

# Caminho Feliz

Projeto: PDV Integrantes: Clara Lino; Laryssa Paiva; Leonardo Calvet; Wesley Ferreira



#### Sumário

- 1. Escopo do sistema
- 2. Testes Unitários
- 3. Testes de Integração
- 4. Atributos de Qualidade ISO 25010
- 5. Testes de Sistemas usando requisitos Funcionais
- 6. Melhorando Conjunto de Casos de Teste

#### Escopo do Sistema

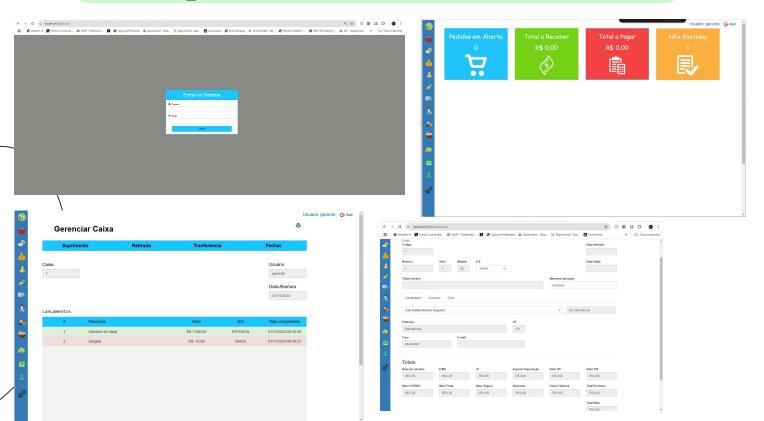
#### pdv

Sistema de ERP web desenvolvido em Java com Spring Framework

#### Recursos

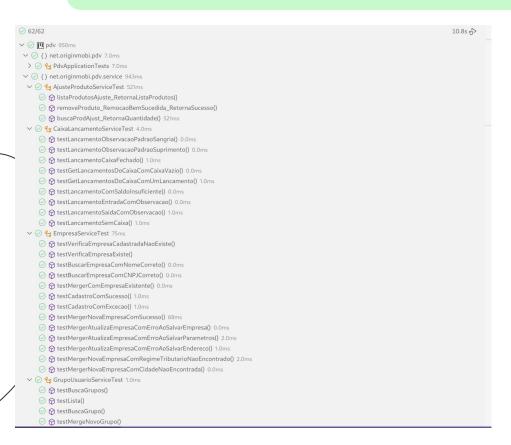
- Cadastro produtos/clientes/fornecedor
- Controle de estoque
- Gerenciar Pagamentos de Despesas
- Adicionar empresas \*
- Controle de fluxo de caixa (Gerenciar Caixa) \*
- Controle de pagar despesas \*
- Gerenciar permissões de usuários por grupos (Gerenciar grupo de usuários) \*
- Cadastrar novas formas de pagamentos
- Relatórios

# Escopo do Sistema





#### **Teste Unitários**



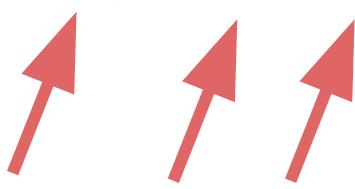
#### **62 Testes Unitários**

- Melhoramos e aumentamos os casos de testes e aumentamos
- Isolamos com Mocks e Stubs

# **Teste Unitários**

#### net.originmobi.pdv.service

Element	Missed Instructions	Cov. \$	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes =
EmpresaService		89%		100%	0	6	9	53	0	4	0	1
PagarService		98%		100%	1	9	1	55	1	4	0	1
GrupoUsuarioService		100%	=	100%	0	11	0	36	0	8	0	1
CaixaLancamentoService		87%		77%	5	14	4	33	0	3	0	1
UsuarioService		88%		66%	4	12	8	47	0	6	0	1
<u>AjusteProdutoService</u>		26%		10%	6	10	23	31	1	5	0	1



#### **Teste Unitários**

#### net.originmobi.pdv.service

Element	Missed Instructions	Cov. \$	Missed Branches		Missed	Cxty	Missed	Lines 🛊	Missed # Me	ethods =	Missed	Classes
EmpresaService		89%	=	100%	0	6	9	53	0	4	0	1
PagarService		98%		100%	1	9	1	55	1	4	0	1
GrupoUsuarioService		100%	=	100%	0	11	0	36	0	8	0	1
CaixaLancamentoService		87%	_	77%	5	14	4	33	0	3	0	1
UsuarioService		88%		66%	4	12	8	47	0	6	0	1
AjusteProdutoService		26%		10%	6	10	23	31	1	5	0	1

#### Pontos Positivos 🔽

- **GrupoUsuarioService** possui **100% de cobertura** de instruções e ramificações.
- PagarService apresenta 98% de cobertura de instruções e 100% de cobertura de ramificações;
   EmpresaService e UsuarioService têm cobertura acima de 85%.

#### Pontos Negativos X

- AjusteProdutoService tem apenas 26% de cobertura de instruções e 10% de cobertura de ramificações;
- AjusteProdutoService também possui 23 linhas e 1 método não testado;

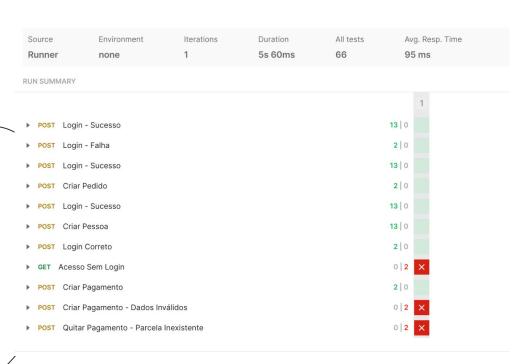
O sistema apresenta baixa testabilidade, conforme os critérios da ISO/IEC 25010, devido à deficiências na injeção de dependências, alto acoplamento entre componentes e falta de modularidade.





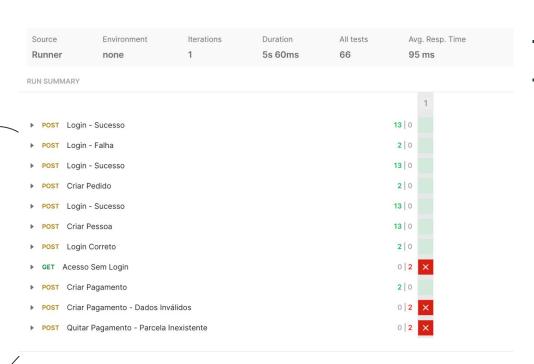
#### <u>66 Testes de</u> <u>Integração</u>

- Utilizamos o Postman.
- Aspecto a ser testado é a integração entre os componentes, os recursos eram adicionados no banco de dados a informação
- Não foi o foco dos testes do projeto, foi focado mais em testes de sistema.



#### <u>66 Testes de</u> <u>Integração</u>

- Utilizamos o Postman.
- Aspecto a ser testado é a integração entre os componentes, os recursos eram adicionados no banco de dados a informação
- Não foi o foco dos testes do projeto, foi focado mais em testes de sistema.



#### <u>66 Testes de</u> <u>Integração</u>

- Utilizamos o Postman.
- Aspecto a ser testado é a integração entre os componentes, os recursos eram adicionados no banco de dados a informação
- Não foi o foco dos testes do projeto, foi focado mais em testes de sistema.

```
POST
                 http://localhost:8080/login
                                              Scripts • Settings
                        pm.test("Status code is 200", function () {
Pre-request
                            pm.response.to.have.status(200);
Post-response •
                        pm.test("Response contains 'Usuário: gerente'", function () {
                            pm.expect(pm.response.text()).to.include("Usuário: gerente");
                        3);
                        pm.test("Response contains 'Pedidos'", function () {
                            pm.expect(pm.response.text()).to.include("Pedidos");
                        pm.test("Response contains 'Total a Receber'", function () {
                            pm.expect(pm.response.text()).to.include("Total a Receber");
                        pm.test("Response contains 'NFe Emitidas'", function () {
                            pm.expect(pm.response.text()).to.include("NFe Emitidas");
                        pm.test("Response contains 'Total a Pagar'", function () {
                            pm.expect(pm.response.text()).to.include("Total a Pagar");
                        3);
                        pm.test("Response contains 'Produtos'", function () {
                            pm.expect(pm.response.text()).to.include("Produtos");
                        pm.test("Response contains 'Usuários'", function () {
```

#### <u>66 Testes de</u> <u>Integração</u>

(%)

- Utilizamos o Postman.
- Aspecto a ser testado é a integração entre os componentes, os recursos eram adicionados no banco de dados a informação
- Não foi o foco dos testes do projeto, foi focado mais em testes de sistema.



# Teste Mutação

# 1726 Mutantes encontrados

Pitest Executado no MODE default

# **Teste Mutação**

# mutante

- Precisava de mais testes verificando cada aresta.
- Se tornou um pouco massante, a quantidade que havia.
- Pitest foi executado no modo DEFAULT





84 Mutantes

# 4. Atributos de Qualidade ISO 25010

## Ranking Qualidade PDV

#### 1. Adequação funcional

 Os requisitos propostos ao sistema devem funcionar adequadamente de acordo com o seu objetivo, atendendo a necessidade do cliente.

#### 2. Confiabilidade

- a. Pela necessidade de se manter disponível;
- Pela importância da recuperabilidade para recuperar os dados do PDV

#### 3. Segurança

- a. Pela importância da integridade das informações que circulam no PDV
- b. É importante que os dados dos usuários se mantenham confidenciais

#### 4. Usabilidade

a. A importância se dá pela operabilidade do sistema

#### Ranking Qualidade PDV

- 5. Eficiência de desempenho
  - a. Necessário o pdv estar com um bom desempenho para garantir que tenha uma alta rotatividade de venda
- 6. Manutenibilidade
  - b. Garantir que possa ser implementadas novas funcionalidades para manter o sistema atualizado
- 7. Compatibilidade
  - c. Pelo software ser executado apenas em uma máquina, a questão da coexistência não é de todo valor
- 8. Portabilidade
  - d. As máquinas que irão executar o software serão exclusivas do mesmo



#### Testes de Sistema

- 14 testes de sistema
- Funcionalidades básicas do PDV
- Selenium em Python
- Muito CRUD

#### **Testes**

- 1. Abrir 2 bancos pelo Caixa / Cofre
- 2. Abrir um pedido e confirmar
- 3. Criar Máquina cartão e Título
- 4. Fechar Pedido
- 5. Editar Estoque
- 6. Criar pessoa, confirmar que ela esta na lista
- 7. Editar pessoa
- 8. Criar fornecedor
- 9. Editar fornecedor
- 10. Lançar despesa, confirma que a despesa foi lançada
- 11. Criar grupo e categoria, criar produto desse grupo e categoria
- 12. Adicionar forma de pagamento
- 13. verifica se soma das despesas na página inicial bate com as despesas lançadas
- 14. Criar empresa e gerar nota fiscal dela

#### **Exemplos**

Adicionar forma de pagamento

```
time.sleep(5)
    campo listarTodos = driver.find element(By.XPATH, '//*[@id="btn-padrao"]/a').click()
    try:
        WebDriverWait(driver, 10).until(
           EC.visibility of any elements located((By.XPATH, f'//table/tbody/tr/td[contains(text(), "{descricao}")]'))
        assert driver.find_element(By.XPATH, f'//table/tbody/tr/td[contains(text(), "{descricao}")]')
        print(f"Pagamento '{descricao}' adicionado com sucesso à lista de pagamentos.")
    except TimeoutException:
        print(f"Pagamento '{descricao}' não apareceu na lista de pagamentos a tempo.")
        raise AssertionError(f"Pagamento '{descricao}' não encontrado na lista de pagamentos.")
try:
```

#### **Exemplos**

Editar pessoa

```
if alert.text == "Pessoa salva com sucesso":
    print(alert.text)
    alert.accept()
elif alert.text == "Já existe uma pessoa cadastrada com este CPF/CNPJ, verifique":
    print("Erro: CPF duplicado.")
    alert.accept()
    raise ValueError("CPF já cadastrado. Verifique os dados.")

except UnexpectedAlertPresentException as e:
    alert = driver.switch_to.alert
    print(f"Alerta inesperado: {alert.text}")
    alert.accept()
    raise e
```

#### Exemplo de erro no sistema

- Estoque não atualiza
- Não é possível fechar pedido
  - Mesmo com a criação de outros produtos

```
assert valor_elemento == "10", f"Valor encontrado ({valor_elemento}) não é igual a 10"

AssertionError: Valor encontrado (0) não é igual a 10

(veny) larvssa@NAPTOP-3PU3J9UB:~/trabalho-qt/uff-qualidade-teste-ndv/entrega@2/teste de sistema
```

```
File "/home/laryssa/trabalho-qt/uff-qualidade-teste-pdv/entrega02/teste_de_sistema/casos_de_tes
assert texto_alerta == "Pedido Fechado!", f"Texto do alerta é '{texto_alerta}'."
AssertionError: Texto do alerta é 'O produto de código 3 não tem estoque suficiente, verifique'.
```

# 6. Melhorando Conjunto de Casos de Teste

# Relatório de inspeção do código-fonte

ter matched 268 of 1645 items			
esource	Date	Desc	ription
		0	Blocker Impact (4 items)
application.properties	6 years ago	0	Make sure this database password gets changed and removed from the code.
PagarController.java	6 years ago	0	Add a call to "setComplete()" on the SessionStatus object in a "@RequestMapping" method
ProdutoController.java	6 years ago	•	Add a call to "setComplete()" on the SessionStatus object in a "@RequestMapping" method
ReceberController.java	6 years ago	•	Add a call to "setComplete()" on the SessionStatus object in a "@RequestMapping" method
		0	
ImagemProdutoService.java	6 years ago	0	Use try-with-resources or close this "FileOutputStream" in a "finally" clause.
AssinaXML.java	6 years ago	0	Use try-with-resources or close this "FileInputStream" in a "finally" clause.
ImagemProdutoService.java	6 years ago	0	Save and re-use this "Random".
ConexaoJDBC.java	6 years ago	0	Revoke and change this password, as it is compromised.
Categoria Controller. java	6 years ago	0	Replace this persistent entity with a simple POJO or DTO object.
Fornecedor Controller. java	6 years ago	0	Replace this persistent entity with a simple POJO or DTO object.
Fornecedor Controller. java	6 years ago	0	Replace this persistent entity with a simple POJO or DTO object.
FornecedorController.java	6 years ago	0	Replace this persistent entity with a simple POJO or DTO object.
GrupoController.java	6 years ago	0	Replace this persistent entity with a simple POJO or DTO object.
GrupoUsuarioController.java	6 years ago	0	Replace this persistent entity with a simple POJO or DTO object.
MaquinaCartaoController.java	6 years ago	0	Replace this persistent entity with a simple POJO or DTO object.
PagamentoTipoController.java	6 years ago	0	Replace this persistent entity with a simple POJO or DTO object.
ProdutoController.java	6 years ago	0	Replace this persistent entity with a simple POJO or DTO object.
TituloController.java	6 years ago	0	Replace this persistent entity with a simple POJO or DTO object.
TributacaoController.java	6 years ago	0	Replace this persistent entity with a simple POJO or DTO object.
UsuarioController.java	6 years ago	0	Replace this persistent entity with a simple POJO or DTO object.
VendaController.java	6 years ago	0	Replace this persistent entity with a simple POJO or DTO object.
AssinaXML.java	6 years ago	0	Replace this call to "replaceAll()" by a call to the "replace()" method.
AssinaXML.java	6 years ago	0	Replace this call to "replaceAll()" by a call to the "replace()" method.
CaixaController.java	6 years ago	0	Replace this call to "replaceAll()" by a call to the "replace()" method.
ConversorXmlNfe.java	6 years ago	0	Replace this call to "replaceAll()" by a call to the "replace()" method.
ConversorXmlNfe.java	6 years ago	0	Replace this call to "replaceAll()" by a call to the "replace()" method.

A análise do projeto com a ferramenta SonarQube revelou mais de 1.500 pontos de falha no código. Dentre eles, 70 foram classificados como de alto impacto (High Impact) ou críticos (Blocker Impact).

# Melhorando Conjunto de Casos

de Teste

Após a implementação de melhorias específicas nessas áreas, conseguimos reduzir em quase 40% a quantidade de falhas classificadas como críticas ou de alto impacto.

