

Documentação dos Testes - API Order

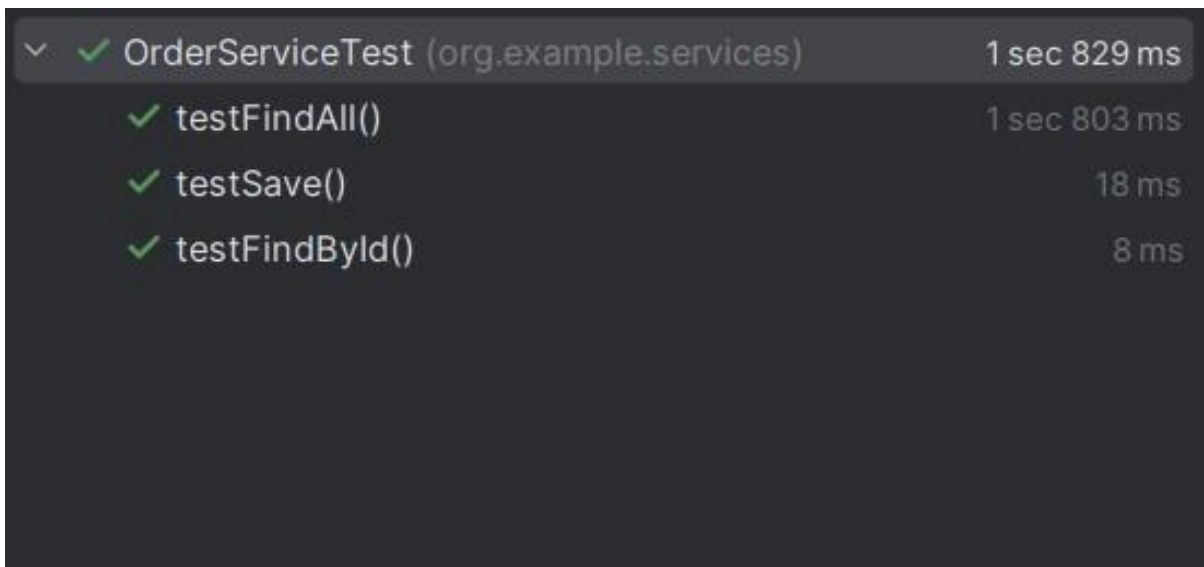
Este documento apresenta a documentação dos testes realizados para a API de pedidos (Order). Foram desenvolvidos testes unitários tanto para o serviço (Service) quanto para o controlador (Controller). Esses testes foram rodados tanto no IntelliJ, por testes unitários, quanto no Jenkins, por pipeline.

Testes de Serviço (OrderServiceTest)

Os testes de serviço têm como objetivo validar a lógica de negócios da aplicação, sem envolver a camada web. Eles verificam as operações principais da classe OrderService, como salvar, buscar por ID e buscar todos os pedidos.

Testes implementados:

- testSave(): Testa o salvamento de um pedido.
- testFindAll(): Testa a busca de todos os pedidos.
- testFindByld(): Testa a busca de um pedido específico pelo ID.

A screenshot of the IntelliJ IDE showing the results of unit tests for the OrderServiceTest class. The interface is dark-themed. At the top, a summary row shows 'OrderServiceTest (org.example.services)' with a green checkmark icon, a total execution time of '1 sec 829 ms', and a dropdown arrow on the left. Below this, three individual test methods are listed, each with a green checkmark icon and its execution time: 'testFindAll()' (1 sec 803 ms), 'testSave()' (18 ms), and 'testFindByld()' (8 ms).

✓ OrderServiceTest (org.example.services)	1 sec 829 ms
✓ testFindAll()	1 sec 803 ms
✓ testSave()	18 ms
✓ testFindByld()	8 ms

(Todos esses testes podem ser encontrados no link ao github anexado a atividade)

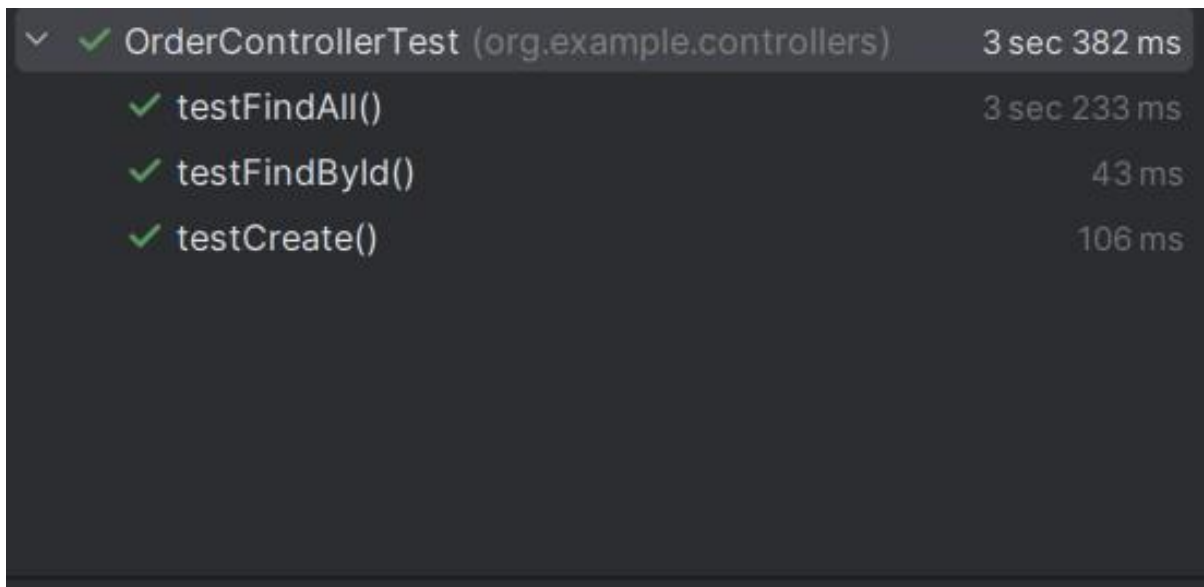
Testes de Controller (OrderControllerTest)

Os testes de controller verificam se os endpoints da API estão funcionando corretamente,

simulando requisições HTTP. Eles testam se a comunicação entre o frontend (ou clientes externos) e a API retorna os resultados esperados.

Testes implementados:

- testCreate(): Testa a criação de um pedido via POST.
- testFindAll(): Testa a busca de todos os pedidos via GET.
- testFindById(): Testa a busca de um pedido específico via GET com ID.



Jenkins

Os testes feitos no Jenkins, tem como base o que foi escrito no JankisFile do nosso projeto Java, que roda todos os testes previamente mencionados, além de testar se o caminho até os testes existe, e reconfirma se todos os testes foram finalizados com sucesso.

Ele tem por objetivo testar todos os códigos de maneira mais dinâmica e completa.



Conclusão

Os testes apresentados garantem que tanto a camada de serviço quanto a de controller da API estão funcionando corretamente para as operações essenciais. A cobertura de testes proporciona maior segurança para manutenção e evolução do sistema.

E os testes feitos pelo Jenkins deixam eles mais fáceis de serem rodados, cria um log visualmente mais fácil de ser interpretado e, se dada a opção, conseguem se utilizar do resultado de um teste para o começo de outro teste, por pipeline.

Alunos

Conrado Perini Fracacio: 04722073

Davi Araújo Gonçalves: 04722033