**Mercado**

**Plano de Teste**

1. **Introdução**

Um programa simples que controla a venda de produtos a clientes um mercado local, o qual oferece as seguintes funcionalidades:

- Gerenciamento de produtos: inclusão e alteração;

- Gerenciamento de clientes: inclusão e alteração;

- Alteração da quantidade em estoque dos produtos;

- Consulta de produtos por descrição;

- Registro de vendas de produtos para clientes;

- Cálculo do faturamento em um determinado mês;

1. **Equipe de Teste**

A equipe de teste é composta pelos seguintes membros:

- Leonardo Destro Bronzato;

- Lucas Antoniale Callegari;

- Lucas Fernando Bocanegra;

- Matheus dos Santos Freitas;

1. **Abordagem**

Será executado teste de unidade, onde o módulo a ser testado é o método. Aplicou-se Teste Funcional Sistemático como critério para criação dos casos de teste. O processo de teste terá o apoio da ferramenta JUnit.

|  |  |
| --- | --- |
| **Método** | **Descrição** |
| Public Produto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException | Inclui um novo produto no Mercado. |
| public void setCodigo(int codigo) throws RuntimeException | Altera o código do produto. |
| public void setDescricao(String descricao) throws RuntimeException | Altera a descrição do produto. |
| public void setPrecoVenda(double precoVenda) throws RuntimeException | Altera o preço de venda do produto. |
| public void setUnidade(String Unidade) throws RuntimeException | Altera a unidade do produto. |
| public void setEstoque(double estoque) throws RuntimeException | Altera a quantidade do produto no estoque. |
| public void setStatus(int status) throws RuntimeException | Altera o status do produto. |
| public void setPrecoCompra(double precoCompra) throws RuntimeException | Altera o preço de compra do produto. |
| public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) | Cadastra um novo cliente |
| public ArrayList<Integer> consultaProduto(String descricao) throws RuntimeException | Retorna o código dos produtos a partir de uma descrição. |
| Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException | Inclui um novo registro de venda no Mercado. |
| public void setNumero(int numero) throws RuntimeException | Altera o número da venda. |
| public void setData(String data) throws RuntimeException | Altera a data da venda. |
| public void setQuantidades(ArrayList<Double> quantidades)throws RuntimeException | Altera a quantidade de itens vendidos. |
| public void setProdutos(ArrayList<Produto> produtos) throws RuntimeException | Altera os produtos vendidos. |
| public double calculaFaturamento(int mes) throws RuntimeException | Retorna o cálculo do faturamento. |
| public void alteraEstoque(int codigo, double estoque) throws RuntimeException | Altera a quantidade em estoque do produto. |

1. **Critérios de Aprovação**

O critério de aprovação é ter uma porcentagem de 90% de execução certa dos testes. Caso não consiga essa porcentagem, o sistema volta para o desenvolvedor para a correção dos defeitos.

1. **Cronograma**

|  |  |
| --- | --- |
| **Data** | **Atividade** |
| 30/04/2015 | Início da criação do plano de teste e definição da equipe |
| 10/05/2015 | Desenvolvimento do sistema |
| 15/05/2015 | Criação dos casos de teste |
| 30/06/2015 | Execução dos casos de teste |
| 3/06/2015 | Correção dos defeitos |
| 10/06/2015 | Relatório final das execuções |

1. **Classes de Equivalência e Casos de Teste**
   1. **Método**
      1. **Classes de Equivalência** 
         1. **Gerenciamento de Produtos – Inclusão e Alteração**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condição de Entrada** | **Classe Válida** | **Num** | **Classe Inválida** | **Num** |
| Código | Value >= 0 | CV1 | Value < 0 | CI1 |
| Código único | CV2 | Código já existente | CI2 |
| Descrição | 3 <= Lenght <= 64 | CV3 | Lenght < 3 | CI3 |
| CV4 | Lenght > 64 | CI4 |
| Alfanumérico | CV5 | Caracteres especiais | CI5 |
| Preço de Compra | Value > 0.0 | CV6 | Value <= 0.0 | CI6 |
| Preço de Venda | Value > 0.0 | CV7 | Valeu <= 0.0 | CI7 |
| Value >= Preço\_de\_ compra.Value | CV8 | Value < Preço\_de\_compra.Value | CI8 |
| Unidade | Lenght = 2 | CV9 | Lenght != 2 | CI9 |
| Alfabético | CV10 | Caracteres especiais e números | CI10 |
| Estoque | Value >= 0.0 | CV11 | Value < 0.0 | CI11 |
| Status | Value = 1 ou 0 | CV12 | Value!= 1 ou 0 | CI12 |
| 0 se e somente se Estoque.Value = 0 | CV13 | Value = 1 se Estoque.Value = 0 | CI13 |
| Saída | RunTimeException | CV14 | Qualquer outra saída | CI14 |
| Sucesso | CV15 | CI15 |

**Casos de Teste – Inclusão de Produtos**

|  |  |
| --- | --- |
| **Caso de Teste CTPI1** | |
| **Descrição** | Verifica se a inclusão do produto é válida |
| **Método(s)** | Public Produto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 0,"Arroz",5.00,6.99,"Un",10,1 |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI2** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando o código como -1 |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | -1,"Arroz",5.00,6.99,"Un",10,1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI3** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, tentando criar um objeto com código 0, porém já há um objeto com código 0 criado |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 0,"Arroz",5.00,6.99,"Un",10,1 |
| **Saída** | Exceção |
| **Dependências** | É necessário criar um objeto da classe Mercado e adicionar um produto com código, pois é a classe Mercado que gerencia os códigos dos produtos |

|  |  |
| --- | --- |
| **Caso de Teste CTPI4** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando a descrição com apenas 2 caracteres |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 1,"AB",5.00,6.99,"Un",10,1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI5** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando a descrição com mais de 64 caracteres |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 1,"Abcdefghijklmnopqrstuvwxyz123456789abcdefghijklmnopqrstuvwxyzaabcdefg",5.00,6.99,"Un",10,1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI6** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando a descrição com um caracter especial |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 1,"\*Arroz",-5.00,6.99,"Un",10,1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI7** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando o preço de venda negativo |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 1,"Arroz",5.00,-6.99,"Un",10,1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI8** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando o preço de venda menor que o preço de compra |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 1,"Arroz",5.00,4.00,"Un",10,1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI9** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando a unidade mais de 2 caracteres |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 1,"Arroz",5.00,6.99,"ABC",10,1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI10** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando a unidade com caracteres não alfabéticos |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 1,"Arroz",5.00,6.99,"12",10,1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI11** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando o estoque com um número negativo |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 1,"Arroz",5.00,6.99,"Un",-1,1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI12** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando o status com um caracter diferente de 0 ou 1 |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 1,"Arroz",5.00,6.99,"Un",10,3 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI13** | |
| **Descrição** | Verifica se a inclusão do produto é válida |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 1,"Arroz",5.00,6.99,"Un",0,0 |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI14** | |
| **Descrição** | Verifica se a inclusão do produto é válida |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 1,"Arroz",5.00,6.99,"Un",0,1 |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI15** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando a descrição com menos de 3 caracteres |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 2,"ab",5.00,6.99,"Un",10,1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI16** | |
| **Descrição** | Verifica se a inclusão do produto é válida |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 2,"AbcdefigAbcdefigAbcdefigAbcdefigAbcdefigAbcdefigAbcdefigabcdefig",5.00,6.99,"Un",10,1 |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI17** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando a descrição com 65 caracteres |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 3,"AbcdefigAbcdefigAbcdefigAbcdefigAbcdefigAbcdefigAbcdefigabcdefigh",5.00,5.99,"Un",10,1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI18** | |
| **Descrição** | Verifica se a inclusão do produto é válida |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | -1,"Arroz",5.00,6.99,"Un",10,1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI19** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando o preço de compra com valor igual a 0 |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 3,"Arroz",0.01,6.99,"Un",10,1 |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI20** | |
| **Descrição** | Verifica se a inclusão do produto é válida |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 4,"Arroz",0.00,6.99,"Un",10,1 |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI21** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando a unidade com um caracter |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 4,"Arroz",0.01,0.01,"U",10,1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI22** | |
| **Descrição** | Verifica se a inclusão do produto é válida |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 5,"Arroz",5.00,6.99,"Un",10,1 |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI23** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando um valor negativo no estoque |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 5,"Arroz",5.00,6.99,"Un",0.00,0 |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI24** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando um valor vazio na descrição |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 6,"Arroz",5.00,6.99,"Un",-0.01,0 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI25** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando um valor vazio na unidade |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 7,"",5.00,6.99,"Un",10,0 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTPI26** | |
| **Descrição** | Verifica se a inclusão do produto é inválida, colocando um valor vazio na unidade |
| **Método(s)** | Public Produto (int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) throws RuntimeException |
| **Entrada** | 7,"Arroz",5.00,6.99,"",10,0 |
| **Saída** | Exceção |
| **Dependências** |  |

**Casos de Teste – Alteração de Produtos**

|  |  |
| --- | --- |
| **Caso de Teste CTPA1** | |
| **Descrição** | Altera o código do produto e espera-se um erro |
| **Método(s)** | public void setCodigo(int codigo) throws RuntimeException |
| **Entrada** | -1 |
| **Saída** | Exceção |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA2** | |
| **Descrição** | Altera descrição do produto e espera-se um erro |
| **Método(s)** | public void setDescricao(String descricao) throws RuntimeException |
| **Entrada** | “AB” |
| **Saída** | Exceção |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA3** | |
| **Descrição** | Altera a descrição do produto e espera-se um erro |
| **Método(s)** | public void setDescricao(String descricao) throws RuntimeException |
| **Entrada** | “Abcdefghijklmnopqrstuvwxyz123456789abcdefghijklmnopqrstuvwxyzabcdefg” |
| **Saída** | Exceção |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA4** | |
| **Descrição** | Altera a descrição do produto e espera-se um erro |
| **Método(s)** | public void setDescricao(String descricao) throws RuntimeException |
| **Entrada** | \*Arroz |
| **Saída** | Exceção |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA5** | |
| **Descrição** | Altera o preço de venda do produto e espera-se um erro |
| **Método(s)** | public void setPrecoVenda(double precoVenda) throws RuntimeException |
| **Entrada** | -6.99 |
| **Saída** | Exceção |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA6** | |
| **Descrição** | Altera o preço de venda do produto e espera-se um erro |
| **Método(s)** | public void setPrecoVenda(double precoVenda) throws RuntimeException |
| **Entrada** | 4.00 |
| **Saída** | Exceção |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA7** | |
| **Descrição** | Altera a unidade do produto e espera-se um erro |
| **Método(s)** | public void setUnidade(String Unidade) throws RuntimeException |
| **Entrada** | “ABC” |
| **Saída** | Qualquer outra saída |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA8** | |
| **Descrição** | Altera a unidade do produto e espera-se um erro |
| **Método(s)** | public void setUnidade(String Unidade) throws RuntimeException |
| **Entrada** | “12” |
| **Saída** | Exceção |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA9** | |
| **Descrição** | Altera o estoque do produto e espera-se um erro |
| **Método(s)** | public void setEstoque(double estoque) throws RuntimeException |
| **Entrada** | -1 |
| **Saída** | Exceção |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA10** | |
| **Descrição** | Altera o status produto e espera-se um erro |
| **Método(s)** | public void setStatus(int status) throws RuntimeException |
| **Entrada** | 3 |
| **Saída** | Exceção |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA11** | |
| **Descrição** | Altera o estoque e o status do produto e espera-se um erro |
| **Método(s)** | public void setEstoque(double estoque) throws RuntimeException  public void setStatus(int status) throws RuntimeException |
| **Entrada** | 0  0 |
| **Saída** | Exceção |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA12** | |
| **Descrição** | Altera o estoque e o status do produto |
| **Método(s)** | public void setEstoque(double estoque) throws RuntimeException  public void setStatus(int status) throws RuntimeException |
| **Entrada** | 0  1 |
| **Saída** | Sucesso |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA13** | |
| **Descrição** | Altera a descrição do produto e espera-se um erro |
| **Método(s)** | public void setDescricao(String descricao) throws RuntimeException |
| **Entrada** | “” |
| **Saída** | Exceção |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA14** | |
| **Descrição** | Altera a descrição do produto |
| **Método(s)** | public void setDescricao(String descricao) throws RuntimeException |
| **Entrada** | "AbcdefigAbcdefigAbcdefigAbcdefigAbcdefigAbcdefigAbcdefigabcdefig" |
| **Saída** | Sucesso |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA15** | |
| **Descrição** | Altera a descrição do produto e espera-se um erro |
| **Método(s)** | public void setDescricao(String descricao) throws RuntimeException |
| **Entrada** | "AbcdefigAbcdefigAbcdefigAbcdefigAbcdefigAbcdefigAbcdefigabcdefigh" |
| **Saída** | Exceção |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA16** | |
| **Descrição** | Altera o preço de compra do produto |
| **Método(s)** | public void setPrecoCompra(double precoCompra) throws RuntimeException |
| **Entrada** | 0.0 |
| **Saída** | Sucesso |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA17** | |
| **Descrição** | Altera o preço de compra do produto e espera-se um erro |
| **Método(s)** | public void setPrecoCompra(double precoCompra) throws RuntimeException |
| **Entrada** | -0.001 |
| **Saída** | Exceção |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA18** | |
| **Descrição** | Altera o preço de compra e o preço de venda do produto |
| **Método(s)** | public void setPrecoCompra(double precoCompra) throws RuntimeException  public void setPrecoVenda(double precoVenda) throws RuntimeException |
| **Entrada** | 0.01  0.01 |
| **Saída** | Sucesso |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA19** | |
| **Descrição** | Altera a unidade do produto e espera-se um erro |
| **Método(s)** | public void setUnidade(String unidade) throws RuntimeException |
| **Entrada** | “U” |
| **Saída** | Exceção |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA20** | |
| **Descrição** | Altera o estoque e o status do produto |
| **Método(s)** | public void setEstoque(double estoque) throws RuntimeException  public void setStatus(int status) throws RuntimeException |
| **Entrada** | 0.00  0 |
| **Saída** | Sucesso |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA21** | |
| **Descrição** | Altera o estoque do produto e espera-se um erro |
| **Método(s)** | public void setEstoque(double estoque) throws RuntimeException |
| **Entrada** | -0.01 |
| **Saída** | Exceção |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA22** | |
| **Descrição** | Altera a descrição do produto e espera-se um erro |
| **Método(s)** | public void setDescricao(String descricao) throws RuntimeException |
| **Entrada** | “” |
| **Saída** | Qualquer outra saída |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

|  |  |
| --- | --- |
| **Caso de Teste CTPA23** | |
| **Descrição** | Altera a unidade do produto e espera-se um erro |
| **Método(s)** | public void setUnidade(String unidade) throws RuntimeException |
| **Entrada** | “” |
| **Saída** | Exceção |
| **Dependências** | Necessário a criação de um Produto com sucesso, para que as alterações sejam feitas sobre esse objeto |

* + - 1. **Gerenciamento de Clientes – Inclusão e Alteração**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condição de Entrada** | **Classe Válida** | **Num** | **Classe Inválida** | **Num** |
| Nome | 2 <= Lenght <= 128 | CV16 | Lenght < 2 | CI16 |
| CV17 | Lenght > 128 | CI17 |
| 2 <= Palavras | CV18 | 2 > Palavras | CI18 |
| Alfabético | CV19 | Não Alfabético | CI19 |
| CPF | Lenght = 11 | CV20 | Lenght != 11 | CI20 |
| \*Dígitos 10 e 11= Gerador de CPF | CV21 | \*Dígitos 10 e 11!= Gerador de CPF | CI21 |
| CPF único | CV22 | CPF já existente | CI22 |
| Endereço | Lenght <= 256 | CV23 | Lenght > 256 | CI23 |
| Alfanumérico | CV24 | Caracteres especiais | CI24 |
| Telefone | Seguindo o template:  (XX) XXXXXXXXX onde X = numero | CV25 | Diferente do template:  (XX) XXXXXXXXX onde X = numero | CI25 |
| Números e Símbolos | CV26 | Caracteres Especiais e Letras | CI26 |
| E-mail | Seguindo o template: “usuario@dominio.complemento” | CV27 | Diferente do template: “usuario@dominio.complemento” | CI27 |
| “usuário” e “domínio” são iniciados por caracteres alfabéticos | CV28 | “usuário” e/ou “domínio” são iniciados por caracteres não alfabéticos | CI28 |
| “usuário” e “domínio” caracteres não iniciais = caracteres alfanuméricos | CV29 | “usuário” e “domínio” caracteres não iniciais = caracteres especiais | CI29 |
| “complemento” = “com” ou “com.br” | CV30 | “complemento” != “com” ou “com.br” | CI30 |
| Status | Value = 1 ou 0, 1 se e somente se “ativo”, e 0 se e somente se “inativo” | CV31 | Value != 1 ou Value != 0 | CI31 |
| Saída | RunTimeException | CV32 | Qualquer outra saída | CI32 |
| Sucesso | CV33 |

*\*Gerador de CPF com base no site: http://www.geradorcpf.com/algoritmo\_do\_cpf.htm.*

**Casos de Teste – Inclusão de Clientes**

|  |  |
| --- | --- |
| **Caso de Teste CTCI1** | |
| **Descrição** | Verifica se a inclusão do cliente é válida |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "Lendro Salazar","13705137123","Av. São Carlos, 35","(16)331432488","leSalazar@hotmail.com",1 |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCI2** | |
| **Descrição** | Verifica se a inclusão do cliente é inválida |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "L","13705137123","Av. São Carlos, 35", "(16)331432488","leSalazar@hotmail.com",1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCI3** | |
| **Descrição** | Verifica se a inclusão do cliente é inválida |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "LeandroLeandroLeandroLeandroLeandroLeandroLeandroLeandroLeandroLeandroLeandroLeandroLeandroLeandroLeandroLeandroLeandroLeandro", "13705137123","Av. São Carlos, 35", "(16)331432488","leSalazar@hotmail.com",0 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCI4** | |
| **Descrição** | Verifica se a inclusão do cliente é inválida |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "Lendro","13705137123","Av. São Carlos, 35", "(16)331432488","leSalazar@hotmail.com",1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCI5** | |
| **Descrição** | Verifica se a inclusão do cliente é inválida |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "L3ndro Salazar","13705137123","Av. São Carlos, 35", "(16)331432488","leSalazar@hotmail.com",1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCI6** | |
| **Descrição** | Verifica se a inclusão do cliente é inválida |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "Lendro Salazar","137051371234","Av. São Carlos, 35", "(16)331432488748569854126","leSalazar@hotmail.com",1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCI7** | |
| **Descrição** | Verifica se a inclusão do cliente é inválida |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "Lendro Salazar","13705137124","Av. São Carlos, 35", "(16)331432488","leSalazarhotmail.com",1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCI8** | |
| **Descrição** | Verifica se a inclusão do cliente é inválida, sendo que o cliente já está cadastrado |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "Lendro Salazar","13705137123","Av. São Carlos, 35","(16)331432488","leSalazar@hotmail.com",1  "Alef Salazar","13705137123","Av. São Carlos, 35","(16)331432488","leSalazar@hotmail.com",1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCI9** | |
| **Descrição** | Verifica se a inclusão do cliente é inválida |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "Lendro Salazar","13705137123","Av.SãoCarlos, 35Av.SãoCarlos, 35Av.SãoCarlos,35Av.SãoCarlos,35Av.SãoCarlos,35Av.SãoCarlos,35Av.SãoCarlos,35Av.SãoCarlos, 35Av.SãoCarlos, "+"35Av.SãoCarlos, 35Av.SãoCarlos, 35Av.SãoCarlos, 35Av.SãoCarlos, 35Av.SãoCarlos, 35Av.SãoCarlos, 35Av.SãoCarlos, 35Av.SãoCarlos, 35","(16)331432488","leSalazar@hotmail.com",1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCI10** | |
| **Descrição** | Verifica se a inclusão do cliente é inválida |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "Lendro Salazar","0000000000","Av. São Carlos\*, 35","(16)331432488","leSalazar@hotmail.com",1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCI11** | |
| **Descrição** | Verifica se a inclusão do cliente é inválida |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "Lendro Salazar","13705137123","Av. São Carlos, 35","(16)3314-32488","leSalazar@hotmail.com",1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCI12** | |
| **Descrição** | Verifica se a inclusão do cliente é inválida |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "Lendro Salazar","13705137123","Av. São Carlos, 35","(16)33143248a","leSalazar@hotmail.com",1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCI13** | |
| **Descrição** | Verifica se a inclusão do cliente é inválida |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "Lendro Salazar","13705137123","Av. São Carlos, 35","(16)331432488","@leSalazar.hotmail.com",1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCI14** | |
| **Descrição** | Verifica se a inclusão do cliente é inválida |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "Lendro Salazar","13705137123","Av. São Carlos, 35","(16)331432488","3leSalazar@hotmail.com",1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCI15** | |
| **Descrição** | Verifica se a inclusão do cliente é inválida |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "Lendro Salazar","13705137123","Av. São Carlos, 35","(16)331432488","le\*Salazar@hotmail.com",0 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCI16** | |
| **Descrição** | Verifica se a inclusão do cliente é inválida |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "Lendro Salazar","13705137123","Av. São Carlos, 35","(16)331432488","leSalazar@hotmail.comx",1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCI17** | |
| **Descrição** | Verifica se a inclusão do cliente é inválida |
| **Método(s)** | public Cliente(String nome, String cpf, String endereco, String telephone, String email, int status) |
| **Entrada** | "Lendro Salazar","13705137123","Av. São Carlos, 35","(16)331432488","leSalazar@hotmail.com",2 |
| **Saída** | Exceção |
| **Dependências** |  |

* + - 1. **Consulta de Produtos por Descrição**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Num** | **Classes Inválidas** | **Num** |
| Descrição | 3 <= Lenght <= 64 | CV34 | 3 > Lenght | CI33 |
| 64 < Lenght | CI34 |
| Caracteres Alfanuméricos | CV35 | Caracteres Especiais | CI35 |
| Saída | Value >= 0 | CV36 | Value = -1 | CI36 |
| Value == Null | CV37 |

**Casos de Teste – Consulta de Produtos por Descrição**

|  |  |
| --- | --- |
| **Caso de Teste CTCPD1** | |
| **Descrição** | Consulta um produto e espera-se um sucesso na consulta |
| **Método(s)** | public ArrayList<Integer> consultaProduto(String descricao) throws RuntimeException |
| **Entrada** | assertTrue(m.consultaProdutos("Arroz").containsAll(array)); |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCPD2** | |
| **Descrição** | Consulta um produto e espera-se um erro na consulta |
| **Método(s)** | public ArrayList<Integer> consultaProduto(String descricao) throws RuntimeException |
| **Entrada** | assertFalse(m.consultaProdutos("AB").containsAll(array)); |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCPD3** | |
| **Descrição** | Consulta um produto e espera-se um erro na consulta |
| **Método(s)** | public ArrayList<Integer> consultaProduto(String descricao) throws RuntimeException |
| **Entrada** | assertFalse(m.consultaProdutos("CajuCajuCajuCajuCajuCajuCajuCajuCajuCajuCajuCajuCajuCajuCajuCajuu").containsAll(array)); |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCPD4** | |
| **Descrição** | Consulta um produto e espera-se um sucesso na consulta |
| **Método(s)** | public ArrayList<Integer> consultaProduto(String descricao) throws RuntimeException |
| **Entrada** | assertTrue(m.consultaProdutos("Leite").containsAll(array)); |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCPD5** | |
| **Descrição** | Consulta um produto e espera-se um erro sucesso na consulta |
| **Método(s)** | public ArrayList<Integer> consultaProduto(String descricao) throws RuntimeException |
| **Entrada** | assertFalse(m.consultaProdutos("").containsAll(array) && !array.isEmpty()); |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCPD6** | |
| **Descrição** | Consulta um produto e espera-se um erro na consulta |
| **Método(s)** | public ArrayList<Integer> consultaProduto(String descricao) throws RuntimeException |
| **Entrada** | assertFalse(m.consultaProdutos("Macarrao").containsAll(array) && !array.isEmpty()); |
| **Saída** | Sucesso |
| **Dependências** |  |

* + - 1. **Registro de Vendas – Inclusão e Alteração**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condição de Entrada** | **Classe Válida** | **Num** | **Classe Inválida** | **Num** |
| Número do Registro | Value >= 0 | CV38 | Value < 0 | CI37 |
| Número único | CV39 | Número já existente | CI38 |
| Data | 0 < Dia <= 31 | CV40 | Lenght < 3 | CI39 |
| Lenght > 64 | CI40 |
| 0 < Mês <= 12 | CV41 | Mês <=0 | CI41 |
| Mês > 12 | CI42 |
| Ano == Ano Atual | CV42 | Ano < Ano Menor | CI43 |
| Ano > Ano Atual | CI44 |
| Se Mês = 2 então Dia <= 28 | CV43 | Se Mês = 2 então Dia > 28 | CI45 |
| Quantidade de Itens | Value > 0 | CV44 | Value <= 0 | CI46 |
| Quantidade de cada Item | Lenght = Quantidade\_de\_itens.Value | CV45 | Lenght != Quantidade\_de\_itens.Value | CI47 |
| Todos os códigos tem que ser maiores que 0 | CV46 | Há código(s) igual ou menores que 0 (não representam um produto válido) | CI48 |
| Estoque de um determinado código deve ser maior que 0 | CV47 | Estoque de um determinado código é menor ou igual a 0 | CI49 |
| CPF | CPF inválido | CV48 | CPF válido | CI50 |

**Casos de Teste – Inclusão de Registro de Vendas**

|  |  |
| --- | --- |
| **Caso de Teste CTRVI1** | |
| **Descrição** | Verifica se a inclusão do registro de venda é válida |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "0;37413815869;09/06/2015;0;7.00;1;5.00;" |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI2** | |
| **Descrição** | Verifica se a inclusão do registro de venda é inválida, colocando um código inválido |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "-1;37413815869;09/06/2015;0;7.00;1;5.00;" |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI3** | |
| **Descrição** | Verifica se a inclusão do registro de venda é inválida, colocando uma data inválida |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "1;37413815869;32/05/2015;0;7.00;1;5.00;" |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI4** | |
| **Descrição** | Verifica se a inclusão do registro de venda é inválida, colocando uma data inválida |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "1;37413815869;05/13/2015;0;7.00;1;5.00;" |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI5** | |
| **Descrição** | Verifica se a inclusão do registro de venda é válida |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "1;37413815869;09/06/2010;0;7.00;1;5.00;" |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI6** | |
| **Descrição** | Verifica se a inclusão do registro de venda é válida |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "1;37413815869;09/06/2016;0;7.00;1;5.00;" |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI7** | |
| **Descrição** | Verifica se a inclusão do registro de venda é inválida, colocando uma data inválida |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "1;37413815869;30/02/2015;0;7.00;1;5.00;" |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI8** | |
| **Descrição** | Verifica se a inclusão do registro de venda é válida |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "1;37413815869;09/06/2015;0;7.00;1;5.00;" |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI9** | |
| **Descrição** | Verifica se a inclusão do registro de venda é inválida, colocando as quantidades de itens em menor número do que o número de produtos |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "1;37413815869;09/06/2015;" |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI10** | |
| **Descrição** | Verifica se a inclusão do registro de venda é inválida, colocando uma quantidade vendida igual a 0 |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "1;37413815869;09/06/2015;0;7.00;1;5.00;2;2.00;3;7.00;4;" |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI11** | |
| **Descrição** | Verifica se a inclusão do registro de venda é válida |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "1;37413815869;09/06/2015;0;7.00;6;5.00;" |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI12** | |
| **Descrição** | Verifica se a inclusão do registro de venda é válida |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "1;37413815869;09/06/2015;-1;7.00;1;5.00;" |
| **Saída** | Sucesso |
| **Dependências** | Produto com código 6 esteja com estoque igual a 0 tem que ser criado |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI13** | |
| **Descrição** | Verifica se a inclusão do registro de venda é inválida |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "2;37413815869;00/06/2015;0;7.00;1;5.00;" |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI14** | |
| **Descrição** | Verifica se a inclusão do registro de venda é válida |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "2;37413815869;01/06/2015;0;7.00;1;5.00;" |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI15** | |
| **Descrição** | Verifica se a inclusão do registro de venda é inválida, colocando um mês inválido |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "3;37413815869;31/06/2015;0;7.00;1;5.00;" |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI16** | |
| **Descrição** | Verifica se a inclusão do registro de venda é válida |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "4;37413815869;10/00/2015;0;7.00;1;5.00;" |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI17** | |
| **Descrição** | Verifica se a inclusão do registro de venda é válida |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "5;37413815869;10/01/2015;0;7.00;1;5.00;" |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI18** | |
| **Descrição** | Verifica se a inclusão do registro de venda é válida |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "6;37413815869;10/12/2015;0;7.00;1;5.00;" |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI19** | |
| **Descrição** | Verifica se a inclusão do registro de venda é válida |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "7;37413815869;09/06/2015;0;7.00;" |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI20** | |
| **Descrição** | Verifica se a inclusão do registro de venda é válida |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "8;37413815869;09/06/2015;0;0.01;" |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI21** | |
| **Descrição** | Verifica se a inclusão do registro de venda é inválida, colocando um código de produto igual a vazio |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "Arroz;37413815869;09/06/2015;0;0.01;" |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI22** | |
| **Descrição** | Verifica se a inclusão do registro de venda é inválida, criando um mercado e associando um Registro Venda a um produto não cadastrado |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "9;37413815869;09/06/2015;0.01;" |
| **Saída** | Exceção |
| **Dependências** | Cria-se um Mercado, e tenta criar um RegistroVenda com um produto que não esteja cadastrado |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI23** | |
| **Descrição** | Verifica se a inclusão do registro de venda é inválida, tentando associar um registro venda a um cliente que não existe |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "10;37413815869;09/06/2015;7.00;5.00;" |
| **Saída** | Exceção |
| **Dependências** | Tenta-se criar um RegistroVenda sem que o Cliente esteja criado |

|  |  |
| --- | --- |
| **Caso de Teste CTRVI24** | |
| **Descrição** | Verifica se a inclusão do registro de venda é inválida, tentando associar um registro venda a um cliente que não existe |
| **Método(s)** | Public RegistroVenda(int numero, String data, int quantidadeItens, ArrayList<Double> quantidades, Cliente cliente, ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | "10;Arroz;37413234323;09/06/2015;7.00;5.00;" |
| **Saída** | Exceção |
| **Dependências** |  |

**Casos de Teste – Alteração de Registro de Vendas**

|  |  |
| --- | --- |
| **Caso de Teste CTRVA1** | |
| **Descrição** | Altera-se o número do produto e espera-se um erro |
| **Método(s)** | public void setNumero(int numero) throws RuntimeException |
| **Entrada** | -1 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA2** | |
| **Descrição** | Altera-se a data do produto e espera-se um erro |
| **Método(s)** | public void setData(String data) throws RuntimeException |
| **Entrada** | "32/05/2015" |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA3** | |
| **Descrição** | Altera-se a data do produto e espera-se um erro |
| **Método(s)** | public void setData(String data) throws RuntimeException |
| **Entrada** | "05/13/2015" |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA4** | |
| **Descrição** | Altera-se a data do produto e espera-se um erro |
| **Método(s)** | public void setData(String data) throws RuntimeException |
| **Entrada** | "09/06/2010;0" |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA5** | |
| **Descrição** | Altera-se a data do produto e espera-se um sucesso |
| **Método(s)** | public void setData(String data) throws RuntimeException |
| **Entrada** | "09/06/2016" |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA6** | |
| **Descrição** | Altera-se a data do produto e espera-se um erro |
| **Método(s)** | public void setData(String data) throws RuntimeException |
| **Entrada** | "30/02/2015" |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA7** | |
| **Descrição** | Altera-se a quantidade de itens e espera-se um erro |
| **Método(s)** | public void setQuantidadeItens(int quantidadeItens) throws RuntimeException |
| **Entrada** | 0 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA8** | |
| **Descrição** | Declara mais produtos do que quantidades vendidas e espera-se um erro |
| **Método(s)** | public void setQuantidades(ArrayList<Double> quantidades)throws RuntimeException  public void setProdutos(ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | quantidadedeItens = [1.00, 2.00, 7.00, 5.00]  Produtos = [0,1,2,3,4] |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA9** | |
| **Descrição** | Declara uma quantidade vendida igual a 0 e espera-se erro |
| **Método(s)** | public void setQuantidadeItens(int quantidadeItens) throws RuntimeException  public void setProdutos(ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | quantidadesdeItens = [0, 5.00]  Produtos = [0,1] |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA10** | |
| **Descrição** | Altera-se a data do produto e espera-se um erro |
| **Método(s)** | public void setData(String data) throws RuntimeException |
| **Entrada** | "00/06/2015" |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA11** | |
| **Descrição** | Altera-se a data do produto |
| **Método(s)** | public void setData(String data) throws RuntimeException |
| **Entrada** | "01/06/2015" |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA12** | |
| **Descrição** | Altera-se a data do produto |
| **Método(s)** | public void setData(String data) throws RuntimeException |
| **Entrada** | "31/06/2015" |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA13** | |
| **Descrição** | Altera-se a data do produto e espera-se um erro |
| **Método(s)** | public void setData(String data) throws RuntimeException |
| **Entrada** | "10/00/2015" |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA14** | |
| **Descrição** | Altera-se a data do produto |
| **Método(s)** | public void setData(String data) throws RuntimeException |
| **Entrada** | "10/01/2015" |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA15** | |
| **Descrição** | Altera-se a data do produto |
| **Método(s)** | public void setData(String data) throws RuntimeException |
| **Entrada** | "10/12/2015" |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA16** | |
| **Descrição** | Altera-se a quantidade de itens para um item apenas |
| **Método(s)** | public void setQuantidades(ArrayList<Double> quantidades)throws RuntimeException  public void setProdutos(ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | Quantidades = [7.00]  Produtos = [0] |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA17** | |
| **Descrição** | Altera-se a quantidade vendida para aquele determinado item |
| **Método(s)** | public void setQuantidades(ArrayList<Double> quantidades)throws RuntimeException  public void setProdutos(ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | Quantidades = [0.01]  Produtos = [0] |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA18** | |
| **Descrição** | Altera-se os produtos para ele não receber nenhum produto e assim espera-se um erro |
| **Método(s)** | public void setQuantidades(ArrayList<Double> quantidades)throws RuntimeException  public void setProdutos(ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | Quantidades = [0.01]  Produtos = [] |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTRVA19** | |
| **Descrição** | Altera-se a quantidade vendida para não receber nenhuma quantidade, e assim espera-se um erro |
| **Método(s)** | public void setQuantidades(ArrayList<Double> quantidades)throws RuntimeException  public void setProdutos(ArrayList<Produto> produtos) throws RuntimeException |
| **Entrada** | Quantidades = []  Produtos = [0] |
| **Saída** | Exceção |
| **Dependências** |  |

* + - 1. **Cálculo do Faturamento em um Determinado Mês**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Num** | **Classes Inválidas** | **Num** |
| Descrição | 1 <= Value <= 12 | CV49 | 1 > Value | CI51 |
| 12 < Value | CI52 |
| Caracteres Numéricos | CV50 | Caracteres Especiais e Letras | CI53 |
| Saída | Value = Nº Real | CV51 | Value = -1.0 | CI53 |

**Casos de Teste – Cálculo do Faturamento em um Determinado Mês**

|  |  |
| --- | --- |
| **Caso de Teste CTCFDM1** | |
| **Descrição** | Verifica o cálculo do faturamento em um determinado mês e espera-se um sucesso |
| **Método(s)** | public double calculaFaturamento(int mes) throws RuntimeException |
| **Entrada** | assertTrue(m.calculaFaturamento(06)==59.88); |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCFDM2** | |
| **Descrição** | Verifica o cálculo do faturamento em um determinado mês e espera-se um erro |
| **Método(s)** | public double calculaFaturamento(int mes) throws RuntimeException |
| **Entrada** | assertTrue(m.calculaFaturamento(00)==0); |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTCFDM3** | |
| **Descrição** | Verifica o cálculo do faturamento em um determinado mês e espera-se um erro |
| **Método(s)** | public double calculaFaturamento(int mes) throws RuntimeException |
| **Entrada** | assertTrue(m.calculaFaturamento(13)==0); |
| **Saída** | Exceção |
| **Dependências** |  |

* + - 1. **Alteração da Quantidade em Estoque dos Produtos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Num** | **Classes Inválidas** | **Num** |
| Código | Value >= 0 | CV52 | Value < 0 | CI54 |
| Caracteres Numéricos | CV53 | Caracteres Especiais e Letras | CI55 |
| Estoque | Value >= 0 | CV54 | Value < 0 | CI56 |
| Caracteres Numéricos | CV55 | Caracteres Especiais e Letras | CI57 |
| Saída |  |  | RuntimeException |  |

**Casos de Teste – Alteração da Quantidade em Estoque dos Produtos**

|  |  |
| --- | --- |
| **Caso de Teste CTAQEP1** | |
| **Descrição** | Altera-se a quantidade no estoque e espera-se um sucesso |
| **Método(s)** | public void alteraEstoque(int codigo, double estoque) throws RuntimeException |
| **Entrada** | 0,0 |
| **Saída** | Sucesso |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTAQEP2** | |
| **Descrição** | Altera-se a quantidade no estoque e espera-se um erro |
| **Método(s)** | public void alteraEstoque(int codigo, double estoque) throws RuntimeException |
| **Entrada** | -1,0 |
| **Saída** | Exceção |
| **Dependências** |  |

|  |  |
| --- | --- |
| **Caso de Teste CTAQEP3** | |
| **Descrição** | Altera-se a quantidade no estoque e espera-se um erro |
| **Método(s)** | public void alteraEstoque(int codigo, double estoque) throws RuntimeException |
| **Entrada** | 0,-1 |
| **Saída** | Exceção |
| **Dependências** |  |