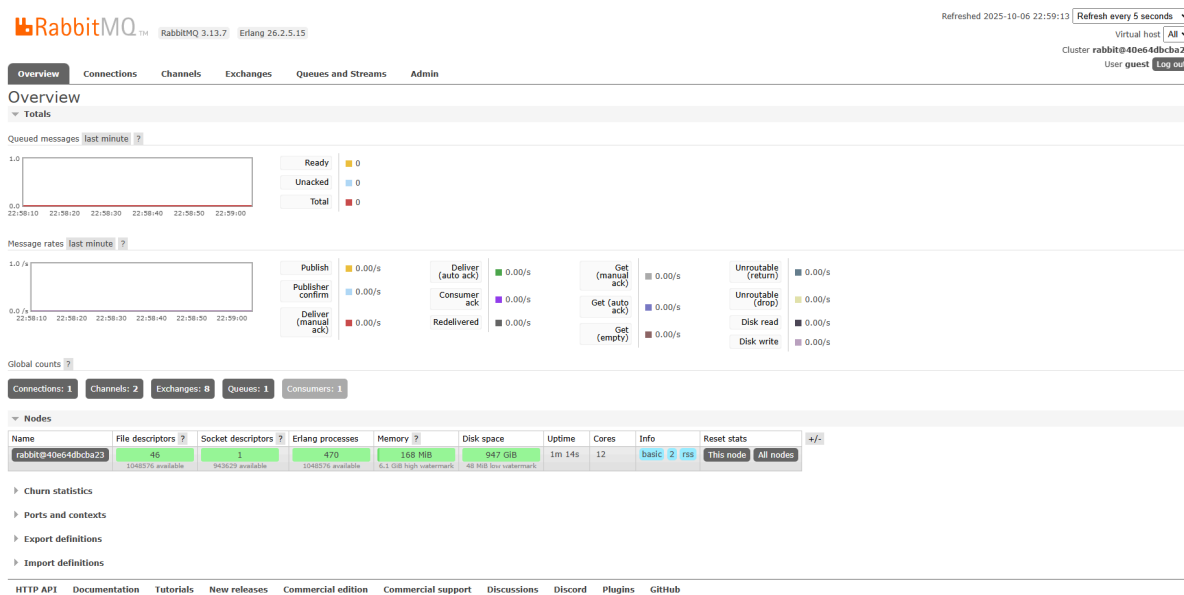
Print da estrutura de pastas do projeto:



Print dos logs da aplicação:

[illegible]

Print da interface web do RabbitMQ:



Queues

All queues (1)

Pagination

Page 1 of 1 - Filter: ☐ Regex

Displaying 1 item , page size up to: 100

Overview		Messages				Message rates				+/-
Virtual host	Name	Type	Features	State	Ready	Unacked	Total	Incoming	deliver / get	
/	order.queue	classic	D	running	0	0	0	0.00/s	0.00/s	0.00/s

Add a new queue

Queue order.queue

Overview

Queued messages last minute



Ready 0
Unacked 0
Total 0

Message rates last minute



Publish 0.00/s
Consumer ack 0.00/s
Get (auto ack) 0.00/s
Deliver (manual ack) 0.00/s
Redelivered 0.00/s
Get (empty) 0.00/s
Deliver (auto ack) 0.00/s
Get (manual ack) 0.00/s

Details

Features		State		Messages		Total	Ready	Unacked	In memory	Persistent	Transient, Paged Out
queue durable: true		Idle		Consumers 1		0	0	0	0	0	0
queue storage version: 1		Consumer capacity 100%		Message body bytes 0 B		0 B	0 B	0 B	0 B	0 B	0 B
Policy		Process memory 10 KiB									
Operator policy											
Effective policy definition											

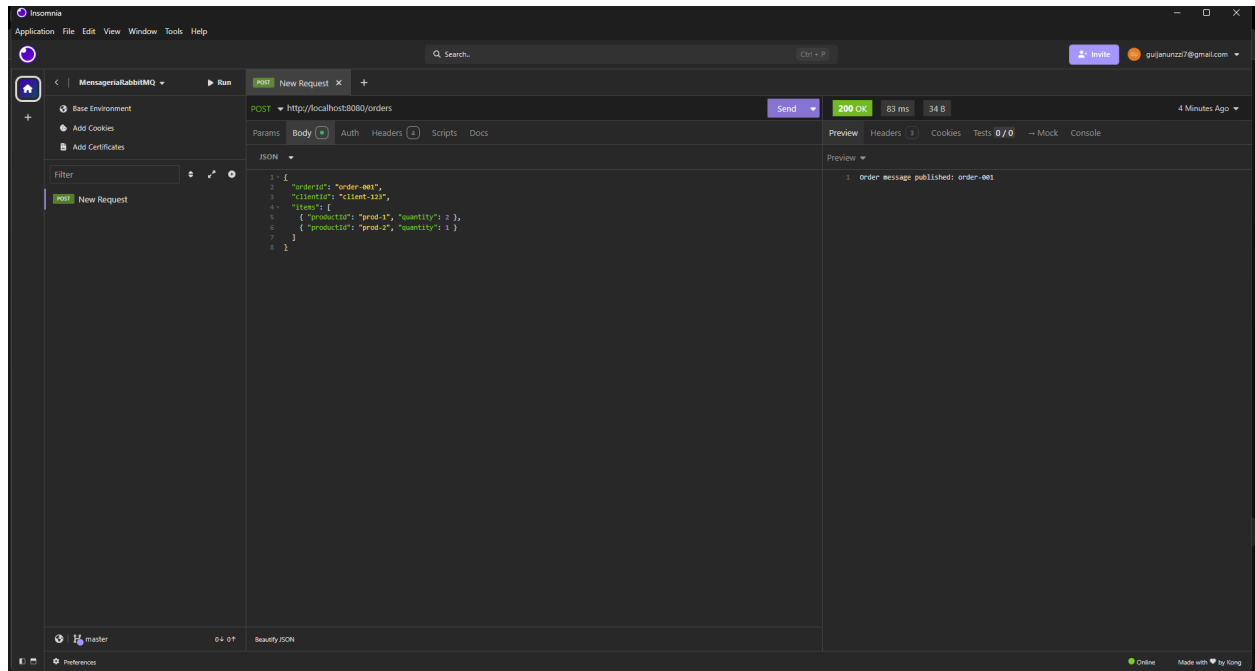
Consumers (1)

Bindings (2)

From	Routing key	Arguments
(Default exchange binding)	order.created	
order.exchange		Unbind

This queue

Print / captura de tela da requisição REST de teste:



Reflexão final / autoavaliação

1. Quais partes do sistema funcionaram exatamente como você esperava?

A configuração inicial da infraestrutura do RabbitMQ com os Beans do Spring (@Bean para Queue, Exchange e Binding) funcionou de forma muito direta e como esperado. Além disso, a criação do endpoint REST no OrderController e a injeção de dependências do Spring para obter o OrderPublisher foram processos fluidos e sem surpresas, demonstrando a robustez do ecossistema Spring Boot para tarefas padrão. O RabbitTemplate também se mostrou muito simples de usar para a publicação inicial.

2. Quais dificuldades ou erros você enfrentou que precisou corrigir?

Enfrentei três dificuldades principais que exigiram depuração e correção:

Erro de Compilação (cannot find symbol: variable log): Inicialmente, o projeto não compilava porque a variável log não existia. A correção foi configurar o Lombok corretamente, adicionando a anotação @Slf4j na classe e ajustando o build.gradle para usar compileOnly e annotationProcessor para a dependência do Lombok.

Erro na Publicação (SimpleMessageConverter only supports...): Ao tentar publicar a primeira mensagem, o sistema falhou porque o conversor padrão não sabia como serializar um record Java. A solução foi criar um @Bean de Jackson2JsonMessageConverter para que o Spring passasse a converter todas as mensagens para o formato JSON.

Erro no Consumo (MessageConversionException): Mesmo com o conversor JSON configurado, o consumidor falhava ao ler a mensagem. O diagnóstico revelou que, ao criar uma SimpleRabbitListenerContainerFactory customizada para o ack manual, ela não herdava o conversor JSON global. A correção foi injetar o MessageConverter na factory e configurá-la explicitamente com factory.setMessageConverter(...).

3. Se algo não funcionou, descreva o que falhou e como você diagnosticou.

O erro mais complexo de diagnosticar foi o MessageConversionException no consumidor.

O que falhou: O consumidor não conseguia desserializar a mensagem que chegava da fila, mesmo a mensagem sendo um JSON válido.

Como diagnostiquei: A chave foi a análise cuidadosa do stack trace. O erro indicava claramente que a falha estava na conversão (Cannot convert from [B] to [OrderCreatedMessage]). Ao revisar o código, percebi que o erro só começou a acontecer depois de introduzir a rabbitListenerContainerFactory para o ack manual. Isso me levou à hipótese de que essa configuração customizada estava isolada do resto da autoconfiguração do Spring. Comparando a RabbitTemplate (que funcionava) com o RabbitListener (que falhava), a única diferença era

essa factory. A solução foi tornar a configuração explícita, passando o conversor JSON para a factory, o que confirmou o diagnóstico e resolveu o problema.