



# Caminho Feliz

**Projeto:** PDV

**Integrantes:**

Clara Lino;

Laryssa Paiva;

Leonardo Calvet;

Wesley Ferreira



# **xIntrodução**

# 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

Entrar no Sistema

Entrar

Sair

Entrar

Usuário: gerente

 sair

Pedidos em Aberto

0




Total a Receber

R\$ 0,00



Total a Pagar

R\$ 0,00



NFe Emitidas

1



[illegible]

**Fatura**

Enviar Código    Salvo e Feito    SEI - Sistema de L.i.e    Lógica de Precalculo    Questionário - Ques...    Página total: Ques...    Documentos

Tópicos em Destaque

---

Emitir Código  <input type="text"/>	Data emissão  <input type="text"/>
Número  <input type="text"/>	Data Saída  <input type="text"/>
Série  <input type="text"/>	Natureza operação Empresa  <input type="text"/>
Modelo  <input type="text"/>	
E/S  <input type="text"/> SAÍDA	

---

Chave acesso

Destinatário    Produto    Fvto

jodo Rafael Mendes Nogueira        015.905.822-32

---

Endereço    UF

Betiminas    RO

Fone    E-mail

(054)42067   

---

### Totais

Base de calculo	ICMS	ST	Imposto Importação	Valor IPI	Valor PIS
R\$ 0.00	R\$ 0.00	R\$ 0.00	R\$ 0.00	R\$ 0.00	R\$ 0.00
Valor CFOPIS	Valor Frete	Valor Seguro	Desconto	Outros Valores	Total Produtos
R\$ 0.00	R\$ 0.00	R\$ 0.00	R\$ 0.00	R\$ 0.00	R\$ 0.00
					Total Nota
					R\$ 0.00



# **X Planejamento de Teste**

# Objetivos

Os objetivos do teste são garantir a **funcionalidade e confiabilidade do sistema de ERP web para PDV**, focando nas operações essenciais, como cadastro de pedidos no caixa, gerenciamento de empresas, de usuários e etc. O objetivo é assegurar que todas essas operações propostas no escopo funcionam.

- Identificar e corrigir defeitos
- Aumentar a confiabilidade do sistema
- Garantir a funcionalidade e estabilidade do sistema
- Garantir a qualidade do software

# Objetivos


- Testes Unitarios
- Testes de Integracao
- Testes Manuais do Sistema

# Fases e organização dos testes

## Plano de Teste

Modelo de Plano de Testes por [www.gurup9.com](http://www.gurup9.com)

Plano de Teste  
PDV Caminho Feliz



Registro de Mudanças

Versão	Data de Mudança	Por	Descrição
1.0	07/05/2021	Marcos	Elaboração do documento

Modelo de Plano de Testes por [www.gurup9.com](http://www.gurup9.com)

<b>1</b>	<b>INTRODUÇÃO</b>	<b>2</b>
1.1	Escopo	2
1.1.1	Do escopo	2
1.1.2	Fora do escopo	2
1.2	Objetivos e Escopo	3
1.3	Política e Responsabilidades	3



# Fases e organização dos testes

Nome do Módulo	Papéis aplicáveis	Descrição
Gerenciar Caixa	Gerente	<b>Gerente:</b> O módulo "Gerenciar Caixa" permite que o gerente controle suprimentos, retiradas, lançamentos e saldo do caixa, além de visualizar o usuário e a data de abertura. Ele garante o monitoramento diário do fluxo financeiro.
Gerenciar grupo de usuários	Gerente	<b>Gerente:</b> O módulo "Gerenciar grupo de usuários" permite que o gerente visualize todos os grupos de usuário, edite atributos dos grupos, adicione e exclua permissões do grupo, apague grupos, visualize todos os grupos que estão associados a determinado usuário e relacione usuários com grupos. Garante o gerenciamento dos grupos de usuários.
Adicionar empresas	Gerente	<b>Gerente:</b> O módulo "adicionar empresas" permite que o gerente adicione empresas fornecedoras de produtos
Gerenciar Pagamentos de Despesas	Gerente	<b>Gerente:</b> O módulo de gerenciar pagamentos de despesas permite com que, dado um valor em caixa e um despesa, o gerente consiga, usando o valor em caixa, diminuir das despesas registradas

# **Fases e organização dos testes**

## **Ferramentas para Teste**

- **JUnit (para testes)**
- **Mockito (para mocks)**
- **Jacoco (para cobertura)**

# Fases e organização dos testes

- Definir classes com complexidade razoável e não CRUD
- Criação do plano de testes (Plano de testes PVD)

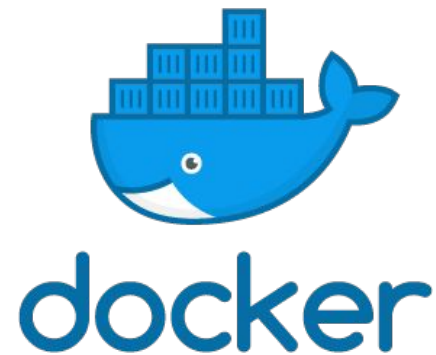
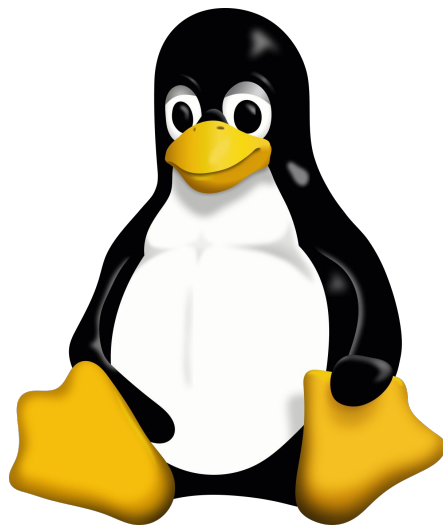
# Metodologia

- Testes Unitários
  - Utilização de **Mocks** para testes unitários
  - Permite testar cada unidade de código sem dependências externas
  - Testes rápidos
- Testes manuais feitos no sistema PVD e apresentados no modelo de documentação de testes manuais fornecido
- Simulação de diversos casos de uso
  - Testes foram considerados etapas dentro do cenário de caso de uso da ferramenta
  - Testes atendem às necessidades reais dos usuários ao utilizar a ferramenta

# Requisitos

- Especifica os critérios que denotam a conclusão bem-sucedida de uma fase de teste
- A **taxa de 100% de execução dos testes é obrigatória**, a menos que um motivo claro seja fornecido.
- A **taxa de 100% de aprovação dos testes é obrigatória**.
- Dos códigos de teste executados, **deve possuir mais de 80% de cobertura**.

# Ambiente



# Resultados

- Testes unitários:
  - CaixaLancamentoService
  - GrupoUsuarioService
  - EmpresaService
  - PagarService
- Testes manuais:
  - Gerenciar caixa
  - Gerencia grupo usuários
  - Gerenciar Empresa
  - Gerenciar Pagamento de Despesas



# **Testes** **xunitários e** **manuais**



# Gerenciar Grupos de usuários

- Testes realizados:
  - Listar todos os grupos de usuários do sistema;
  - Editar atributos do grupo de usuário
  - Adicionar permissões no grupo de usuário
  - Excluir permissões no grupo de usuário
  - Excluir grupo de usuário
  - Adicionar usuário ao grupo
  - Excluir usuário do grupo

# Gerenciar Grupos de usuários

```
[INFO] -----  
[INFO] T E S T S  
[INFO] -----  
[INFO] Running net.originmobi.pdv.service.GrupoUsuarioServiceTest  
[INFO] Tests run: 11, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.277 s - in net.originmobi.pdv.service.GrupoUsuarioServiceTest  
[INFO]  
[INFO] Results:  
[INFO]  
[INFO] Tests run: 11, Failures: 0, Errors: 0, Skipped: 0  
[INFO]  
[INFO]  
[INFO] --- jacoco-maven-plugin:0.8.8:report (report) @ pdv ---  
[INFO] Loading execution data file /app/target/jacoco.exec  
[INFO] Analyzed bundle 'pdv' with 171 classes  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 5.283 s  
[INFO] Finished at: 2024-11-04T17:31:43Z  
[INFO] -----  
root@1ed5bc622dfd:/app#
```

# Gerenciar Grupos de usuários

```
localhost:5500/target/site/jacoco/net.originmobi.pdv.service/GrupoUsuarioService.java.html#L14
Altmetric it! Evento 2 a 6 de dez... STI IdUFF - Sistema de i... Lógica de Predicados Questionário - Desa... Página Inicial - Qua... Documentos
```

```
7- import org.springframework.web.servlet.mvc.support.RedirectAttributes;
8-
9- import net.originmobi.pdv.model.GrupoUsuario;
10- import net.originmobi.pdv.model.Usuario;
11- import net.originmobi.pdv.repository.GrupoUsuarioRepository;
12-
13- @Service
14- public class GrupoUsuarioService {
15-
16-     @Autowired
17-     private GrupoUsuarioRepository grupousuarios;
18-
19-     public List<GrupoUsuario> buscaGrupos(Usuario usuario) {
20-         return grupousuarios.findByUsuarioIn(usuario);
21-     }
22-
23-     public List<GrupoUsuario> lista() {
24-         return grupousuarios.findAll();
25-     }
26-
27-     public GrupoUsuario buscaGrupo(long codigoGru) {
28-         return grupousuarios.findById(codigoGru);
29-     }
30-
31-     public void merge(GrupoUsuario grupoUsuario, RedirectAttributes attributes) {
32-
33-         if (grupoUsuario.getCodigo() == null) {
34-             try {
35-                 attributes.addFlashAttribute("mensagem", "grupo adicionado com sucesso");
36-                 grupousuarios.save(grupoUsuario);
37-             } catch (Exception e) {
38-                 System.out.println(e);
39-             }
40-
41-         } else {
42-             attributes.addFlashAttribute("mensagem", "Grupo atualizado com sucesso");
43-             grupousuarios.update(grupoUsuario.getCodigo(), grupoUsuario.getDescricao(), grupoUsuario.getCodigo());
44-         }
45-     }
46-
47-
48-     public String remove(Long codigo, RedirectAttributes attributes) {
49-         // Verifico se o grupo está vinculado a algum usuário
50-         int aux = grupousuarios.grupoTemUsuario(codigo);
51-
52-         if (aux > 0) {
53-             attributes.addFlashAttribute("mensagemErro", "Este grupo possui usuários vinculados a ele, verifique");
54-             return "redirect:/grupousuario/" + codigo;
55-         }
56-
57-         try {
58-             grupousuarios.deleteById(codigo);
59-         } catch (Exception e) {
60-             System.out.println(e);
61-         }
62-
63-         return "redirect:/grupousuario/";
64-     }
65-
66-     public String addPermissao(Long codgrupo, Long codpermissao) {
67-         int aux = grupousuarios.grupoTemPermissao(codgrupo, codpermissao);
68-
69-         if (aux > 0) {
70-             throw new RuntimeException("Esta permissão já está adicionada a este grupo");
71-         }
72-
73-         try {
74-             grupousuarios.addPermissao(codgrupo, codpermissao);
75-         } catch (Exception e) {
76-             System.out.println(e);
77-             throw new RuntimeException("Erro ao tentar adicionar permissão, chame o suporte");
78-         }
79-
80-         return "Permissão adicionada com sucesso";
81-     }
82- }
```

# Gerenciar Caixa

- Testes realizados:
  - Testa se a observação padrão para uma saída de caixa (sangria) é aplicada corretamente
  - Testa se a observação padrão para uma entrada de caixa (suprimento) é aplicada corretamente
  - Testa se o lançamento de uma entrada é bloqueado quando o caixa está fechado
  - Testa se o sistema retorna uma lista vazia quando não há lançamentos no caixa
  - Testa se o sistema retorna corretamente um lançamento quando há um único lançamento no caixa
  - Testa se o sistema mostra uma mensagem de "saldo insuficiente" ao tentar realizar uma saída de caixa com valor maior do que o saldo disponível

# Gerenciar Caixa

PROBLEMS DEBUG CONSOLE TERMINAL OUTPUT PORTS GIT LENS COMMENTS

```
[INFO] --- maven-resources-plugin:3.0.1:testResources (default-testResources) @ pdv ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /app/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.7.0:testCompile (default-testCompile) @ pdv ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.21.0:test (default-test) @ pdv ---
[INFO]
[INFO] -----
[INFO]  T E S T S
[INFO] -----
[INFO] Running net.originmobi.pdv.service.CaixaLancamentoServiceTest
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.215 s - in net.originmobi.pdv.service.CaixaLancamentoServiceTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.8:report (report) @ pdv ---
[INFO] Loading execution data file /app/target/jacoco.exec
[INFO] Analyzed bundle 'pdv' with 171 classes
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  5.177 s
[INFO] Finished at: 2024-11-04T17:34:50Z
[INFO] -----
```

# Gerenciar Caixa

pdv > net.originmobi.pdv.service > CaixaLancamentoService

## CaixaLancamentoService

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
<a href="#">lancamento(CaixaLancamento)</a>		86%		72%	6	12	4	30	0	1
<a href="#">lancamentosDoCaixa(Caixa)</a>		100%		n/a	0	1	0	1	0	1
<a href="#">CaixaLancamentoService()</a>		100%		n/a	0	1	0	2	0	1
Total	18 of 139	87%	6 of 22	72%	6	14	4	33	0	3


# Gerenciar Caixa

```
3: import java.util.Timestamp;
4: import java.util.List;
5: import java.util.Optional;
6:
7: import org.springframework.beans.factory.annotation.Autowired;
8: import org.springframework.stereotype.Service;
9: import org.springframework.transaction.annotation.Transactional;
10: import org.springframework.transaction.annotation.Transactional;
11:
12: import ret.origimobi.pdv.enumerado.caixa.EstiloLancamento;
13: import ret.origimobi.pdv.enumerado.caixa.Tipolancamento;
14: import ret.origimobi.pdv.model.Caixa;
15: import ret.origimobi.pdv.model.CaixaLancamento;
16: import ret.origimobi.pdv.repository.CaixaLancamentoRepository;
17:
18: @Service
19: public class CaixaLancamentoService {
20:
21:     @Autowired
22:     private CaixaLancamentoRepository caixaLancamento;
23:
24:     @Autowired
25:     UserService usuarios;
26:
27:     private Timestamp datahoraAtual;
28:
29:     public CaixaLancamentoService() {
30:     }
31:
32:     @Transactional(readOnly = false, propagation = Propagation.REQUIRED)
33:     public String lancamento(CaixaLancamento lancamento) {
34:         datahoraAtual = new Timestamp(System.currentTimeMillis());
35:
36:         try {
37:             lancamento.setData_cadastro(datahoraAtual);
38:
39:             if (!lancamento.getCaixa().isPresent())
40:                 throw new RuntimeException("Nenhuma caixa aderida");
41:
42:         }
43:
44:         // se for realizar uma saída de caixa, verifica se tem saldo
45:         // suficiente para isso
46:         if (lancamento.getEstilo().equals(EstiloLancamento.SAIDA)) {
47:             Optional vTotalCaixa = lancamento.getCaixa().map(Caixa::getValor_total);
48:
49:             if (lancamento.getValor() > vTotalCaixa.get()) {
50:                 return "Saldo insuficiente para realizar esta operação";
51:             }
52:         }
53:
54:         // se for do tipo SAIDA, converte o valor que vier para negativo
55:         if (lancamento.getEstilo().equals(EstiloLancamento.SAIDA) && lancamento.getValor() > 0) {
56:             Double valorNegativo = (lancamento.getValor() * -1);
57:             lancamento.setValor(valorNegativo);
58:         }
59:
60:         // verifica o tipo de lançamento e de acordo com ele coloca uma
61:         // observação no lançamento caso a mesma venha vazia.
62:         if (lancamento.getObservacao().isEmpty()) {
63:             String observacao = "";
64:             if (lancamento.getTipo().equals(Tipolancamento.SANGRIA))
65:                 observacao = lancamento.getObservacao().isEmpty() ? "Sangria de caixa" : lancamento.getObservacao();
66:             else if (lancamento.getTipo().equals(Tipolancamento.SUPRIMENTO))
67:                 observacao = lancamento.getObservacao().isEmpty() ? "Suprimento de caixa" : lancamento.getObservacao();
68:             lancamento.setObservacao(observacao);
69:
70:             lancamento.setObservacao(observacao);
71:         }
72:
73:     } catch (Exception e) {
74:         e.printStackTrace();
75:         throw new RuntimeException();
76:     }
77:
78:     try {
79:         caixaLancamento.save(lancamento);
80:     } catch (Exception e) {
81:         e.getMessage();
82:         throw new RuntimeException("Erro ao realizar lançamento, chame o suporte");
83:     }
84:
85:     return "Lancamento realizado com sucesso";
86: }
87:
88: public List<CaixaLancamento> lancamentoDoCaixa(Caixa caixa) {
89:     return caixaLancamento.findByCaixaId(caixa);
90: }
91: }
```

# Gerenciar Caixa

## Caso de teste de sistema: Gerenciar Caixa História de Usuário U2: Gerenciar Caixa

<b>Objetivo:</b> Verificar a exatidão e funcionalidade da tela de Gerenciamento de Caixa, garantindo que as operações de adição de suprimentos, retirada, transferência e fechamento de caixa sejam realizadas corretamente. Além disso, validar que o sistema lida adequadamente com restrições de valor, como valores negativos e saldo insuficiente, e apresenta mensagens de erro apropriadas para garantir a integridade das operações de caixa.	
<b>Informações sobre a execução do teste:</b> <b>Nome do testador:</b> Wesley Ferreira <b>Data (s) do teste:</b> 03/11	<b>Pré-requisitos para este teste:</b> <ul style="list-style-type: none"><li>- Existir pelo menos caixa configurado</li><li>- Estar logado como gerente do sistema</li></ul> <b>Configuração necessária:</b> Instalação do Ubuntu e Docker
<b>NOTAS e RESULTADOS:</b> <ul style="list-style-type: none"><li>- Os testes foram tratados como etapas dentro do escopo do Caso de Uso Gerenciar Caixas</li></ul>	

ETAPAS / RESULTADOS DO SCRIPT DE TESTE					
PASSO	ETAPA / ENTRADA DO TESTE	RESULTADOS ESPERADOS	RESULTADOS REAIS	Requisitos validados	PASSOU / FALHOU
<b>Obtenha direções para um restaurante presente no banco de dados - Requisitos validados:</b>					
1.	Acessar o menu lateral e clicar no ícone de Caixa. 	A tela de Gerenciamento de Caixa é exibida com as opções "Suprimento", "Retirada", "Transferência" e "Fechar".	A tela de Gerenciamento de Caixa foi exibida corretamente, apresentando as opções "Suprimento", "Retirada", "Transferência" e "Fechar".	Gerenciar Caixa	PASSOU





# Adicionar empresa no Banco









- Testes realizados:
  - Verificar se empresa cadastrada não existe
  - Verificar se empresa cadastrada existe
  - Buscar empresa com nome correto
  - Buscar empresa com CNPJ correto
  - Teste merger com empresa já existente

# Adicionar empresa no Banco

```
[INFO] -----  
[INFO] T E S T S  
[INFO] -----  
[INFO] Running net.originmobi.pdv.service.EmpresaServiceTest  
[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 3.492 s - in net.originmobi.pdv.service.EmpresaServiceTest  
[INFO] Results:  
[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0  
[INFO] --- jacoco-maven-plugin:0.8.8:report (report) @ pdv ---  
[INFO] Loading execution data file /app/target/jacoco.exec  
[INFO] Analyzed bundle 'pdv' with 171 classes  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----
```

# Adicionar empresa no Banco

## EmpresaService

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
merger(Long, String, String, String, String, int, int, Long, Long, Long, String, String, String, String, String, Double)		23%		50%	1	2	34	42	0	1
cadastro(Empresa)		0%		n/a	1	1	5	5	1	1
verificaEmpresaCadastrada()		100%		100%	0	2	0	5	0	1
EmpresaService()		100%		n/a	0	1	0	1	0	1
Total	117 of 165	29%	1 of 4	75%	2	6	39	53	1	4

# Adicionar empresa no Banco

```
59. public String merger(Long codigo, String nome, String nome_fantasia, String cnpj, String ie, int serie,
60. int ambiente, Long codRegime, Long codendereco, Long codcidade, String rua, String bairro, String numero,
61. String cep, String referencia, Double aliqCalcCredito) {
62.
63.     if (codigo != null) {
64.         try {
65.             empresas.update(codigo, nome, nome_fantasia, cnpj, ie, codRegime);
66.         } catch (Exception e) {
67.             System.out.println(e);
68.             return "Erro ao salvar dados da empresa, chame o suporte";
69.         }
70.
71.         try {
72.             parametros.update(serie, ambiente, aliqCalcCredito);
73.         } catch (Exception e) {
74.             System.out.println(e);
75.             return "Erro ao salvar dados da empresa, chame o suporte";
76.         }
77.
78.         try {
79.             enderecos.update(codendereco, codcidade, rua, bairro, numero, cep, referencia);
80.         } catch (Exception e) {
81.             System.out.println(e);
82.             return "Erro ao salvar dados da empresa, chame o suporte";
83.         }
84.     } else {
85.         EmpresaParametro parametro = new EmpresaParametro();
86.
87.         try {
88.             parametro.setAmbiente(ambiente);
89.             parametro.setSerie_nfe(serie);
90.             parametro.setCredSN(aliqCalcCredito);
91.             parametros.save(parametro);
92.         } catch (Exception e) {
93.             System.out.println(e);
94.             return "Erro ao salvar dados da empresa, chame o suporte";
95.         }
96.
97.         Optional<RegimeTributario> tributario = regimes.busca(codRegime);
98.         Optional<Cidade> cidade = cidades.busca(codcidade);
99.
100.         LocalDate dataAtual = LocalDate.now();
101.         Endereco endereco = new Endereco(rua, bairro, numero, cep, referencia, Date.valueOf(dataAtual),
102.             cidade.get());
103.
104.         try {
105.             enderecos.cadastrar(endereco);
106.         } catch (Exception e) {
107.             System.out.println(e);
108.             return "Erro ao salvar dados da empresa, chame o suporte";
109.         }
110.
111.         try {
112.             Empresa empresa = new Empresa(nome, nome_fantasia, cnpj, ie, tributario.get(), endereco, parametro);
113.             empresas.save(empresa);
114.         } catch (Exception e) {
115.             System.out.println(e);
116.             return "Erro ao salvar dados da empresa, chame o suporte";
117.         }
118.     }
```

# Adicionar empresa no Banco

Caso de teste de sistema:  
Adicionar empresas

## História de Usuário U2: Adicionar empresas

<b>Objetivo:</b> Adicionar empresas no formulário Instruções: Tudo que estiver em VERMELHO é obrigatório, o restante é opcional e só deve ser colocado se for necessário para esclarecer como o teste foi realizado.	
<b>Informações sobre a execução do teste:</b> <b>Nome do testador:</b> Laryssa <b>Data (s) do teste:</b> 10/24 Local / servidor em uso: Servidor de teste A3 do Google Maps	<b>Pré-requisitos para este teste:</b> Nenhum <b>Versões de software:</b>  <b>Configuração necessária:</b> instalação do docker e clonagem do repositório correspondente <a href="https://github.com/ovvesley/uff-qualidade-teste-pdv">https://github.com/ovvesley/uff-qualidade-teste-pdv</a>
<b>NOTAS e RESULTADOS:</b>	

ETAPAS / RESULTADOS DO SCRIPT DE TESTE					
PASSO	ETAPA / ENTRADA DO TESTE	RESULTADOS ESPERADOS	RESULTADOS REAIS	Requisitos validados	PASSOU / FALHOU
<b>Obtenha direções para um restaurante presente no banco de dados - Requisitos validados: 3.7.1, 3.7.2, 3.7.5, 4.1-4.8</b>					
1.	Complete os campos do formulário com as informações da empresa	Capaz de inserir texto	Capaz de inserir o texto		PASSOU
2.	Pressione o botão de salvar	Dados salvos no banco	Nenhum resultado retornado		PASSOU
3.					
4.					
5.					
6.					
7.					
8.					

# Gerenciar pagamento de despesa

- Testes realizados:
  - Verificar se empresa cadastrada não existe
  - Verificar se empresa cadastrada existe
  - Buscar empresa com nome correto
  - Buscar empresa com CNPJ correto
  - Teste merger com empresa já existente

# PagarService

```
[INFO] Running net.originmobi.pdv.service.PagarServiceTest
[INFO] Tests run: 8, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.91 s - in net.originmobi.pdv.service.PagarServiceTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 8, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.8:report (report) @ pdv ---
[INFO] Loading execution data file /app/target/jacoco.exec
[INFO] Analyzed bundle 'pdv' with 171 classes
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 7.082 s
[INFO] Finished at: 2024-11-04T18:55:57Z
[INFO] -----
```

# PagarService

pdv > net.originmobi.pdv.service > PagarService

## PagarService

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
<a href="#">quitar(Long, Double, Double, Double, Long)</a>	<div><div></div></div>	98%	<div><div></div></div>	75%	2	5	0	36	0	1
<a href="#">listar()</a>	<div><div></div></div>	0%		n/a	1	1	1	1	1	1
<a href="#">cadastrar(Long, Double, String, LocalDate, PagarTipo)</a>	<div><div></div></div>	95%	<div><div></div></div>	50%	1	2	0	17	0	1
<a href="#">PagarService()</a>	<div><div></div></div>	100%		n/a	0	1	0	1	0	1
Total	11 of 304	96%	3 of 10	70%	4	9	1	55	1	4



# PagarService

localhost:5006/net.originmobi.pdv.service/PagarService.java.html

```
44: @Autowired
45: private CaixaLancamentoService lancamentos;
46:
47: public List<Pagar> listar() {
48:     return pagarRepo.findAll();
49: }
50:
51: @Transactional(readOnly = false, propagation = Propagation.REQUIRED)
52: public String cadastrar(Long codFornecedor, Double valor, String obs, LocalDate vencimento, PagarTipo tipo) {
53:     LocalDate dataAtual = LocalDate.now();
54:     DataAtual dataAtual = new DataAtual();
55:
56:     if (obs.isEmpty()) tipo.getDescricao();
57:
58:     Optional<Fornecedor> fornecedor = fornecedores.busca(codFornecedor);
59:     Fornecedor fornecedor = fornecedor.get();
60:
61:     Pagar pagar = new Pagar(obs, valor, dataAtual, fornecedor, tipo);
62:
63:     try {
64:         pagarRepo.save(pagar);
65:     } catch (Exception e) {
66:         e.getMessage();
67:         throw new RuntimeException("Erro ao lancar despesa, chame o suporte");
68:     }
69:
70:     try {
71:         pagarParcelaServ.cadastrar(valor, valor, 0, dataAtual.dataAtualTimeStamp(), vencimento, pagar);
72:     } catch (Exception e) {
73:         e.getMessage();
74:         throw new RuntimeException("Erro ao lancar despesa, chame o suporte");
75:     }
76:
77:     return "Despesa lancada com sucesso";
78: }
79:
80: @Transactional(readOnly = false, propagation = Propagation.REQUIRED)
81: public String quitar(Long codParcela, Double vPago, Double vDesc, Double vVarec, Long codCaixa) {
82:
83:     Optional<PagarParcela> parcela = pagarParcelaServ.busca(codParcela);
84:     DecimalFormat df = new DecimalFormat("00.00");
85:
86:     if (vPago > Double.valueOf(df.format(parcela.map(PagarParcela::getValorRestante).get().replace(".", "-"))))
87:         throw new RuntimeException("Valor de pagamento invalido");
88:
89:     Double vQuitado = (vPago + vVarec) * parcela.map(PagarParcela::getValorPago).get();
90:     Double vRestante = (parcela.map(PagarParcela::getValorRestante).get() - (vPago + vDesc));
91:     Double vDesconto = parcela.map(PagarParcela::getValorDesconto).get() + vDesc;
92:     Double vAcrecimo = parcela.map(PagarParcela::getValorAcrecimo).get() + vVarec;
93:
94:     vRestante = vRestante < 0 ? 0.0 : vRestante;
95:
96:     int quitado = Double.valueOf(df.format(vRestante).replace(".", "-")) <= 0 ? 1 : 0;
97:
98:     DataAtual dataAtual = new DataAtual();
99:
100:     parcela.get().setValor_pago(vQuitado);
101:     parcela.get().setValor_restante(vRestante);
102:     parcela.get().setValor_desconto(vDesconto);
103:     parcela.get().setValor_acrecimo(vAcrecimo);
104:     parcela.get().setQuitado(quitado);
105:     parcela.get().setUltPagamento(dataAtual.dataAtualTimeStamp());
106:
107:     try {
108:         pagarParcelaServ.merge(parcela.get());
109:     } catch (Exception e) {
110:         e.getMessage();
111:         throw new RuntimeException("Ocorreu um erro ao realizar o pagamento, chame o suporte");
112:     }
113:
114:     Aplicacao aplicacao = Aplicacao.getInstance();
115:
116:     Usuario usuario = usuarios.busca(usuario.aplicacao.getUsuarioAtual());
117:     Optional<Caixa> caixa = caixas.busca(codCaixa);
118:
119:     if (vPago + vVarec + caixa.map(Caixa::getValorTotal).get())
120:         throw new RuntimeException("Saldo insuficiente para realizar este pagamento");
121:
122:     try {
123:         CaixaLancamento lancamento = new CaixaLancamento("Referente a pagamento de despesas", vPago + vVarec,
124:             TipoLancamento.PAGAMENTO, TipoLancamento.SAIDA, caixa.get(), usuario);
125:
126:         // vincula a parcela do pagar ao caixa lancamento
127:         lancamento.setParcelaPagar(parcela.get());
128:
129:         lancamentos.lancamento(lancamento);
130:     } catch (Exception e) {
131:         e.getMessage();
132:         throw new RuntimeException("Ocorreu um erro ao realizar o pagamento, chame o suporte");
133:     }
134:
135:     return "Pagamento realizado com sucesso";
136: }
137:
138: }
```



**X Futuramente  
na entrega 2...**