

TRABALHO PRÁTICO

Sistema PDV

Apresentando sistema:

- ERP – Sistema integrado de gestão empresarial
- Spring Boot
- Principais funcionalidades
 - Cadastro produtos/clientes/fornecedor
 - Controle de estoque



Despesas

Q

Novo

#	Fornecedor	Observação	Valor Total	Valor Pago	Restante	Vencimento
1	Fornecedor Padrão	Despesa com fornecedor	R\$ 100,00	R\$ 0,00	R\$ 100,00	04/07/2024

Anterior 1/1 Proxima

Relatório de inspeção

- Ferramenta utilizada: Sonarlint
- Classe escolhida: CaixaLancamentoService
- e CaixaService
- Principais Desafios: Presença de um super metodo
- Solução: Foi necessário a quebra em métodos menores

J AjusteProdutoService.java	18	@Service
J AjusteService.java	19	public class CaixaLancamen
J CaixaLancamentoService.java	20	
J CaixaService.java	21	@Autowired
J CategoriaService.java	22	private CaixaLancament
J CfpService.java	23	
J CidadeService.java	24	@Autowired
J CstCsosnService.java	25	UsuarioService usuario
J CstService.java	26	
J EmpresaParametrosService.java	27	private Timestamp data
J EmpresaService.java	28	
J EnderecoService.java	29	public CaixaLancamento
J EstadoService.java	30	}
J FornecedorService.java	31	
	32	@Transactional(readOnl
	33	public String lancamen
	34	detalhesAtual

> notafiscal	
J AjusteProdutoService.java	
J AjusteService.java	
J CaixaLancamentoService.java	M
J CaixaService.java	M
J CategoriaService.java	
J CfpService.java	
J CidadeService.java	
J CstCsosnService.java	
J CstService.java	
J EmpresaParametrosService.java	
J EmpresaService.java	
J EnderecoService.java	
J EstadoService.java	
J FornecedorService.java	

Teste unitários

- Ferramenta utilizada: Junit5 e Mockito
- Classe escolhida: CaixaLancamentoService e CaixaService

```
@Test
void testLancamento_semCaixaAberto_lancaExcecao() {
    CaixaLancamento lancamento = new CaixaLancamento();
    lancamento.setCaixa(null);

    RuntimeException thrown = assertThrows(
        RuntimeException.class,
        () -> caixaLancamentoService.lancamento(lancamento),
        "Esperava lançar RuntimeException"
    );

    assertTrue(thrown.getMessage().contains("Nenhum caixa aberto"));
}

@Test
void testLancamento_saldoInsuficiente_lancaExcecao() {
    Caixa caixa = new Caixa();
    caixa.setValor_total(100.0);

    CaixaLancamento lancamento = new CaixaLancamento();
    lancamento.setCaixa(caixa);
    lancamento.setEstilo(EstiloLancamento.SAIDA);
    lancamento.setValor(200.0);

    String resultado = caixaLancamentoService.lancamento(lancamento);

    assertEquals("Saldo insuficiente para realizar esta operação", resultado);
}
```

Coverage Summary for Class: CaixaLancamentoService (net.originmobi.pdv.service)			
Class	Class, %	Method, %	Line, %
CaixaLancamentoService	100% (1/1)	100% (7/7)	82,8% (24/29)

Current scope: [all classes](#) | [net.originmobi.pdv.service](#)

Class	Class, %	Method, %	Line, %
CaixaLancamentoService	100% (1/1)	100% (7/7)	82,8% (24/29)

```
1 package net.originmobi.pdv.service;
2
3 import java.sql.Timestamp;
4 import java.util.List;
5 import java.util.Optional;
6
7 import org.springframework.stereotype.Service;
8 import org.springframework.transaction.annotation.Propagation;
9 import org.springframework.transaction.annotation.Transactional;
10
11 import net.originmobi.pdv.enumerado.caixa.EstiloLancamento;
12 import net.originmobi.pdv.enumerado.caixa.TipoLancamento;
13 import net.originmobi.pdv.model.Caixa;
14 import net.originmobi.pdv.model.CaixaLancamento;
15 import net.originmobi.pdv.repository.CaixaLancamentoRepository;
16
17 @Service
18 public class CaixaLancamentoService {
19
20     private final CaixaLancamentoRepository caixaLancamento;
21     final UsuarioService usuarios;
22     private Timestamp dataHoraAtual;
23
24     public CaixaLancamentoService(CaixaLancamentoRepository caixaLancamento, UsuarioService usuarios) {
25         this.caixaLancamento = caixaLancamento;
26         this.usuarios = usuarios;
27     }
28
29     @Transactional(readOnly = false, propagation = Propagation.REQUIRED)
30     public String lancamento(CaixaLancamento lancamento) {
31         dataHoraAtual = new Timestamp(System.currentTimeMillis());
32         lancamento.setData_cadastro(dataHoraAtual);
33
34         if (!isCaixaValido(lancamento)) {
35             throw new RuntimeException("Nenhum caixa aberto");
36         }
37
38         if (isSaidaComSaldoInsuficiente(lancamento)) {
39             return "Saldo insuficiente para realizar esta operação";
40         }
41
42         ajustarValorParaSaida(lancamento);
43         ajustarObservacao(lancamento);
44
45         try {
46             caixaLancamento.save(lancamento);
47         } catch (Exception e) {
48             throw new RuntimeException("Erro ao realizar lançamento, chame o suporte", e);
49         }
50
51         return "Lançamento realizado com sucesso";
52     }
53
54     private boolean isCaixaValido(CaixaLancamento lancamento) {
55         return lancamento.getCaixa().isPresent() && !lancamento.getCaixa().map(Caixa::getData_fechamento).isPresent();
56     }
57
58     private boolean isSaidaComSaldoInsuficiente(CaixaLancamento lancamento) {
59         if (lancamento.getEstilo().equals(EstiloLancamento.SAIDA)) {
60             Optional<Double> vlTotalCaixa = lancamento.getCaixa().map(Caixa::getValor_total);
61             return lancamento.getValor() > vlTotalCaixa.orElse(0.0);
62         }
63     }
```

Teste de sistema

- Ferramenta utilizada: Selenium e Pytest
- Funcionalidades testadas:
Login – Verificar Autenticidade
Cadastro produto – Requisitos funcionais

```
✓ def test_cadastro_produto_vazio(driver):
    driver.get("http://localhost:8080/login")
    time.sleep(2)
    username_input = driver.find_element(By.ID, "user")
    password_input = driver.find_element(By.ID, "password")
    username_input.send_keys("gerente")
    password_input.send_keys("123")
    time.sleep(1)
    login_button = driver.find_element(By.ID, "btn-login")
    login_button.click()
    time.sleep(2)
    driver.get("http://localhost:8080/produto/form")
    time.sleep(2)
    login_button = driver.find_element(By.XPATH, '//*[@id="form_produto"]/input[2]')
    login_button.click()
    time.sleep(2)
    error_message = driver.find_element(By.XPATH, ' // *[@ id = "descricao-error"]').text
    assert error_message == "Este campo é requerido."

✓ def test_cadastro_sem_preco(driver):
    driver.get("http://localhost:8080/login")
    time.sleep(2)
    username_input = driver.find_element(By.ID, "user")
    password_input = driver.find_element(By.ID, "password")
    username_input.send_keys("gerente")
    password_input.send_keys("123")
    time.sleep(1)
    login_button = driver.find_element(By.ID, "btn-login")
    login_button.click()
    time.sleep(2)
    driver.get("http://localhost:8080/produto/form")
    time.sleep(2)
    login_button = driver.find_element(By.XPATH, '//*[@id="form_produto"]/input[2]')
    login_button.click()
    time.sleep(2)
    name_input = driver.find_element(By.ID, "descricao")
    name_input.send_keys("Teste2")
    login_button = driver.find_element(By.XPATH, '//*[@id="form_produto"]/input[2]')
    login_button.click()
    error_message = driver.find_element(By.ID, 'valorVenda-error').text
    assert error_message == "Este campo é requerido."
```

Buscar produtos

Q

Produto

Novo

Código	Descrição	Valor Venda	Grupo	Categoria	Estoque	
1	Picolé	R\$ 6,50	Padrão	Padrão	0	
2	Sorvete Kg	R\$ 0,00	Padrão	Padrão	0	
3	Teste1	R\$ 0,01	Padrão	Padrão	0	
4	teste89	R\$ 0,09	Padrão	Padrão	0	
5	Teste2	R\$ 7,50	Padrão	Padrão	0	
6	teste891	R\$ 0,01	Padrão	Padrão	0	
7	Teste3	R\$ 7,50	Padrão	Padrão	0	
8	Teste4	R\$ 7,50	Padrão	Padrão	0	
9	Teste4	R\$ 7,50	Padrão	Padrão	0	
10	Teste5	R\$ 7,50	Padrão	Padrão	0	
11	testeqwqw	R\$ -10,00	Padrão	Padrão	0	
12	Teste5	R\$ -10,00	Padrão	Padrão	0	

Anterior

1/1

Proxima

Valor da balança?

NAO

Valor Custo

5,00

Valor Venda

7,50

Data Validade

01/01/2020

Quantidade Estoque

0

Atributos de Qualidade – ISO/IEC 25010

Adequação Funcional

Nota Ideal: 5

O sistema deve garantir que todas as funcionalidades sejam de fato funcionais e de forma apropriada conforme o previsto.

Eficiência

Nota Ideal: 3.5

O sistema deve ter uma resposta rápida as operações críticas, porém, não deverá ter uma capacidade de usuários simultaneos enorme.

Compatibilidade

Nota Ideal: 3.5

O sistema deve ter uma compatibilidade com outros sistemas de pagamentos e financeiros.
O sistema deve operar bem em ambientes onde outros aplicativos estão em uso, sem causar conflitos significativos.

Usabilidade

Nota Ideal: 4

O sistema deve garantir ter uma interface intuitiva de facil aprendizado e com bastante proteção a erros de usuários.

Atributos de Qualidade – ISO/IEC 25010

Confiabilidade

Nota Ideal: 4

O sistema deve ter pouquíssimas falhas e uma alta disponibilidade para não prejudicar o funcionamento do estabelecimento.

Segurança

Nota Ideal: 4

O sistema deve ter uma forte autenticidade, não permitir que usuários desconhecidos logarem. Também possuir não deve possuir muitas vulnerabilidades.

Manutenibilidade e

Nota Ideal: 4

O sistema deve ser dividido em bastantes módulos e uma alta testabilidade por se tratar de um sistema com operações financeiras.

Portabilidade

Nota Ideal: 4

O sistema deve ter uma alta adaptatividade ao diversos sistemas operacionais e uma fácil e rápida instalação .