



**INSTITUTO FEDERAL**

Rio Grande do Norte

Campus Natal-Central

## Plano de Testes

### Fretex

SEMINÁRIO DE ORIENTAÇÃO AO PROJETO DE DESENVOLVIMENTO DE SISTEMA  
COORPORATIVO

**Professor:**

Marcelo Romulo

**Equipe:**

Arthur Medeiros

Italo Gabriel

Lucas Oliveira

Tonny Wesley

Mathews Dantas

Sávio Araújo

NATAL  
2023

# Histórico de Revisões

Data	Versão	Descrição	Autor
06 de julho de 2023	1.0	Release Inicial	Mathews Dantas e Italo Gabriel

# Índice

<b>1. Introdução</b>	4
1.1 Objetivos	4
1.2 A Fretex	4
1.2.1 Escopo	4
1.2.2 Identificação do projeto	4
<b>2. Requisitos a Testar</b>	5
2.1 Teste de Integridade de Dados e do Banco de Dados	5
2.2 Teste Unitário	5
2.3 Teste de Sistema	6
<b>3. Estratégia de teste</b>	6
3.1 Tipos de teste	6
3.1.1 Teste de Integridade de Dados e do Banco de Dados	6
3.1.2 Teste Unitário	7
3.1.3 Teste de Sistema	7
3.1.4 Ferramentas	8
<b>4. Recursos</b>	9
4.1 Trabalhadores	9
4.2 Sistema	10
<b>5. Cronograma</b>	11

# **1. Introdução**

## **1.1 Objetivos**

Esse documento como Plano de Testes da Fretex como software. compõe-se dos seguintes objetivos:

- Identificar informações de projeto existentes e os componentes de software que devem ser testados.
- Listar os Requisitos a Testar recomendados (alto nível).
- Recomendar e descrever as estratégias de teste a serem empregadas.
- Identificar os recursos necessários e prover uma estimativa dos esforços de teste.
- Listar os elementos resultantes do projeto de testes.

## **1.2 A Fretex**

A FRETEx é uma plataforma web especializada no transporte de encomendas urbanas, com foco em cargas domésticas. Nosso objetivo é simplificar a comunicação entre transportadores autônomos e clientes, proporcionando uma solução simples e justa para suas demandas diárias. Para atender às necessidades dos nossos clientes e dos freteiros que buscam uma renda alternativa, decidimos lançar nossa plataforma na web. Assim, nossa plataforma atua como intermediária e facilitadora de comunicação entre as partes, oferecendo um ambiente seguro, intuitivo e de fácil acesso.

### **1.2.1 Escopo**

A Fretex passará pelos testes unitários, de integração e de sistema para garantir a qualidade do software. Os testes unitários verificam o comportamento individual de unidades de código, enquanto os testes de integração lidam com a qualidade funcional, bases de dados, interface gráfica e controle de acesso. Por sua vez, os testes de sistema avaliam o sistema como um todo, verificando se atende aos requisitos do usuário e está pronto para implantação, abrangendo aspectos como desempenho e usabilidade.

### **1.2.2 Identificação do projeto**

A tabela abaixo identifica a documentação e disponibilidade usados para desenvolver o plano de testes:

Documento	Criado ou Disponível	Recebido ou Revisado
Especificação de Requisitos	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Plano de Projeto	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Modelo de Análise	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Modelo de Projeto	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Documento de Arquitetura	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Protótipo	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não
Manual do Usuário	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não	<input type="checkbox"/> Sim <input checked="" type="checkbox"/> Não
Lista de Riscos	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não	<input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não

## 2. Requisitos a Testar

A lista abaixo identifica aqueles itens – use cases, requisitos funcionais e não funcionais – que foram identificados como alvos de teste. Essa lista representa o que será testado.

### 2.1 Teste de Integridade de Dados e do Banco de Dados

- Verifique que os usuários do tipo Cliente podem ser cadastrados, atualizados, consultados e removidos.
- Verifique que os usuários do tipo Freteiro podem ser cadastrados, atualizados, consultados e removidos.

### 2.2 Teste Unitário

- Verifique no objeto Freteiro se o campo `capa\_foto` tem as propriedades blank e null configuradas.
- Verifique no objeto Freteiro se o campo `url\_foto` tem a propriedade `upload\_to` configurada corretamente.
- Verifique no objeto Freteiro se o campo `cpf` tem o comprimento máximo(`max\_length`) igual a 15

- Verifique no objeto Freteiro se o atributo `endereco` do objeto `Freteiro` é uma instância da classe `Endereco`
- Verifique no objeto Pedido se o valor do campo `status` está presente na lista de opções de escolha definidas em `Pedido.STATUS\_CHOICES`
- Verifique no objeto Pedido se o valor do campo `turno` está presente na lista de opções de escolha definidas em `Pedido.TURNO`
- Verifique no objeto Pedido se o campo `data\_criacao` não é nulo após a criação do pedido

## 2.3 Teste de Sistema

- Verifique que a camada de usuário por meio de diferentes tipos de requisições (POST, GET, DELETE, PUT) estão surtindo o efeito esperado no banco de dados para classe de Cliente.
- Verifique que a camada de usuário por meio de diferentes tipos de requisições (POST, GET, DELETE, PUT) estão surtindo o efeito esperado no banco de dados para classe de Freteiro.

## 3. Estratégia de teste

### 3.1 Tipos de teste

#### 3.1.1 Teste de Integridade de Dados e do Banco de Dados

Objetivo do Teste:	Garantir que os métodos e processos de acesso ao banco de dados funcionam apropriadamente e sem corrupção dos dados
Técnica:	<ul style="list-style-type: none"> <li>• Invocar cada método e processo de acesso ao banco de dados, alimentando cada um com dados ou requisições de dados válidos e inválidos.</li> <li>• Inspeccionar o banco de dados para garantir que os dados foram populados como pretendido, que todos os eventos do banco de dados ocorreram apropriadamente, ou revisar os dados retornados para garantir que os dados corretos foram recuperados pelas razões corretas.</li> </ul>
Critério de Finalização:	Todos os métodos e processos de acesso à base de dados funcionam como projetados e sem nenhuma corrupção de dados.

Considerações Especiais:	<ul style="list-style-type: none"> <li>• O teste pode necessitar de um ambiente de desenvolvimento ou drivers de SGBD para inserir ou modificar os dados diretamente na base de dados</li> <li>• Processos devem ser invocados manualmente</li> <li>• Bases de dados pequenas ou minimizadas (número de registros limitados) devem ser usados para aumentar a visibilidade de eventos não-aceitáveis.</li> </ul>
--------------------------	--

### 3.1.2 Teste Unitário

Objetivo do Teste:	Garantir que a corretude do comportamento individual de unidades de código, como funções, métodos ou classes. Os testes unitários visam garantir que cada unidade de código funcione conforme o esperado, isoladamente dos demais componentes do sistema.
Técnica:	Verificar o comportamento de cada unidade de código isoladamente. Isso é feito fornecendo entradas específicas para os métodos ou funções e verificando se as saídas resultantes estão de acordo com o esperado. É comum utilizar mocks ou stubs para simular o comportamento de dependências externas e isolar as unidades de código a serem testadas.
Critério de Finalização:	<ul style="list-style-type: none"> <li>• Todos os testes planejados foram executados.</li> <li>• Todos os defeitos identificados foram tratados.</li> </ul>
Considerações Especiais:	Os testes unitários são executados em um ambiente isolado, portanto, não há interação direta com um banco de dados real.

### 3.1.3 Teste de Sistema

Objetivo do Teste:	Garantir que o sistema como um todo atende aos requisitos do usuário e está pronto para ser implantado. Ele envolve a interação com a camada de usuário por meio de diferentes tipos de requisições (POST, GET, DELETE, PUT) e verifica se essas requisições estão surtindo o efeito esperado no banco de dados. O teste de sistema busca identificar falhas, erros de integração, problemas de desempenho, usabilidade e outros aspectos relevantes para garantir a qualidade do sistema.
Técnica:	A técnica utilizada no teste de sistema pode variar, mas geralmente envolve a criação de casos de teste que abrangem as principais funcionalidades do sistema. Esses casos de teste

	são executados em um ambiente que simula o ambiente de produção, utilizando dados reais ou dados fictícios representativos. É comum utilizar frameworks e ferramentas de teste automatizado para agilizar e facilitar a execução dos casos de teste.
Critério de Finalização:	O critério de finalização do teste de sistema é atingido quando todas as funcionalidades do sistema foram testadas e verificadas quanto à conformidade com os requisitos, sem a ocorrência de problemas críticos ou bloqueadores que impeçam a implantação do sistema. Além disso, também é importante que os resultados do teste estejam devidamente documentados e que quaisquer problemas encontrados sejam registrados para posterior correção.
Considerações Especiais:	<ul style="list-style-type: none"> <li>● Integração: O teste de sistema envolve testar a integração entre os diferentes componentes do sistema, como a interação entre a camada de usuário e o banco de dados. É importante garantir que as informações sejam transmitidas corretamente e que as ações do usuário tenham o efeito esperado no sistema.</li> <li>● Conformidade com requisitos: O teste de sistema visa garantir que o sistema atenda aos requisitos definidos. É importante verificar se todas as funcionalidades foram implementadas corretamente e se os resultados estão de acordo com o esperado.</li> </ul>

### 3.1.4 Ferramentas

As seguintes ferramentas serão empregadas para esse projeto:

	Ferramenta	Vendedor
Gerenciamento de Teste	Coverage.py	Coverage
Projeto de Teste	pytest	pytest
Gerenciamento de Projeto	GitHub Project	GitHub
Ferramentas do SGBD	SQLite3	SQLite



## 4. Recursos

Essa seção apresenta os recursos recomendados para o projeto do Fretex, suas principais responsabilidades, e seus conhecimentos ou conjunto de habilidades.

### 4.1 Trabalhadores

Essa tabela mostra as suposições de recrutamento para o projeto.

Recursos Humanos		
Trabalhador	Recursos Mínimos Recomendados	Responsabilidades Específicas ou Comentários
Gerente de Teste, Gerente do Projeto de Teste	Arthur Medeiros	Fornecer supervisão gerencial. Responsabilidades: <ul style="list-style-type: none"><li>• provê direcionamento técnico</li><li>• adquire recursos apropriados</li><li>• fornece relatórios de gerenciamento</li></ul>
Test Designer	Lucas Oliveira	Identifica, prioriza, e implementa os casos de teste. Responsabilidades: <ul style="list-style-type: none"><li>• gera o plano de teste</li><li>• cria o modelo de teste</li><li>• avalia a efetividade do esforço de teste</li></ul>
Testador	Mathews Dantas Italo Gabriel	Executa os testes. Responsabilidades: <ul style="list-style-type: none"><li>• executar os testes</li><li>• registrar os resultados</li><li>• reestabelecer-se dos erros</li><li>• documentar solicitações de mudança</li></ul>
Administrador do Sistema de Teste	Tonny Wesley	Garante que o ambiente e os bens de teste sejam gerenciados e mantidos. Responsabilidades:

		<ul style="list-style-type: none"> <li>• administrar o sistema de gerenciamento teste</li> <li>• instalar e gerenciar o acesso do trabalhador ao sistema de testes</li> </ul>
Gerente do Banco de Dados, Administrador do Banco de Dados	Arthur Medeiros	<p>Garante que o ambiente e bens de teste de dados (banco de dados) sejam gerenciados e mantidos.</p> <p>Responsabilidades:</p> <ul style="list-style-type: none"> <li>• administrar os dados de teste (base de dados)</li> </ul>
Designer	Tonny Wesley	<p>Identifica e define as operações, atributos, e associações das classes de teste.</p> <p>Responsabilidades:</p> <ul style="list-style-type: none"> <li>• identificar e definir as classes de teste</li> <li>• identificar e definir os pacotes de teste</li> </ul>
Implementador	Mathews Dantas Italo Gabriel	<p>Implementa e faz os testes unitários das classes e pacotes de teste.</p> <p>Responsabilidades:</p> <ul style="list-style-type: none"> <li>• cria as classes e pacotes de teste implementados no modelo de teste</li> </ul>

## 4.2 Sistema

A tabela seguinte expõe os recursos do sistema para o projeto de teste.

Recursos do Sistema
<p>Servidor de Banco de Dados</p> <p>Sqlite3 DataBase Server</p>

## 5. Cronograma

Milestone	Data de Início	Data de Término
Planejar Teste	29/06/23	29/06/23
Projetar Teste	30/06/23	30/06/23
Implementar Teste	30/06/23	30/06/23
Executar Teste	02/07/23	02/07/23
Avaliar Teste	06/07/23	06/07/23

Anexo 01 -