TestsCantor - Unitário

Cantorcontrollertests/Cantorentitytests/Cantorservicetests

ACFA | JOCOCO | 02/04/2025

JaCoCo 100%

o que o espera para atingir uma cobertura próxima de **100%**:

- **Testar todos os métodos públicos** das classes de serviço e controller.
- **Cobrir cenários positivos e negativos** (sucesso e falha).
- **Incluir testes para exceções e fluxos alternativos**.
- **Testar todas as branches de decisões condicionais** (if, else, switch).
- **Garantir que todos os construtores sejam chamados nos testes**.
- **Testar DTOs, models e métodos utilitários**, mesmo que só tenham getters e setters.
- **Mockar dependências externas** corretamente para evitar falhas inesperadas.
- **Evitar código não testável**, como lógica complexa em blocos static.
- **Executar testes automatizados com frequência** para acompanhar a cobertura.

importantes:

✅ **Testar exceções esperadas**:

- Quando um método lança uma exceção, crie um teste que confirme esse comportamento.
- Use assertThrows() no JUnit para validar a exceção correta.

✅ **Cobrir fluxos de erro**:

- Simule cenários que levem a exceções, como entradas inválidas ou falhas no banco de dados.
- Teste o catch em blocos try-catch para garantir que o código de tratamento seja executado.

✅ **Mocks e exceções**:

- Configure Mockito.when().thenThrow() para simular falhas em serviços dependentes.