Universidade Federal de São Carlos Tópicos Avançados A

Projeto Mercado

Plano de Teste

# Introdução

O programa deve gerenciar a venda de produtos a clientes, oferecendo as seguintes funcionalidades:

* Gerenciamento de produtos: inclusão e alteração. Para cada produto, os seguintes dados são armazenados: código (único), descrição, preço de compra, preço de venda, unidade, quantidade em estoque e status (ativo ou inativo).
* Gerenciamento de clientes: inclusão e alteração. Para cada cliente, os seguintes dados são armazenados: nome, CPF (u ́nico), endereço, telefone, e-mail e status (ativo ou inativo).

* Alteração da quantidade em estoque dos produtos, de acordo com o código do produto informado.

* Consulta de produtos por descrição.
* Registro de vendas de produtos para clientes, com os seguintes dados: número do registro de venda, CPF do cliente, data da venda, lista de produtos adquiridos e a quantidade vendida de cada item. Uma venda pode incluir mais de um produto, sendo que cada item vendido pode ter uma quantidade diferente. Somente produtos cadastrados podem compor os registros de vendas.

Cálculo do faturamento em um determinado mês. Para isso, o usuário deve informar o mês desejado.

# Equipe de Teste

|  |  |
| --- | --- |
| **Nome** | **Função** |
| **Thales Menato** | Programador e Tester |
| **Mateus Takata** | Programador e Tester |
| **Bruno Cintra** | Programador e Tester |
| **Cristiano Faustino** | Documentação e Planejamento de Testes |
| **Felipe Fantoni** | Tester |

# Abordagem

As atividades de teste no projeto Mercado consistem no teste de unidade, aplicando-se o Teste Funcional Sistemático (que utiliza-se do critério de Análise de Valor Limite, Particionamento de Equivalência e algumas outras definições para a definição de casos de teste), utilizando-se da ferramenta JUnit para facilitar a execução dos testes definidos. Para esse teste, método é considerado como a unidade.

Como diversos métodos implementados nas classes (principalmente as de Gerenciamento de classes básicas como Produto e Cliente) utilizam-se de diversos outros métodos, o Teste de Integração é feito conjuntamente (pensando apenas em Teste Funcional), pois todas as integrações entre métodos e classes vão ser testadas uma por uma.

Abaixo está a lista de métodos, a serem testados pelos critérios determinados acima, e suas respectivas descrições de funcionamento:

|  |  |  |
| --- | --- | --- |
| **Classe** | **Método** | **Descrição** |
| Cliente | public String **getNome**() | Método para retornar o Nome de um Cliente. |
| Cliente | public void **setNome**(String nome) | Altera o parâmetro Nome de um objeto da classe Cliente. |
| Cliente | public String **getCpf**() | Método para retornar o CPF de um Cliente. |
| Cliente | public void **setCpf**(String cpf) | Altera o parâmetro CPF de um objeto da classe Cliente. |
| Cliente | public String **getEndereco**() | Método para retornar o Endereço de um Cliente. |
| Cliente | public void **setEndereco**(String endereco) | Altera o parâmetro Endereço de um objeto da classe Cliente. |
| Cliente | public String **getTelefone**() | Método para retornar o Telefone de um Cliente. |
| Cliente | public void **setTelefone**(String telefone) | Altera o parâmetro Telefone de um objeto da classe Cliente. |
| Cliente | public String **getEmail**() | Método para retornar o Email de um Cliente. |
| Cliente | public void **setEmail**(String email) | Altera o parâmetro Email de um objeto da classe Cliente. |
| Cliente | public int **getStatus**() | Método para retornar o Status (Ativo, Inativo) de um Cliente. |
| Cliente | public void **setStatus**(int status) | Altera o parâmetro Status (Inativo, Ativo) de um objeto da classe Cliente. |
| Cliente | public void **alteraCliente**(String nome, String cpf, String endereco, String telefone, String email, int status) | Utiliza-se dos setters da classe Cliente para alterar dados de um cliente em específico. |
| Produto | public void **setCodigo**(int codigo) | Altera o parâmetro Codigo de um objeto da classe Produto. |
| Produto | public int **getCodigo**() | Método para retornar o Código de um Produto. |
| Produto | public void **setDescricao**(String descricao) | Altera o parâmetro Descrição de um objeto da classe Produto. |
| Produto | public String **getDescricao**() | Método para retornar a Descrição de um Produto. |
| Produto | public void **setPrecoCompra**(double precoCompra) | Altera o parâmetro Preço de Compra de um objeto da classe Produto. |
| Produto | public double **getPrecoCompra**() | Método para retornar o Preço de Compra de um Produto. |
| Produto | public void **setPrecoVenda**(double precoVenda) | Altera o parâmetro Preço de Venda de um objeto da classe Produto. |
| Produto | public double **getPrecoVenda**() | Método para retornar o Preço de Venda de um Produto. |
| Produto | public void **setUnidade**(String unidade) | Altera o parâmetro Unidade de um objeto da classe Produto. |
| Produto | public String **getUnidade**() | Método para retornar a Unidade de um Produto. |
| Produto | public void **setEstoque**(double estoque) | Altera o parâmetro Estoque de um objeto da classe Produto. |
| Produto | public double **getEstoque**() | Método para retornar o Estoque de um Produto. |
| Produto | public void **setStatus**(int status) | Altera o parâmetro Status (Ativo, Inativo) de um objeto da classe Produto. |
| Produto | public int **getStatus**() | Método para retornar o Status (Ativo, Inativo) de um Produto. |
| Produto | public void **alteraProduto**(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) | Utiliza-se dos setters da classe Produto para alterar dados de um produto em específico. |
| GerenciadorClientes | public static void **insereCliente**(String nome, String cpf, String endereco, String telefone, String email, int status) | Insere um novo Cliente na lista de clientes cadastrados. |
| GerenciadorClientes | private static boolean **validaCpfUnico**(String cpf) |  |
| GerenciadorClientes | public static Cliente **consultaClientePorCpf**(String cpf) |  |
| Mercado | private static double **calculaFaturamento**(int mes) | Calcula faturamento de um mes em especifico |
| Mercado | public ArrayList<Integer> **consultaProdutos** (String descricao) | Busca produtos usando a descrição como filtro. |
| GerenciadorProdutos | public static void **insereProduto**(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) | Insere um novo Produto na lista de produtos cadastrados. |
| GerenciadorProdutos | public static void **alteraEstoque**(int cod, double estoque) | Altera o estoque de um determinado produto |
| GerenciadorProdutos | public static Produto **consultaProdutoPorCodigo** (int codigo) | Faz a consulta de um produto com base em seu código. |
| GerenciadorProdutos | public static ArrayList<Integer> **consultaPorDescricao** (String descricao) | Busca produtos usando a descrição como filtro. |
| GerenciadorProdutos | private static boolean **validaCodigo**(int codigo) | Verifica se determinado código de produto já está cadastrado. |
| RegistroVenda | public **RegistroVenda**(String registroVenda) | Construtor da classe RegistroVenda |
| RegistroVenda | private void **setCodigoVenda**(String regCodigoVenda) | Altera o código do RegistroVenda |
| RegistroVenda | private void **setDataVenda**(String regDataVenda) | Altera a Data de venda |
| RegistroVenda | private void **setQuantidadesEProdutos**(String regProdutosQuantidades) | Altera as quantiades e produtos do RegistroVenda |
| RegistroVenda | private void **setCliente**(String regCpfCliente) | Altera o Cliente relacionado ao RegistroVenda |
| RegistroVenda | public ArrayList<Produto> **getProdutos**() | Método para retornar um array dos produtos do RegistroVenda |
| RegistroVenda | public ArrayList<Double> **getQuantidades**() | Método para retornar as quantidades de cada produto do RegistroVenda |
| RegistroVenda | public int **getQuantidadeItens**() | Método para retornar a quantidade total de itens do RegistroVenda |
| RegistroVenda | public int **getNumero**() | Método para retornar o número do RegistroVenda |
| RegistroVenda | public Date **getData**() | Método para retornar a data do RegistroVenda |
| RegistroVenda | public double **faturamento**() | Método para retornar o faturamento do RegistroVenda |
| GerenciadorRegistrosVenda | public static void **insereRegistroVenda** (String str) | Insere um novo RegistroVenda no array de registros |
| GerenciadorRegistrosVenda | public static RegistroVenda **buscaPorNumero** (int numero) | Busca e retorna um RegistroVenda com base em seu número |
| GerenciadorRegistrosVenda | public static ArrayList<RegistroVenda> **getRegistros**() | Retorna uma lista de todos RegistroVenda |
| GerenciadorRegistrosVenda | public static ArrayList<RegistroVenda> **getRegistroDoMes** (int mes) | Retorna uma lista de RegistroVenda feitos em um deteterminado mês |

# Critérios de Aprovação

Os critérios adotados no Teste Funcional consistem na aplicação do Particionamento de Equivalencia juntamente à Análise de Valor Limite e os demais critérios definidos no Teste Funcional Sistemático. Deseja-se alcançar 100% de cobertura para os métodos citados na seção de Abordagem.

# Cronograma

|  |  |  |
| --- | --- | --- |
| **Tarefa** | **Data Esperada** | **Data Limite** |
| **Planejamento dos Testes** | 01/05/2015 | 15/05/2015 |
| **Implementação dos Testes em JUnit** | 15/05/2015 | 22/05/2015 |
| **Execução e Análise dos Testes** | 22/05/2015 | 29/05/2015 |
| **Correções Baseadas nos Testes** | 29/05/2015 | 05/06/2015 |
| **Execução dos Testes na Implementação Corrigida** | 05/06/2015 | 12/06/2015 |

# Classes de Equivalência e Casos de Teste

* 1. Metodo **insereCliente** – Classe **GerenciadorClientes**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Nome | String alfabetica de no máximo 128 caracteres, com pelo menos 2 palavras, cada uma com no mínimo tamanho 2. | - Não ser alfabetica  - Ter mais de 128 caracteres.  - Menos de duas palavras.  - Palavras com tamanho menor que 2. |
| CPF | String, 11 Digitos, com digito verificador respeitando o procedimento definido em:  <http://www.geradorcpf.com/algoritmo_do_cpf.htm> | - Menos de 11 digitos  - Mais de 11 digitos  - Digito verificador não respeitar procedimento.  - Não pode ser repetido com outro Cliente cadastrado. |
| Endereço | String alfanumérica de no máximo 256 caracteres e no minimo 1 caracter. | - Não ser alfanumérico.  - Ter mais de 256 caracteres.  - Ter menos de 1 caractere |
| Telefone | (XX)XXXXXXXXX, com X sendo números | - Não seguir o padrão definido. |
| Email | String da forma [usuario@dominio.complemento](mailto:usuario@dominio.complemento)  Onde usuario e dominio sao strings que começam em letras e tem apenas letras e numeros  Complemento = .com ou .com.br | - Não seguir padrão.  - Ter usuário que não comece em letra.  - Ter domínio que não comece em letra.  - Usar complemento diferente dos pré-definidos.  - Ter domínio que não tenha apenas letras e numeros.  - Ter usuário que não tenha apenas letras e numeros. |
| Status | Ativo, Inativo | Qualquer coisa diferente dos dois status pré-definidos. |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 01-A1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Nome *“Robert Richards”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-B1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Nome *“Ro Ri”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-C1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Nome *“Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque pes”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-D1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Nome *“Robert Richards Jr”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-E1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Nome *“Ros Rim”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-F1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Nome *“Ro Rim”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-G1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Nome *“Ro Ri Do”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-H1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Nome *“Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque pe”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvA1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um com Nome igual *a uma variavel int i = 3.* Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de nome inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvB1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Nome *“Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque pesa”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de nome inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvC1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Nome *“Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque pesam”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de nome inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvD1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Nome *“Robert”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | RuntimeException de nome inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvE1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Nome *“”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | RuntimeException de nome inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvF1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Nome *“Robert R Silva”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | RuntimeException de nome inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvG1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um com Nome *“R R”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | RuntimeException de nome inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvH1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um com Nome *“Robert@ Silva”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | RuntimeException de nome inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvI1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um com Nome *“Robert1 Silva12”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | RuntimeException de nome inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvJ1** | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Nome *“Robert 123”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | RuntimeException de nome inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-A2** | |
| Descrição | Verifica se o Cpf inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Cpf “40114627487” (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-B2** | |
| Descrição | Verifica se o Cpf inserido não é repetido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Cpf “40114627487” (sem aspas). Demais parâmetros também com suas entradas válidas.  Insere um Cliente com Cpf “26597244311” (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | Dois objetos Cliente criados e salvos na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvA2** | |
| Descrição | Verifica se o Cpf inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Cpf “5555555510” (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de CPF inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvB2** | |
| Descrição | Verifica se o Cpf inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Cpf “555555555512” (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de CPF inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvC2** | |
| Descrição | Verifica se o Cpf inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Cpf “5555555555513” (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de CPF inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvD2** | |
| Descrição | Verifica se o Cpf inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Cpf “11111111111” (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de CPF inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvE2** | |
| Descrição | Verifica se o Cpf inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Cpf “48391291828” (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de CPF inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvE2** | |
| Descrição | Verifica se o Cpf inserido não é repetido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Cpf “40114627487” (sem aspas). Demais parâmetros também com suas entradas válidas.  Insere um Cliente com Cpf “40114627487” (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | No Primeiro Cliente ? Objeto Cliente criado e salvo na lista de Clientes  No segundo Cliente  RuntimeException de CPF inválido devido a um cliente ja estar registrado com aquele CPF. |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-A3** | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Endereço “Rua Rosetto 123” (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-B3** | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Endereço “R” (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-C3** | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Endereço *“tttttttttttttttttttttttttttttttttttttttttt 3333333333333 tfffffffffffffffffffff ddddddddddddddddddddddddddddddddddddddd sssssssssssssssssssssssssssssssssssssssssssssss 333333333333333 ssssssssssssssssssssssssssssssssssss 44444444444444444444 dfdddfdfdfd256”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-D3** | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Endereço *“tttttttttttttttttttttttttttttttttttttttttt 3333333333333 tfffffffffffffffffffff ddddddddddddddddddddddddddddddddddddddd sssssssssssssssssssssssssssssssssssssssssssssss 333333333333333 ssssssssssssssssssssssssssssssssssss 44444444444444444444 dfdddfdfdf255”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-E3** | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Endereço “Ru” (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvA3** | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Endereço *“tttttttttttttttttttttttttttttttttttttttttt 3333333333333 tfffffffffffffffffffff ddddddddddddddddddddddddddddddddddddddd sssssssssssssssssssssssssssssssssssssssssssssss 333333333333333 ssssssssssssssssssssssssssssssssssss 44444444444444444444 dfdddfdfdfdf257”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Endereço inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvB3** | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Endereço *“tttttttttttttttttttttttttttttttttttttttttt 3333333333333 tfffffffffffffffffffff ddddddddddddddddddddddddddddddddddddddd sssssssssssssssssssssssssssssssssssssssssssssss 333333333333333 ssssssssssssssssssssssssssssssssssss 44444444444444444444 dfdddfdfdfdfd258”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Endereço inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvC3** | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Endereço *“”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Endereço inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvE3** | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Endereço *“Rua X, $#@”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Endereço inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvF3** | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Endereço *“#”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Endereço inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-A4** | |
| Descrição | Verifica se o Telefone inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Telefone *“(19)999999999”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-A4** | |
| Descrição | Verifica se o Telefone inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Telefone *“(19)999999999”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvA4** | |
| Descrição | Verifica se o Telefone inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Telefone *“(19999999999”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Telefone inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvA4** | |
| Descrição | Verifica se o Telefone inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Telefone *“19999999999”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Telefone inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvB4** | |
| Descrição | Verifica se o Telefone inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Telefone *“()19999999999”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Telefone inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvC4** | |
| Descrição | Verifica se o Telefone inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Telefone *“(19)1999999999”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Telefone inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvD4** | |
| Descrição | Verifica se o Telefone inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Telefone *“”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Telefone inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvE4** | |
| Descrição | Verifica se o Telefone inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Telefone *“(19)#99999999”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Telefone inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-A5** | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Emaill *“algo@algo.com”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-B5** | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Emaill *“algo@algo.com.br”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-C5** | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Emaill *“algo11@al11go1.com”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-D5** | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Emaill *“a@a.com.br”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvA5** | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Emaill *“1algo@algo.com”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Email inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvB5** | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Emaill *“algo@1algo.com”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Email inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvC5** | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Emaill *“@algo.com”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Email inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvD5** | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Emaill *“algo@.com”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Email inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvE5** | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Emaill *“algo@algo.xyz”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Email inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvF5** | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Emaill *“algoalgo.xyz”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Email inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvG5** | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Emaill *“algo@@algo.xyz”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Email inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-A6** | |
| Descrição | Verifica se o Status inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Status Ativo, ou seja, *“1”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-B6** | |
| Descrição | Verifica se o Status inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Status Inativo, ou seja, *“0”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | Objeto Cliente criado e salvo na lista de Clientes |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvA6** | |
| Descrição | Verifica se o Status inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Status *“3”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Status inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 01-InvB6** | |
| Descrição | Verifica se o Status inserido é válido. |
| Método(s) | public static void insereCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Status *“-1”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Status inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

* 1. Metodo **alteraCliente** – Classe **Cliente**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Nome | String alfabetica de no máximo 128 caracteres, com pelo menos 2 palavras, cada uma com no mínimo tamanho 2. | - Não ser alfabetica  - Ter mais de 128 caracteres.  - Menos de duas palavras.  - Palavras com tamanho menor que 2. |
| CPF | String, 11 Digitos, com digito verificador respeitando o procedimento definido em:  <http://www.geradorcpf.com/algoritmo_do_cpf.htm> | - Menos de 11 digitos  - Mais de 11 digitos  - Digito verificador não respeitar procedimento.  - Não pode ser repetido com outro Cliente cadastrado. |
| Endereço | String alfanumérica de no máximo 256 caracteres e no minimo 1 caracter. | - Não ser alfanumérico.  - Ter mais de 256 caracteres.  - Ter menos de 1 caractere |
| Telefone | (XX)XXXXXXXXX, com X sendo números | - Não seguir o padrão definido. |
| Email | String da forma [usuario@dominio.complemento](mailto:usuario@dominio.complemento)  Onde usuario e dominio sao strings que começam em letras e tem apenas letras e numeros  Complemento = .com ou .com.br | - Não seguir padrão.  - Ter usuário que não comece em letra.  - Ter domínio que não comece em letra.  - Usar complemento diferente dos pré-definidos.  - Ter domínio que não tenha apenas letras e numeros.  - Ter usuário que não tenha apenas letras e numeros. |
| Status | Ativo, Inativo | Qualquer coisa diferente dos dois status pré-definidos. |

### Casos de Teste

Todos os casos de teste do método **insereCliente** (exceto os 01-InvE2 e 01-B2) aplicados em um determinado Cliente já cadastrado. Dependências incluem a função insereCliente. Saída é o Objeto ser alterado com sucesso, ou RuntimeException.

Dois casos específicos:

|  |  |
| --- | --- |
| **Caso de Teste 02-A2** | |
| Descrição | Verifica se o Cpf inserido já está cadastrado em outro Cliente |
| Método(s) | public void alteraCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Cpf “40114627487” (sem aspas). Demais parâmetros também com suas entradas válidas.  Altera o mesmo Cliente, colocando Cpf “40114627487” (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | No Insere  Objeto Cliente criado e salvo na lista de Clientes  Na Alteração  Cliente alterado com sucesso. |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus, insereCliente |

|  |  |
| --- | --- |
| **Caso de Teste 02-InvA2** | |
| Descrição | Verifica se o Cpf inserido já está cadastrado em outro Cliente |
| Método(s) | public void alteraCliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Insere um Cliente com Cpf “40114627487” (sem aspas). Demais parâmetros também com suas entradas válidas.  Insere um Cliente com Cpf “26597244311” (sem aspas). Demais parâmetros também com suas entradas válidas.  Altera o primeiro Cliente, colocando Cpf “26597244311” (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | No Primeiro Insere  Objeto Cliente criado e salvo na lista de Clientes  No Segundo Insere  Objeto Cliente criado e salvo na lista de Clientes  Na Alteração  RuntimeException de CPF inválido devido a um cliente ja estar registrado com aquele CPF (e não ser o próprio Cliente a ser alterado). |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus, insereCliente |

* 1. Metodo **getNome** – Classe **Cliente**

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 03-A** | |
| Descrição | Verifica se o getNome retorna o parametro Nome inserido |
| Método(s) |  |
| Entrada | Cria objeto Cliente c com Nome “Roberto Souza”.  Execute c.getNome |
| Saída | “Roberto Souza” (sem aspas). |
| Dependências | setNome |

* 1. Metodo **setNome** – Classe **Cliente**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **getCpf** – Classe **Cliente**

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 05-A** | |
| Descrição | Verifica se o getCpf retorna o parametro CPF inserido |
| Método(s) |  |
| Entrada | Cria objeto Cliente c com Cpf “26597244311”. Demais parametros válidos.  Execute c.getCpf |
| Saída | “26597244311” (sem aspas). |
| Dependências | setCpf |

* 1. Metodo **setCpf** – Classe **Cliente**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **getEndereco** – Classe **Cliente**

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 07-A** | |
| Descrição | Verifica se o getEndereco retorna o parametro Endereço inserido |
| Método(s) |  |
| Entrada | Cria objeto Cliente c com Endereço “Rua Tal 123”.  Execute c.getEndereco |
| Saída | “Rua Tal 123” (sem aspas). |
| Dependências | setEndereco |

* 1. Metodo **setEndereco** – Classe **Cliente**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **getTelefone** – Classe **Cliente**

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 09-A** | |
| Descrição | Verifica se o getTelefone retorna o parametro Telefone inserido |
| Método(s) |  |
| Entrada | Cria objeto Cliente c com Telefone “*(19)999999999*”.  Execute c.getTelefone |
| Saída | “*(19)999999999*” (sem aspas). |
| Dependências | setTelefone |

* 1. Metodo **setTelefone** – Classe **Cliente**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **getEmail** – Classe **Cliente**

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 11-A** | |
| Descrição | Verifica se o getEmail retorna o parametro Email inserido |
| Método(s) |  |
| Entrada | Cria objeto Cliente c com Email “*algo@algo.com*”.  Execute c.getEmail |
| Saída | “*algo@algo.com*” (sem aspas). |
| Dependências | setEmail |

|  |  |
| --- | --- |
| **Caso de Teste 11-B** | |
| Descrição | Verifica se o getEmail retorna o parametro Email inserido |
| Método(s) |  |
| Entrada | Cria objeto Cliente c com Email “*algo@algo.com.br*”.  Execute c.getEmail |
| Saída | “*algo@algo.com.br*” (sem aspas). |
| Dependências | setEmail |

* 1. Metodo **setEmail** – Classe **Cliente**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **getStatus** – Classe **Cliente**

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 13-A** | |
| Descrição | Verifica se o getStatus retorna o parametro Status inserido |
| Método(s) |  |
| Entrada | Cria objeto Cliente c com Status “*1*”, ou seja, Ativo.  Execute c.getStatus |
| Saída | “*1*” (sem aspas). |
| Dependências | setStatus |

|  |  |
| --- | --- |
| **Caso de Teste 13-B** | |
| Descrição | Verifica se o getStatus retorna o parametro Status inserido |
| Método(s) |  |
| Entrada | Cria objeto Cliente c com Status “0”, ou seja, Inativo.  Execute c.getStatus |
| Saída | “0” (sem aspas). |
| Dependências | setStatus |

* 1. Metodo **setStatus** – Classe **Cliente**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **getCodigo** – Classe **Produto**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **setCodigo** – Classe **Produto**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **getDescricao** – Classe **Produto**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **setDescricao** – Classe **Produto**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **getPrecoCompra** – Classe **Produto**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **setPrecoCompra** – Classe **Produto**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **getPrecoVenda** – Classe **Produto**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **setPrecoVenda** – Classe **Produto**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **getUnidade** – Classe **Produto**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **setUnidade** – Classe **Produto**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **getEstoque** – Classe **Produto**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **setEstoque** – Classe **Produto**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **getStatus** – Classe **Produto**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **setStatus** – Classe **Produto**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **alteraProduto** – Classe **Produto**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **validaCpfUnico** – Classe **GerenciadorClientes**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **consultaClientePorCpf** – Classe **GerenciadorClientes**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **calculaFaturamento** – Classe **Mercado**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **consultaProdutos** – Classe **Mercado**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **insereProduto** – Classe **GerenciadorProdutos**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **alteraEstoque** – Classe **GerenciadorProdutos**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **consultaProdutoPorCodigo** – Classe **GerenciadorProdutos**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **consultaPorDescricao** – Classe **GerenciadorProdutos**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **validaCodigo** – Classe **GerenciadorProdutos**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **listaProdutos** – Classe **GerenciadorProdutos**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **exibeProduto(1)** – Classe **GerenciadorProdutos**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **exibeProduto(2)** – Classe **GerenciadorProdutos**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **Construtor** – Classe **RegistroVenda**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Valor da moeda | Valor = 1 | Valor = 2 |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **setNumeroVenda** – Classe **RegistroVenda**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Número de registro | Deve ser inteiro positivo e único | <= 0 |
| Não inteiro |
| Não único |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 43-A** | |
| Descrição | Verifica se o número de venda é valido |
| Método(s) | setNumeroVenda(int) |
| Entrada | Insere 1 no número de venda |
| Saída | Número de registro aceito |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 43-InvA** | |
| Descrição | Verifica se o número de venda é valido |
| Método(s) | setNumeroVenda(int) |
| Entrada | Insere 0.1 no número de venda |
| Saída | Lança RuntimeException |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 43-InvB** | |
| Descrição | Verifica se o número de venda é valido |
| Método(s) | setNumeroVenda(int) |
| Entrada | Cria um RegistroVenda anterior que já tem o número=1, insere 1 no novo número de venda |
| Saída | Número de registro não ceito |
| Dependências |  |

* 1. Metodo **setDataVenda** – Classe **RegistroVenda**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Data de registro | Formato DD/MM/AAAA e válida no calendário | Fora do formato |
| Inválida no calendario |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 44-A** | |
| Descrição | Verifica se a data é válida |
| Método(s) | setDataVenda(String) |
| Entrada | String “01/01/2015” |
| Saída | Não lança exceção |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 44-InvA** | |
| Descrição | Verifica se a data é válida |
| Método(s) | setDataVenda(String) |
| Entrada | String “2015/01/01” |
| Saída | Lança exceção |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 44-InvB** | |
| Descrição | Verifica se a data é válida |
| Método(s) | setDataVenda(String) |
| Entrada | String “30/02/2015” |
| Saída | Lança exceção |
| Dependências |  |

* 1. Metodo **setQuantidadesEProdutos** – Classe **RegistroVenda**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Quantidade de Itens | Número de partições da string é par | Número de partições da string é impar |
| Quantidade de cada item | Array de número reais positivos | Um dos números é negativo |
| Um dos números zero |
| Códigos dos produtos | Array de números inteiros positivos | Um dos números é negativo |
| Um dos número é zero |
| Um dos números é real |
| Produto[i] vs. Quantidade[i] | Deve haver em estoque cada quantidade de determinado produto | Não há no estoque |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 45-A1** | |
| Descrição | Verifica se a quantidade itens é valida |
| Método(s) | setQuantidadesEProdutos(String) |
| Entrada | Entrada “01; 10; 02; 10” |
| Saída | Não lança exceção |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 45-Inv1** | |
| Descrição | Verifica se a quantidade itens é valida |
| Método(s) | setQuantidadesEProdutos(String) |
| Entrada | Entrada “01; 10; 02” |
| Saída | Lança exceção |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 45-A2** | |
| Descrição | Verifica se a quantidade de cada item é valida |
| Método(s) | setQuantidadesEProdutos(String) |
| Entrada | Entrada “01; 10; 02; 10” |
| Saída | Não lança exceção |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 45-InvA2** | |
| Descrição | Verifica se a quantidade de cada item é valida |
| Método(s) | setQuantidadesEProdutos(String) |
| Entrada | Entrada “01; -1; 02; 10” |
| Saída | Lança exceção |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 45-InvB2** | |
| Descrição | Verifica se a quantidade de cada item é valida |
| Método(s) | setQuantidadesEProdutos(String) |
| Entrada | Entrada “01; 0; 02; 10” |
| Saída | Lança exceção |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 45-A3** | |
| Descrição | Verifica se o codigo do produto é valido |
| Método(s) | setQuantidadesEProdutos(String) |
| Entrada | Entrada “01; 0; 02; 10” |
| Saída | Não lança exceção |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 45-InvA3** | |
| Descrição | Verifica se o codigo do produto é valido |
| Método(s) | setQuantidadesEProdutos(String) |
| Entrada | Entrada “-1; 10; 02; 10” |
| Saída | Não lança exceção |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 45-InvB3** | |
| Descrição | Verifica se o codigo do produto é valido |
| Método(s) | setQuantidadesEProdutos(String) |
| Entrada | Entrada “0; 10; 02; 10” |
| Saída | Não lança exceção |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 45-InvC3** | |
| Descrição | Verifica se o codigo do produto é valido |
| Método(s) | setQuantidadesEProdutos(String) |
| Entrada | Entrada “0.1; 10; 02; 10” |
| Saída | Não lança exceção |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 45-A4** | |
| Descrição | Verifica se o produto X tem na quantidade Y em estoque |
| Método(s) | setQuantidadesEProdutos(String) |
| Entrada | Produto 01 possui em estoque 100, então entra-se com setQuantidadesEProdutos(“01; 5; 02; 10”) |
| Saída | Produto 01 tem seu estoque reduzido de 100 para 95 |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 45-Inv4** | |
| Descrição | Verifica se o produto X tem na quantidade Y em estoque |
| Método(s) | setQuantidadesEProdutos(String) |
| Entrada | Produto 01 possui em estoque 10, então entra-se com setQuantidadesEProdutos(“01; 50; 02; 10”) |
| Saída | Lança exceção |
| Dependências |  |

* 1. Metodo **setCliente** – Classe **RegistroVenda**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| **CPF do cliente** | O cliente deve estar no cadastro | Cliente não cadastrado |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 46-A** | |
| Descrição | Verifica se o cliente com CPF entrado existe |
| Método(s) | setCliente(String) |
| Entrada | Cadastra cliente com CPF x, entra com setCliente(x); |
| Saída | Não lança exceção |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 46-Inv** | |
| Descrição | Verifica se o cliente com CPF entrado existe |
| Método(s) | setCliente(String) |
| Entrada | Não cliente com CPF x, entra com setCliente(x); |
| Saída | Lança exceção |
| Dependências |  |

* 1. Metodo **getProdutos** – Classe **RegistroVenda**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| produtos != null | Sim (retorna array produtos) | Não (lança exceção) |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 47-A** | |
| Descrição | Verifica se algum produto foi cadastrado no registro |
| Método(s) | getProdutos(), setQuantidadesEProdutos(String) |
| Entrada | Usa-se o método setProdutosEQuantidades(String) para cadastrar produtos e quantidades e então usa-se getProdutos() para retorna-los |
| Saída | Array de produtos cadastrados |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 47-Inv** | |
| Descrição | Verifica se algum produto foi cadastrado no registro |
| Método(s) | getProdutos(), setQuantidadesEProdutos(String) |
| Entrada | Usar getProdutos() antes que algum produto tenha sido registrado na venda |
| Saída | Lança exceção |
| Dependências |  |

* 1. Metodo **getQuantidades** – Classe **RegistroVenda**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| quantidades != null | Sim (retorna array quantidades) | Não (lança exceção) |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 48-A** | |
| Descrição | Verificar se as quantidade de cada produto já foi registrada |
| Método(s) | getQuantidades(), setQuantidadesEProdutos(String) |
| Entrada | Usa-se o método setProdutosEQuantidades(String) para cadastrar produtos e quantidades e então usa-se getQuantidades() para retorna-los |
| Saída | Array de quantidades |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 48-Inv** | |
| Descrição | Verificar se as quantidade de cada produto já foi registrada |
| Método(s) | getQuantidades() |
| Entrada | Usar getQuantidades() antes que algum produto tenha sido registrado na venda |
| Saída | Lança exceção |
| Dependências |  |

* 1. Metodo **getQuantidadeItens** – Classe **RegistroVenda**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| quantidadeItens já foi definida | Sim (retorna quantidadeItens) | Não (Lança exceção) |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 49-A** | |
| Descrição | Verifica se a quantidade total de itens já foi registrada |
| Método(s) | getQuantidadeItens(), setQuantidadesEProdutos(String) |
| Entrada | Usar setQuantidadesEProdutos com um valor valido em seguida usar getQuantidadeItens() |
| Saída | Retorna quantidade total de itens |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 49-Inv** | |
| Descrição | Verifica se a quantidade total de itens já foi registrada |
| Método(s) | getQuantidadeItens() |
| Entrada | Usar getQuantidadeItens() antes de colocar um setQuantidadesEProdutos válido |
| Saída | Exceção |
| Dependências |  |

* 1. Metodo **getNumero** – Classe **RegistroVenda**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Número ja foi definido | Sim (retorna numero) | Não (lança exceção) |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 50-A** | |
| Descrição | Verificar se o número do resistro venda já foi registrado |
| Método(s) | setNumeroVenda(String), getNumero() |
| Entrada | Usar setNumeroVenda(“10”), depois usar getNumero() |
| Saída | Retorna 10 |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 50-Inv** | |
| Descrição | Verificar se o número do resistro venda já foi registrado |
| Método(s) | getNumero() |
| Entrada | Usar getNumero() sem usra setNumeroVenda(String) antes. |
| Saída | Exceção |
| Dependências |  |

* 1. Metodo **getData** – Classe **RegistroVenda**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Data já foi definida | Sim (retorna data) | Não (retorna null) |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 51-A** | |
| Descrição | Verificar se a data foi registrada |
| Método(s) | setData(), getData() |
| Entrada | SetData(“01/01/2015”), getData() |
| Saída | Retorna a data 01/01/2015 |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 51-Inv** | |
| Descrição | Verifica se a data foi registrada |
| Método(s) | getData() |
| Entrada | GetData() sem ter usado setData(String) |
| Saída | Exceção |
| Dependências |  |

* 1. Metodo **faturamento** – Classe **RegistroVenda**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| RegistroVenda já registrado | Sim (retorna o valor correto) | Não (retorna o valor incorreto) |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 52-A** | |
| Descrição | Verificar se o faturamento retorna o valor certo |
| Método(s) | RegistroVenda(String), faturamento() |
| Entrada | Inserir RegistroVenda válido, usar faturamento() para retornar faturamento |
| Saída | Faturamento = lucro da venda |
| Dependências |  |

* 1. Metodo **insereRegistroVenda** – Classe **GerenciadorRegistrosVenda**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
|  |  |  |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **buscaPorNumero** – Classe **GerenciadorRegistrosVenda**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
| Numero do RegistroVenda a ser buscado | Está no array de RegistroVenda |  |
| Não está no array |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 54-A** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

|  |  |
| --- | --- |
| **Caso de Teste 54-B** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **printRegistro** – Classe **GerenciadorRegistrosVenda**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
|  |  |  |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **getRegistros** – Classe **GerenciadorRegistrosVenda**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
|  |  |  |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |

* 1. Metodo **getRegistroDoMes** – Classe **GerenciadorRegistrosVenda**

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| **Condição de Entrada** | **Classes Válidas** | **Classes Inválidas** |
|  |  |  |
|  |  |
|  |  |

### Casos de Teste

|  |  |
| --- | --- |
| **Caso de Teste 0001** | |
| Descrição |  |
| Método(s) |  |
| Entrada |  |
| Saída |  |
| Dependências |  |