

# Projeto PDV

QUALIDADE E TESTE - TCC00365

---

## GRUPO:

- GUILHERME VIEIRA
- ISADORA DUARTE
- IZABEL SOARES
- PATRICIA CARVALHO

# Sobre o Projeto

---

O projeto escolhido foi de um sistema de **ERP web** desenvolvido em Java com a utilização do Spring Framework.

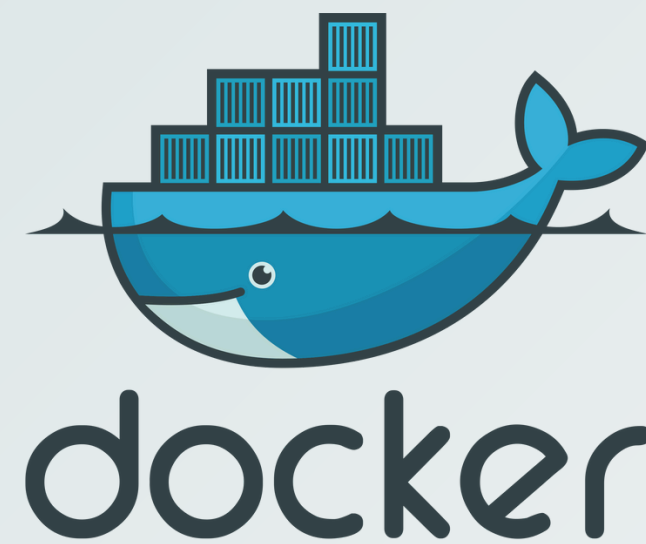
Algumas de suas funcionalidades incluem:

- **Cadastro de produtos, clientes e fornecedores;**
- **Controle de estoque com monitoramento de entradas e saídas;**
- **Gerenciamento de comandas e vendas para acompanhar pedidos e transações;**
- **Controle financeiro, incluindo fluxo de caixa e contas a pagar/receber;**
- **Integração com meios de pagamento, como cartões e cadastro de novas formas;**
- **Geração de relatórios para análise de desempenho e tomada de decisões;**
- **Gestão de permissões de usuários com controle de acesso por grupos.**

# Ferramentas

---

JUnit



# Escopo

---

Dentro do escopo da primeira entrega do projeto PDV estão os testes manuais e testes unitários realizados pelos componentes do grupo. Os testes unitários realizados foram das seguintes classes: **AjusteService**, **CaixaService**, **NotaFiscalService** e **VendaService**.

Fora do escopo do projeto PDV estão os testes envolvendo interfaces de usuário, interfaces de hardware e de software, bem como a lógica de banco de dados. Além disso, também não estão incluídas as interfaces de comunicação do sistema.

# Metodologia de Teste

---

**Testes Unitários:** realizados pelos desenvolvedores para validar os componentes individuais do sistema. Isolam serviços com mocks de repositórios e colaborações.

**Testes Manuais:** realizados pelos testadores para identificar falhas de usabilidade e reproduzir cenários reais de uso.

## **Critérios de Suspensão e Requisitos de Retomada:**

- Se os membros da equipe relatarem que há 30% dos casos de teste com falha, suspenda o teste até que a equipe de desenvolvimento corrija todos os casos com falha.

## **Completude do Teste:**

- 100% dos casos de teste manuais e unitários planejados executados;
- Todos os erros abertos foram corrigidos ou serão na próxima versão;
- Defeitos críticos e de alta prioridade foram corrigidos e retestados.



# Metodologia de Teste

Marco	Atividade
Até 17/09	Escolha do projeto, clonagem do repositório
Até 26/09	Seleção do escopo
Até 03/10	Primeira versão dos testes unitários
Até 03/10	Testes manuais
Até 06/10	Finalização do plano de testes e da primeira apresentação de slides, atualização do README
Até 08/10	Primeira apresentação

# Testes Unitários

## NotaFiscalService

- retornaListaDeNotasFiscais()
- retornaTotalDeNotasEmitidas()
- buscaNotaFiscalPorCodigo()
- geraDVCorretamente()
- salvaArquivoXML()
- removeArquivoXMLExistente()
- cadastroDeNotaFiscal()

## AjusteService

- testeAjusteStatusAPROCESSAR()
- testeBuscaAjustePorCodigo()
- testeSeNaoRemoveAjusteProcessado()
- testeSeRemoveAjusteProcessado()
- testeProcessaAjusteComEntradaProduto()
- testeListaComFiltroCodigo()

# Testes Unitários

## VendaService

- testAbreVenda\_SemCodigo\_PreencheCamposBuscaUsuarioESalva
- testAbreVenda\_Existecodigo\_AtualizaDados
- testBusca\_FilterTemCodigo\_ChamaFindByCodigoIn
- testBusca\_FilterSemCodigo\_ChamaFindBySituacao
- testBusca\_SituacaoFechada\_ChamaFindBySituacaoFechada
- testAddProduto\_VendaAberta\_SalvaERetornaOk
- testAddProduto\_VendaFechada\_RetornaMensagem
- testRemoveProduto\_VendaAberta\_RemoveERetornaOk
- testRemoveProduto\_VendaFechada\_RetornaMensagem
- testLista\_RetornaVendas
- testQtdAbertos\_RetornaTotalDeVendasEmAberto

- testFechaVenda\_fluxoVistaDinheiro\_FechaVenda
- testFechaVenda\_VendaFechada\_RetornaMensagem
- testFechaVenda\_ValorProdutoInvalido\_RetornaMensagem
- testFechaVenda\_CaixaFechadoAVistaDinheiro\_RetornaMensagem
- testFechaVenda\_AprazoSemCliente\_RetornaMensagem
- testFechaVenda\_Aprazo\_GeraParcelasEFecha
- testFechaVenda\_Cartao\_ChamaCartaoLancamento
- testFechaVenda\_FalhaAoCadastrarReceber\_RetornaErro
- testFechaVenda\_FalhaAoFechar\_RetornaErro
- testAprazo\_ParcelaVazia\_RetornaMensagem
- testAvistaDinheiro\_SomaParcelasDiferenteTotal\_RetornaMensagem
- testAvistaDinheiro\_ParcelaVazia\_RetornaMensagem



# Testes Unitários

## CaixaService

- cadastro\_quandoCaixaJaAberto\_RetornaErro()
- cadastro\_quandoValorAberturaNegativo\_RetornaErro()
- cadastro\_BancoSemSaldoInicial\_NormalizaESalvaSemLancamento()
- cadastro\_SaldoInicialPositivo\_LancaSaldoInicial()
- cadastro\_FalhaAoSalvar\_RetornaMensagem()
- cadastro\_FalhaNoLancamentoInicial\_RetornaMensagem()
- fechaCaixa\_SenhaVazia\_RetornaMensagem()
- fechaCaixa\_SenhaCorreta\_FechaESalva()
- fechaCaixa\_JaFechado\_RetornaErro()
- fechaCaixa\_SenhaIncorreta\_RetornaMensagem()
- fechaCaixa\_FalhaAoSalvar\_RetornaMensagem()
- caixaIsAberto\_HaCaixaAberto\_retornaTrue()
- caixaIsAberto\_NaoHaCaixaAberto\_retornaFalse()

- listaTodos\_retornaListaOrdenadaPorCodigo()
- caixaAberto\_retornaOptionalDoCaixa()
- caixasAbertos\_retornaListaDeCaixasEmAberto()
- busca\_CodigoInformado\_retornaOptional()
- listaBancos\_retornaCaixasTipoBanco()
- listaCaixasAbertosTipo\_retornaCaixasDoTipoInformado()
- listarCaixas\_FiltroTemData\_buscaPorData()
- listarCaixas\_FiltroSemData\_listaAbertos()
- buscaCaixaUsuario\_Existente\_RetornaOptional()
- buscaCaixaUsuario\_NaoExistente\_RetornaVazio()
- listaBancosAbertosTipoFilterBanco\_TemData\_buscaPorTipoEData()
- listaBancosAbertosTipoFilterBanco\_SemData\_buscaPorBanco()

# Testes Manuais

## Abrir pedido e inserir produto

- Tester: Guilherme Vieira
- Teste aprovado

## Alterar a quantidade de produtos no estoque

- Tester: Isadora Duarte
- Teste aprovado
















## Abrir e fechar caixa

- Tester: Izabel Soares
- Teste aprovado

## Cadastrar pessoa (Testlink)

- Tester: Patrícia Carvalho
- Teste aprovado

# Teste Manual – Testlink

PDV25.2-1 : Cadastrar nova pessoa - Versão1  				
<b>Objetivo do Teste:</b>				
Verificar a história de usuário				
<b>Pré-condições</b>				
Versões do software: Sistema PDV 1.0.0				
Configuração necessária: é necessário criar o banco de dados denominado "pdv" no MySQL, configurar o arquivo application.properties com as credenciais do usuário root, e em seguida executar o projeto diretamente pelo Eclipse ou gerar o arquivo .jar para execução.				
Alternativamente, a aplicação também pode ser executada por meio de um container Docker.				
 #	Ações do Passo	Resultados Esperados:	Execução	
1	Efetuar login usando as credenciais disponíveis (usuário "gerente" e senha "123")	Possível logar, redirecionamento para a página inicial da aplicação	Manual	 
2	Colocar o cursor sobre a barra na lateral esquerda	Expansão do menu e surgimento das opções	Manual	 
3	Clicar em PESSOAS	Redirecionamento para a página de pessoas	Manual	 
4	Clicar em NOVO	Redirecionamento para a página de cadastramento de pessoa	Manual	 
5	Preencher dados da pessoa, inclusive endereço	Possível digitar	Manual	 
6	Clicar em SALVAR	Possível clicar, aparece popup dizendo PESSOA SALVA COM SUCESSO	Manual	 



Obrigado!

---