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 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-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 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-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 *“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-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 *“”* (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 *“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-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 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-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 *“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-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 *“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-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 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-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 *“()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-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 *“(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-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 *“”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Telefone inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 01-InvF4 | |
| 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) | public String getNome() |
| 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 |
| 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. |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 04-A | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public void setNome(String nome) |
| Entrada | Cria objeto Cliente c  Seta Nome “Roberto Souza”. |
| Saída | Nome do Cliente c salvo como “Roberto Souza” |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 04-B | |
| 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 | Cria objeto Cliente c  Seta Nome *“Ro Ri”* (sem aspas). |
| Saída | Nome do Cliente c salvo como o valor fornecido na entrada do metodo setNome |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 04-C | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public void setNome(String nome) |
| Entrada | Cria objeto Cliente c  Seta Nome *“Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque pes”* (sem aspas). |
| Saída | Nome do Cliente c salvo como o valor fornecido na entrada do metodo setNome |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 04-D | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public void setNome(String nome) |
| Entrada | Cria objeto Cliente c  Seta Nome *“Robert Richards Jr”* (sem aspas). |
| Saída | Nome do Cliente c salvo como o valor fornecido na entrada do metodo setNome |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 04-E | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public void setNome(String nome) |
| Entrada | Cria objeto Cliente c  Seta Nome *“Ros Rim”* (sem aspas). |
| Saída | Nome do Cliente c salvo como o valor fornecido na entrada do metodo setNome |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 04-F | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public void setNome(String nome) |
| Entrada | Cria objeto Cliente c  Seta Nome *“Ro Rim”* (sem aspas). |
| Saída | Nome do Cliente c salvo como o valor fornecido na entrada do metodo setNome |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 04-G | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public void setNome(String nome) |
| Entrada | Cria objeto Cliente c  Seta Nome *“Ro Ri Do”* (sem aspas). |
| Saída | Nome do Cliente c salvo como o valor fornecido na entrada do metodo setNome |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 04-H | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public void setNome(String nome) |
| Entrada | Cria objeto Cliente c  Seta Nome *“Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque pe”* (sem aspas). |
| Saída | Nome do Cliente c salvo como o valor fornecido na entrada do metodo setNome |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 04-InvB | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public void setNome(String nome) |
| Entrada | Cria objeto Cliente c  Seta Nome *“Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque pesa”* (sem aspas). |
| Saída | RuntimeException de nome inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 04-InvC | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public void setNome(String nome) |
| Entrada | Cria objeto Cliente c  Seta Nome *“Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque pesam”* (sem aspas). |
| Saída | RuntimeException de nome inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 04-InvD | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public void setNome(String nome) |
| Entrada | Cria objeto Cliente c  Seta Nome *“Robert”* (sem aspas). |
| Saída | RuntimeException de nome inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 04-InvE | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public void setNome(String nome) |
| Entrada | Cria objeto Cliente c  Seta Nome *“”* (sem aspas). |
| Saída | RuntimeException de nome inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 04-InvF | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public void setNome(String nome) |
| Entrada | Cria objeto Cliente c  Seta Nome *“Robert R Silva”* (sem aspas). |
| Saída | RuntimeException de nome inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 04-InvG | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public void setNome(String nome) |
| Entrada | Cria objeto Cliente c  Seta Nome *“R R”* (sem aspas). |
| Saída | RuntimeException de nome inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 04-InvH | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public void setNome(String nome) |
| Entrada | Cria objeto Cliente c  Seta Nome *“Robert@ Silva”* (sem aspas). |
| Saída | RuntimeException de nome inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 04-InvI | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public void setNome(String nome) |
| Entrada | Cria objeto Cliente c  Seta Nome *“Robert1 Silva12”* (sem aspas). |
| Saída | RuntimeException de nome inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 04-InvJ | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public void setNome(String nome) |
| Entrada | Cria objeto Cliente c  Seta Nome *“Robert 123”* (sem aspas). |
| Saída | RuntimeException de nome inválido |
| 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) | public String getCpf() |
| Entrada | Cria objeto Cliente c com Cpf “26597244311”.  Execute c.getCpf |
| Saída | “26597244311” (sem aspas). |
| Dependências | setCpf |

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

### Classes de Equivalência

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

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 06-A | |
| Descrição | Verifica se o Cpf inserido é válido. |
| Método(s) | public void setCpf(String cpf) |
| Entrada | Cria objeto Cliente c  Seta Cpf“40114627487” (sem aspas). |
| Saída | Cpf do Cliente c salvo como o valor fornecido na entrada do metodo setCpf |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 06-InvA | |
| Descrição | Verifica se o Cpf inserido é válido. |
| Método(s) | public void setCpf(String cpf) |
| Entrada | Cria objeto Cliente c  Seta Cpf“5555555510” (sem aspas). |
| Saída | RuntimeException de CPF inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 06-InvB | |
| Descrição | Verifica se o Cpf inserido é válido. |
| Método(s) | public void setCpf(String cpf) |
| Entrada | Cria objeto Cliente c  Seta Cpf“555555555512” (sem aspas). |
| Saída | RuntimeException de CPF inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 06-InvC | |
| Descrição | Verifica se o Cpf inserido é válido. |
| Método(s) | public void setCpf(String cpf) |
| Entrada | Cria objeto Cliente c  Seta Cpf “5555555555513” (sem aspas). |
| Saída | RuntimeException de CPF inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 06-InvD | |
| Descrição | Verifica se o Cpf inserido é válido. |
| Método(s) | public void setCpf(String cpf) |
| Entrada | Cria objeto Cliente c  Seta Cpf“11111111111” (sem aspas). |
| Saída | RuntimeException de CPF inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 06-InvE | |
| Descrição | Verifica se o Cpf inserido é válido. |
| Método(s) | public void setCpf(String cpf) |
| Entrada | Cria objeto Cliente c  Seta Cpf“48391291828” (sem aspas). |
| Saída | RuntimeException de CPF inválido |
| 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) | public String getEndereco() |
| 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 |
| 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 |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 08-A | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public void setEndereco(String endereco) |
| Entrada | Cria objeto Cliente c  Seta Endereço“Rua Rosetto 123” (sem aspas). |
| Saída | Endereco do Cliente c salvo como o valor fornecido na entrada do metodo setEndereco |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 08-B | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public void setEndereco(String endereco) |
| Entrada | Cria objeto Cliente c  Seta Endereço“R” (sem aspas). |
| Saída | Endereco do Cliente c salvo como o valor fornecido na entrada do metodo setEndereco |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 08-C | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public void setEndereco(String endereco) |
| Entrada | Cria objeto Cliente c  Seta Endereço *“tttttttttttttttttttttttttttttttttttttttttt 3333333333333 tfffffffffffffffffffff ddddddddddddddddddddddddddddddddddddddd sssssssssssssssssssssssssssssssssssssssssssssss 333333333333333 ssssssssssssssssssssssssssssssssssss 44444444444444444444 dfdddfdfdfd256”* (sem aspas). |
| Saída | Endereco do Cliente c salvo como o valor fornecido na entrada do metodo setEndereco |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 08-D | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public void setEndereco(String endereco) |
| Entrada | Cria objeto Cliente c  Seta Endereço *“tttttttttttttttttttttttttttttttttttttttttt 3333333333333 tfffffffffffffffffffff ddddddddddddddddddddddddddddddddddddddd sssssssssssssssssssssssssssssssssssssssssssssss 333333333333333 ssssssssssssssssssssssssssssssssssss 44444444444444444444 dfdddfdfdf255”* (sem aspas). |
| Saída | Endereco do Cliente c salvo como o valor fornecido na entrada do metodo setEndereco |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 08-E | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public void setEndereco(String endereco) |
| Entrada | Cria objeto Cliente c  Seta Endereço“Ru” (sem aspas). |
| Saída | Endereco do Cliente c salvo como o valor fornecido na entrada do metodo setEndereco |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 08-InvA | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public void setEndereco(String endereco) |
| Entrada | Cria objeto Cliente c  Seta Endereço *“tttttttttttttttttttttttttttttttttttttttttt 3333333333333 tfffffffffffffffffffff ddddddddddddddddddddddddddddddddddddddd sssssssssssssssssssssssssssssssssssssssssssssss 333333333333333 ssssssssssssssssssssssssssssssssssss 44444444444444444444 dfdddfdfdfdf257”* (sem aspas). |
| Saída | RuntimeException de Endereço inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 08-InvB | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public void setEndereco(String endereco) |
| Entrada | Cria objeto Cliente c  Seta Endereço *“tttttttttttttttttttttttttttttttttttttttttt 3333333333333 tfffffffffffffffffffff ddddddddddddddddddddddddddddddddddddddd sssssssssssssssssssssssssssssssssssssssssssssss 333333333333333 ssssssssssssssssssssssssssssssssssss 44444444444444444444 dfdddfdfdfdfd258”* (sem aspas). |
| Saída | RuntimeException de Endereço inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 08-InvC | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public void setEndereco(String endereco) |
| Entrada | Cria objeto Cliente c  Seta Endereço *“”* (sem aspas). |
| Saída | RuntimeException de Endereço inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 08-InvD | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public void setEndereco(String endereco) |
| Entrada | Cria objeto Cliente c  Seta Endereço *“Rua X, $#@”* (sem aspas) |
| Saída | RuntimeException de Endereço inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 08-InvE | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public void setEndereco(String endereco) |
| Entrada | Cria objeto Cliente c  Seta Endereço *“#”* (sem aspas). |
| Saída | RuntimeException de Endereço inválido |
| 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) | public String getTelefone() |
| 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 |
| Telefone | (XX)XXXXXXXXX, com X sendo números | - Não seguir o padrão definido. |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 10-A | |
| Descrição | Verifica se o Telefone inserido é válido. |
| Método(s) | public void setTelefone(String telefone) |
| Entrada | Cria objeto Cliente c  Seta Telefone *“(19)999999999”* (sem aspas). |
| Saída | Telefone do Cliente c salvo como o valor fornecido na entrada do metodo setTelefone |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 10-InvA | |
| Descrição | Verifica se o Telefone inserido é válido. |
| Método(s) | public void setTelefone(String telefone) |
| Entrada | Cria objeto Cliente c  Seta Telefone *“(19999999999”* (sem aspas). |
| Saída | RuntimeException de Telefone inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 10-InvB | |
| Descrição | Verifica se o Telefone inserido é válido. |
| Método(s) | public void setTelefone(String telefone) |
| Entrada | Cria objeto Cliente c  Seta Telefone *“19999999999”* (sem aspas). |
| Saída | RuntimeException de Telefone inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 10-InvC | |
| Descrição | Verifica se o Telefone inserido é válido. |
| Método(s) | public void setTelefone(String telefone) |
| Entrada | Cria objeto Cliente c  Seta Telefone *“()19999999999”* (sem aspas). |
| Saída | RuntimeException de Telefone inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 10-InvD | |
| Descrição | Verifica se o Telefone inserido é válido. |
| Método(s) | public void setTelefone(String telefone) |
| Entrada | Cria objeto Cliente c  Seta Telefone *“(19)1999999999”* (sem aspas). |
| Saída | RuntimeException de Telefone inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 10-InvE | |
| Descrição | Verifica se o Telefone inserido é válido. |
| Método(s) | public void setTelefone(String telefone) |
| Entrada | Cria objeto Cliente c  Seta Telefone *“”* (sem aspas). |
| Saída | RuntimeException de Telefone inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 10-InvF | |
| Descrição | Verifica se o Telefone inserido é válido. |
| Método(s) | public void setTelefone(String telefone) |
| Entrada | Cria objeto Cliente c  Seta Telefone *“(19)#99999999”* (sem aspas). |
| Saída | RuntimeException de Telefone inválido |
| 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) | public String getEmail() |
| 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) | public String getEmail() |
| 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 |
| 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. |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 12-A | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public void setEmail(String email) |
| Entrada | Cria objeto Cliente c  Seta Email *“algo@algo.com”* (sem aspas). |
| Saída | Email do Cliente c salvo como o valor fornecido na entrada do metodo setEmail |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 12-B | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public void setEmail(String email) |
| Entrada | Cria objeto Cliente c  Seta Email *“algo@algo.com.br”* (sem aspas). |
| Saída | Email do Cliente c salvo como o valor fornecido na entrada do metodo setEmail |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 12-C | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public void setEmail(String email) |
| Entrada | Cria objeto Cliente c  Seta Email *“algo11@al11go1.com”* (sem aspas). |
| Saída | Email do Cliente c salvo como o valor fornecido na entrada do metodo setEmail |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 12-D | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public void setEmail(String email) |
| Entrada | Cria objeto Cliente c  Seta Email *“a@a.com.br”* (sem aspas). |
| Saída | Email do Cliente c salvo como o valor fornecido na entrada do metodo setEmail |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 12-InvA | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public void setEmail(String email) |
| Entrada | Cria objeto Cliente c  Seta Email *“1algo@algo.com”* (sem aspas). |
| Saída | RuntimeException de Email inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 12-InvB | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public void setEmail(String email) |
| Entrada | Cria objeto Cliente c  Seta Email *“algo@1algo.com”* (sem aspas). |
| Saída | RuntimeException de Email inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 12-InvC | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public void setEmail(String email) |
| Entrada | Cria objeto Cliente c  Seta Email *“@algo.com”* (sem aspas). Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de Email inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 12-InvD | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public void setEmail(String email) |
| Entrada | Cria objeto Cliente c  Seta Email *“algo@.com”* (sem aspas). |
| Saída | RuntimeException de Email inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 12-InvE | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public void setEmail(String email) |
| Entrada | Cria objeto Cliente c  Seta Email *“algo@algo.xyz”* (sem aspas). |
| Saída | RuntimeException de Email inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 12-InvF | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public void setEmail(String email) |
| Entrada | Cria objeto Cliente c  Seta Email *“algoalgo.xyz”* (sem aspas). |
| Saída | RuntimeException de Email inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 12-InvG | |
| Descrição | Verifica se o Email inserido é válido. |
| Método(s) | public void setEmail(String email) |
| Entrada | Cria objeto Cliente c  Seta Email *“algo@@algo.xyz”* (sem aspas). |
| Saída | RuntimeException de Email inválido |
| 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) | public int getStatus() |
| 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) | public int getStatus() |
| 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 |
| Status | Ativo, Inativo | Qualquer coisa diferente dos dois status pré-definidos. |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 14-A | |
| Descrição | Verifica se o Status inserido é válido. |
| Método(s) | public void setStatus(int status) |
| Entrada | Cria objeto Cliente c  Seta StatusAtivo, ou seja, *“1”* (sem aspas). |
| Saída | Status do Cliente c salvo como o valor fornecido na entrada do metodo setStatus |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 14-B | |
| Descrição | Verifica se o Status inserido é válido. |
| Método(s) | public void setStatus(int status) |
| Entrada | Cria objeto Cliente c  Seta StatusInativo, ou seja, *“0”* (sem aspas). |
| Saída | Status do Cliente c salvo como o valor fornecido na entrada do metodo setStatus |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 14-InvA | |
| Descrição | Verifica se o Status inserido é válido. |
| Método(s) | public void setStatus(int status) |
| Entrada | Cria objeto Cliente c  Seta Status *“3”* (sem aspas). |
| Saída | RuntimeException de Status inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 14-InvB | |
| Descrição | Verifica se o Status inserido é válido. |
| Método(s) | public void setStatus(int status) |
| Entrada | Cria objeto Cliente c  Seta Status *“-1”* (sem aspas). |
| Saída | RuntimeException de Status inválido |
| Dependências |  |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| Codigo | Número inteiro positivo  Não pode ser repetido | - Codigo <=0  - Código já existente em outro Produto |
| Descrição | String, minimo 3 caracteres, máximo 64 | - Tamanho menor que 3  - Tamanho maior que 64 |
| Preço de Compra | Número real positivo | - Número negativo  - Zero |
| Preço de Venda | Número real positivo, maior que o Preço de Compra | - Número negativo  - Número maior que o Preço de Compra.  - Zero |
| Unidade | Alfabético, tamanho = 2 | - Sem ser alfabetico  - Sem ter tamanho = 2 |
| Estoque | Número real positivo (incluindo zero) | - Número negativo |
| Status | Ativo, Inativo  Inativo apenas se estoque de produto for zero. | - Qualquer coisa diferente dos dois status pré-definidos.  - Inativar produto com estoque maior que zero. |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 15-A1 | |
| Descrição | Verifica se o código inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Código = 1, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-B1 | |
| Descrição | Verifica se o código inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Código = 10, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-C1 | |
| Descrição | Verifica se o código não está incluido em outro Produto |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Código = 10, demais parametros válidos.  Insere Produto com Código = 11, demais parametros válidos. |
| Saída | Produto 10 criado e inserido na lista de Produtos  Produto 11 criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvA1 | |
| Descrição | Verifica se o código inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Código = 0, demais parametros válidos. |
| Saída | RuntimeException de Código Inválido |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvB1 | |
| Descrição | Verifica se o código inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Código = -1, demais parametros válidos. |
| Saída | RuntimeException de Código Inválido |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvC1 | |
| Descrição | Verifica se o código não está incluido em outro Produto. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Código = 10, demais parametros válidos.  Insere Produto com Código = 10, demais parametros válidos. |
| Saída | Primeiro produto criado e inserido na lista de Produtos  Segundo Produto --> RuntimeException de Código Inválido, pois é repetido. |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-A2 | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Descricao = “Produto1”, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-B2 | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Descricao = “Uva”, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-C2 | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Descricao = “AUva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1U64”, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-D2 | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Descricao = “AUva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva163”, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-E2 | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Descricao = “AUv4”, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvA2 | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Descricao = “AUva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uv65”, demais parametros válidos. |
| Saída | RuntimeException de Descrição Inválida |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvB2 | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Descricao = “Uv”, demais parametros válidos. |
| Saída | RuntimeException de Descrição Inválida |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvC2 | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Descricao = “”, demais parametros válidos. |
| Saída | RuntimeException de Descrição Inválida |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvD2 | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Descricao = “Uv@ Uva1”, demais parametros válidos. |
| Saída | RuntimeException de Descrição Inválida |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-A3 | |
| Descrição | Verifica se o preço de compra inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com PrecoCompra = 1.00, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-B3 | |
| Descrição | Verifica se o preço de compra inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com PrecoCompra = 4.57, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-C3 | |
| Descrição | Verifica se o preço de compra inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com PrecoCompra = 0.01, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvA3 | |
| Descrição | Verifica se o preço de compra inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com PrecoCompra = 0.00, demais parametros válidos. |
| Saída | Runtime Exception de PrecoCompra |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |
| Caso de Teste 15-InvB3 | |
| Descrição | Verifica se o preço de compra inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com PrecoCompra = - 0.01, demais parametros válidos. |
| Saída | Runtime Exception de PrecoCompra |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvC3 | |
| Descrição | Verifica se o preço de compra inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com PrecoCompra = - 3.00, demais parametros válidos. |
| Saída | Runtime Exception de PrecoCompra |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-A4 | |
| Descrição | Verifica se o preço de venda inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com PrecoVenda = 1.00 e PrecoCompra=0.99, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-B4 | |
| Descrição | Verifica se o preço de venda inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com PrecoVenda = 3.12 e PrecoCompra = 2.61, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-C4 | |
| Descrição | Verifica se o preço de venda inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com PrecoVenda = 0.02 e PrecoCompra=0.01, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvA4 | |
| Descrição | Verifica se o preço de venda inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com PrecoVenda = 0.00, demais parametros válidos. |
| Saída | Runtime Exception de PrecoVenda |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvB4 | |
| Descrição | Verifica se o preço de venda inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com PrecoVenda = - 0.01, demais parametros válidos. |
| Saída | Runtime Exception de PrecoVenda |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvC4 | |
| Descrição | Verifica se o preço de venda inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com PrecoVenda = - 3.00, demais parametros válidos. |
| Saída | Runtime Exception de PrecoVenda |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvD4 | |
| Descrição | Verifica se o preço de venda inserido é maior que preço de compra |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com PrecoVenda = 0.02 e PrecoCompra = 0.02, demais parametros válidos. |
| Saída | Runtime Exception de PrecoVenda |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvE4 | |
| Descrição | Verifica se o preço de venda inserido é maior que preço de compra |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com PrecoVenda = 1.02 e PrecoCompra = 2.02, demais parametros válidos. |
| Saída | Runtime Exception de PrecoVenda |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvF4 | |
| Descrição | Verifica se o preço de venda inserido é maior que preço de compra |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com PrecoVenda = 2.02 e PrecoCompra = 2.03, demais parametros válidos. |
| Saída | Runtime Exception de PrecoVenda |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-A5 | |
| Descrição | Verifica se a unidade inserida é válida. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Unidade = “KG”, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-B5 | |
| Descrição | Verifica se a unidade inserida é válida. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Unidade = “ml”, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvA5 | |
| Descrição | Verifica se a unidade inserida é válida. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Unidade = “m”, demais parametros válidos. |
| Saída | RuntimeException de Unidade |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |
| Caso de Teste 15-InvB5 | |
| Descrição | Verifica se a unidade inserida é válida. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Unidade = “”, demais parametros válidos. |
| Saída | RuntimeException de Unidade |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvC5 | |
| Descrição | Verifica se a unidade inserida é válida. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Unidade = “mlp”, demais parametros válidos. |
| Saída | RuntimeException de Unidade |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-A6 | |
| Descrição | Verifica se o estoque inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Estoque = 4, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-B6 | |
| Descrição | Verifica se o estoque inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Estoque = 0, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvA6 | |
| Descrição | Verifica se o estoque inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Estoque = -1, demais parametros válidos. |
| Saída | RuntimeException de Estoque |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvB6 | |
| Descrição | Verifica se o estoque inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Estoque = -300, demais parametros válidos. |
| Saída | RuntimeException de Estoque |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-A7 | |
| Descrição | Verifica se o Status inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Status = 1, ou seja, ativo, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-B7 | |
| Descrição | Verifica se o Status inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Estoque = 0 e Status = 0 , ou seja, inativo, demais parametros válidos. |
| Saída | Produto criado e inserido na lista de Produtos |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvA7 | |
| Descrição | Verifica se o Status inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Estoque = 1 e Status = 0 , ou seja, inativo, demais parametros válidos. |
| Saída | RuntimeException de Status |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 15-InvB7 | |
| Descrição | Verifica se o Status inserido é válido. |
| Método(s) | public static void insereProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Estoque = 0 e Status = 3, demais parametros válidos. |
| Saída | RuntimeException de Status |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| Codigo | Número inteiro positivo  Não pode ser repetido | - Codigo <=0  - Código já existente em outro Produto |
| Descrição | Alfanumerico, minimo 3 caracteres, máximo 64 | - Tamanho menor que 3  - Tamanho maior que 64  - Não ser alfanumerico |
| Preço de Compra | Número real positivo | - Número negativo  - Zero |
| Preço de Venda | Número real positivo, maior que o Preço de Compra | - Número negativo  - Número maior que o Preço de Compra.  - Zero |
| Unidade | Alfabético, tamanho = 2 | - Sem ser alfabetico  - Sem ter tamanho = 2 |
| Estoque | Número real positivo (incluindo zero) | - Número negativo |
| Status | Ativo, Inativo  Inativo apenas se estoque de produto for zero. | - Qualquer coisa diferente dos dois status pré-definidos.  - Inativar produto com estoque maior que zero. |

### Casos de Teste

Todos os casos de teste do método **insereProduto** (exceto os 15-C1 e 15-InvC1) aplicados em um determinado Produto já inserido. Dependências incluem a função insereProduto. Saída é o Objeto ser alterado com sucesso, ou RuntimeException.

|  |  |
| --- | --- |
| Caso de Teste 16-C1 | |
| Descrição | Verifica se o código não está incluido em outro Produto |
| Método(s) | public void alteraProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Código = 10, demais parametros válidos.  Insere Produto com Código = 11, demais parametros válidos.  Altera o produto de codigo 11, mudando outros valores e nao seu codigo. |
| Saída | Produto 11 alterado com sucesso |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus, insereProduto |

|  |  |
| --- | --- |
| Caso de Teste 16-InvC1 | |
| Descrição | Verifica se o código não está incluido em outro Produto |
| Método(s) | public void alteraProduto(int codigo, String descricao, double precoCompra, double precoVenda, String unidade, double estoque, int status) |
| Entrada | Insere Produto com Código = 10, demais parametros válidos.  Insere Produto com Código = 11, demais parametros válidos.  Altera o produto de codigo 11, mudando seu código para 10. |
| Saída | RuntimeException de Codigo repetido |
| Dependências | setCodigo, setDescricao, setPrecoCompra, setPrecoVenda,setUnidade, setEstoque, setStatus, insereProduto |

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

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 17-A | |
| Descrição | Verifica se o getCodigo retorna o parametro Codigo inserido |
| Método(s) | public String getCodigo() |
| Entrada | Cria objeto Cliente c com Codigo “*1*”.  Execute c.getCodigo |
| Saída | “*1*” (sem aspas). |
| Dependências | setCodigo |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| Codigo | Número inteiro positivo | - Codigo <=0 |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 18-A | |
| Descrição | Verifica se o código inserido é válido. |
| Método(s) | public void setCodigo(int codigo) |
| Entrada | Cria produto  Seta Código = 1. |
| Saída | Codigo do Produto p salvo como o valor fornecido na entrada do metodo setCodigo |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 18-B | |
| Descrição | Verifica se o código inserido é válido. |
| Método(s) | public void setCodigo(int codigo) |
| Entrada | Cria produto  Seta Código = 10. |
| Saída | Codigo do Produto p salvo como o valor fornecido na entrada do metodo setCodigo |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 18-InvA | |
| Descrição | Verifica se o código inserido é válido. |
| Método(s) | public void setCodigo(int codigo) |
| Entrada | Cria produto  Seta Código = 0, demais parametros válidos. |
| Saída | RuntimeException de Código Inválido |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 18-InvB | |
| Descrição | Verifica se o código inserido é válido. |
| Método(s) | public void setCodigo(int codigo) |
| Entrada | Cria produto  Seta Código = -1, demais parametros válidos. |
| Saída | RuntimeException de Código Inválido |
| Dependências |  |

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

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 19-A | |
| Descrição | Verifica se o getDescricao retorna o parametro Descricao inserido |
| Método(s) | public String getDescricao() |
| Entrada | Cria objeto Cliente c com Descrição “*Uva*”.  Execute c. getDescricao |
| Saída | “*Uva*” (sem aspas). |
| Dependências | setDescricao |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| Descrição | String, minimo 3 caracteres, máximo 64 | - Tamanho menor que 3  - Tamanho maior que 64 |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 20-A | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public void setDescricao(String descricao) |
| Entrada | Cria produto  Seta Descricao = “Produto1”. |
| Saída | Descrição do Produto p salvo como o valor fornecido na entrada do metodo setDescricao |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 20-B | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public void setDescricao(String descricao) |
| Entrada | Cria produto  Seta Descricao = “Uva”. |
| Saída | Descrição do Produto p salvo como o valor fornecido na entrada do metodo setDescricao |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 20-C | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public void setDescricao(String descricao) |
| Entrada | Cria produto  Seta Descricao = “AUva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1U64”. |
| Saída | Descrição do Produto p salvo como o valor fornecido na entrada do metodo setDescricao |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 20-D | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public void setDescricao(String descricao) |
| Entrada | Cria produto  Seta Descricao = “AUva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva163”. |
| Saída | Descrição do Produto p salvo como o valor fornecido na entrada do metodo setDescricao |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 20-E | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public void setDescricao(String descricao) |
| Entrada | Cria produto  Seta Descricao = “AUv4”. |
| Saída | Descrição do Produto p salvo como o valor fornecido na entrada do metodo setDescricao |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 20-InvA | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public void setDescricao(String descricao) |
| Entrada | Cria produto  Seta Descricao = “AUva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uva1Uv65”. |
| Saída | RuntimeException de Descrição Inválida |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 20-InvB | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public void setDescricao(String descricao) |
| Entrada | Cria produto  Seta Descricao = “Uv”. |
| Saída | RuntimeException de Descrição Inválida |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 20-InvC | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public void setDescricao(String descricao) |
| Entrada | Cria produto  Seta Descricao = “”. |
| Saída | RuntimeException de Descrição Inválida |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 20-InvD | |
| Descrição | Verifica se a descrição inserida é válida. |
| Método(s) | public void setDescricao(String descricao) |
| Entrada | Cria produto  Seta Descricao = “Uv@ Uva1”. |
| Saída | RuntimeException de Descrição Inválida |
| Dependências |  |

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

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 19-A | |
| Descrição | Verifica se o getPrecoCompra retorna o parametro PrecoCompra inserido |
| Método(s) | public String getPrecoCompra() |
| Entrada | Cria objeto Cliente c com PrecoCompra “*1.30*”.  Execute c. getPrecoCompra |
| Saída | “*1.30*” (sem aspas). |
| Dependências | setPrecoCompra |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| Preço de Compra | Número real positivo | - Número negativo  - Zero |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 22-A | |
| Descrição | Verifica se o preço de compra inserido é válido. |
| Método(s) | public void setPrecoCompra(double precoCompra) |
| Entrada | Cria produto  Seta PrecoCompra = 1.00. |
| Saída | PrecoCompra do Produto p salvo como o valor fornecido na entrada do metodo setPrecoCompra |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 22-B | |
| Descrição | Verifica se o preço de compra inserido é válido. |
| Método(s) | public void setPrecoCompra(double precoCompra) |
| Entrada | Cria produto  Seta PrecoCompra = 4.57. |
| Saída | PrecoCompra do Produto p salvo como o valor fornecido na entrada do metodo setPrecoCompra |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 22-C | |
| Descrição | Verifica se o preço de compra inserido é válido. |
| Método(s) | public void setPrecoCompra(double precoCompra) |
| Entrada | Cria produto  Seta PrecoCompra = 0.01. |
| Saída | PrecoCompra do Produto p salvo como o valor fornecido na entrada do metodo setPrecoCompra |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 22-InvA | |
| Descrição | Verifica se o preço de compra inserido é válido. |
| Método(s) | public void setPrecoCompra(double precoCompra) |
| Entrada | Cria produto  Seta PrecoCompra = 0.00. |
| Saída | RuntimeException de PrecoCompra |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 22-InvB | |
| Descrição | Verifica se o preço de compra inserido é válido. |
| Método(s) | public void setPrecoCompra(double precoCompra) |
| Entrada | Cria produto  Seta PrecoCompra = - 0.01. |
| Saída | RuntimeException de PrecoCompra |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 22-InvC | |
| Descrição | Verifica se o preço de compra inserido é válido. |
| Método(s) | public void setPrecoCompra(double precoCompra) |
| Entrada | Cria produto  Seta PrecoCompra = - 3.00. |
| Saída | RuntimeException de PrecoCompra |
| Dependências |  |

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

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 23-A | |
| Descrição | Verifica se o getPrecoVenda retorna o parametro PrecoVenda inserido |
| Método(s) | public String getPrecoVenda() |
| Entrada | Cria objeto Cliente c com PrecoCompra “*1.30*” e PrecoVenda “1.40”.  Execute c. getPrecoVenda |
| Saída | “*1.40*” (sem aspas). |
| Dependências | setPrecoCompra, setPrecoVenda |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| Preço de Venda | Número real positivo, maior que o Preço de Compra | - Número negativo  - Número maior que o Preço de Compra.  - Zero |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 24-A | |
| Descrição | Verifica se o preço de venda inserido é válido. |
| Método(s) | public void setPrecoVenda(double precoVenda) |
| Entrada | Cria produto  Seta PrecoVenda = 1.00 e PrecoCompra=0.99. |
| Saída | PrecoVenda do Produto p salvo como o valor fornecido na entrada do metodo setPrecoVenda |
| Dependências | setPrecoCompra |

|  |  |
| --- | --- |
| Caso de Teste 24-B | |
| Descrição | Verifica se o preço de venda inserido é válido. |
| Método(s) | public void setPrecoVenda(double precoVenda) |
| Entrada | Cria produto  Seta PrecoVenda = 3.12 e PrecoCompra = 2.61. |
| Saída | PrecoVenda do Produto p salvo como o valor fornecido na entrada do metodo setPrecoVenda |
| Dependências | setPrecoCompra |

|  |  |
| --- | --- |
| Caso de Teste 24-C | |
| Descrição | Verifica se o preço de venda inserido é válido. |
| Método(s) | public void setPrecoVenda(double precoVenda) |
| Entrada | Cria produto  Seta PrecoVenda = 0.02 e PrecoCompra=0.01. |
| Saída | PrecoVenda do Produto p salvo como o valor fornecido na entrada do metodo setPrecoVenda |
| Dependências | setPrecoCompra |

|  |  |
| --- | --- |
| Caso de Teste 24-InvA | |
| Descrição | Verifica se o preço de venda inserido é válido. |
| Método(s) | public void setPrecoVenda(double precoVenda) |
| Entrada | Cria produto  Seta PrecoVenda = 0.00. |
| Saída | Runtime Exception de PrecoVenda |
| Dependências | setPrecoCompra |

|  |  |
| --- | --- |
| Caso de Teste 24-InvB | |
| Descrição | Verifica se o preço de venda inserido é válido. |
| Método(s) | public void setPrecoVenda(double precoVenda) |
| Entrada | Cria produto  Seta PrecoVenda = - 0.01. |
| Saída | Runtime Exception de PrecoVenda |
| Dependências | setPrecoCompra |

|  |  |
| --- | --- |
| Caso de Teste 24-InvC | |
| Descrição | Verifica se o preço de venda inserido é válido. |
| Método(s) | public void setPrecoVenda(double precoVenda) |
| Entrada | Cria produto  Seta PrecoVenda = - 3.00. |
| Saída | Runtime Exception de PrecoVenda |
| Dependências | setPrecoCompra |

|  |  |
| --- | --- |
| Caso de Teste 24-InvD | |
| Descrição | Verifica se o preço de venda inserido é maior que preço de compra |
| Método(s) | public void setPrecoVenda(double precoVenda) |
| Entrada | Cria produto  Seta PrecoVenda = 0.02 e PrecoCompra = 0.02. |
| Saída | Runtime Exception de PrecoVenda |
| Dependências | setPrecoCompra |

|  |  |
| --- | --- |
| Caso de Teste 24-InvE | |
| Descrição | Verifica se o preço de venda inserido é maior que preço de compra |
| Método(s) | public void setPrecoVenda(double precoVenda) |
| Entrada | Cria produto  Seta PrecoVenda = 1.02 e PrecoCompra = 2.02. |
| Saída | Runtime Exception de PrecoVenda |
| Dependências | setPrecoCompra |

|  |  |
| --- | --- |
| Caso de Teste 24-InvF | |
| Descrição | Verifica se o preço de venda inserido é maior que preço de compra |
| Método(s) | public void setPrecoVenda(double precoVenda) |
| Entrada | Cria produto  Seta PrecoVenda = 2.02 e PrecoCompra = 2.03. |
| Saída | Runtime Exception de PrecoVenda |
| Dependências | setPrecoCompra |

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

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 25-A | |
| Descrição | Verifica se o getUnidade retorna o parametro Unidade inserido |
| Método(s) | public String getUnidade() |
| Entrada | Cria objeto Cliente c com Unidade “*KG*”.  Execute c. getUnidade |
| Saída | “*KG*” (sem aspas). |
| Dependências | setUnidade |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| Unidade | Alfabético, tamanho = 2 | - Sem ser alfabetico  - Sem ter tamanho = 2 |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 26-A | |
| Descrição | Verifica se a unidade inserida é válida. |
| Método(s) | public void setUnidade(String unidade) |
| Entrada | Cria produto  Seta Unidade = “KG” |
| Saída | Unidade do Produto p salvo como o valor fornecido na entrada do metodo setUnidade |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 26-B | |
| Descrição | Verifica se a unidade inserida é válida. |
| Método(s) | public void setUnidade(String unidade) |
| Entrada | Cria produto  Seta Unidade = “ml” |
| Saída | Unidade do Produto p salvo como o valor fornecido na entrada do metodo setUnidade |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 26-InvA | |
| Descrição | Verifica se a unidade inserida é válida. |
| Método(s) | public void setUnidade(String unidade) |
| Entrada | Cria produto  Seta Unidade = “m” |
| Saída | RuntimeException de Unidade |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 26-InvB | |
| Descrição | Verifica se a unidade inserida é válida. |
| Método(s) | public void setUnidade(String unidade) |
| Entrada | Cria produto  Seta Unidade = “” |
| Saída | RuntimeException de Unidade |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 26-InvC | |
| Descrição | Verifica se a unidade inserida é válida. |
| Método(s) | public void setUnidade(String unidade) |
| Entrada | Cria produto  Seta Unidade = “mlp” |
| Saída | RuntimeException de Unidade |
| Dependências |  |

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

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 27-A | |
| Descrição | Verifica se o getEstoque retorna o parametro Estoque inserido |
| Método(s) | public String getUnidade() |
| Entrada | Cria objeto Cliente c com Estoque “*22*”.  Execute c. getEstoque |
| Saída | “*22*” (sem aspas). |
| Dependências | setEstoque |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| Estoque | Número real positivo (incluindo zero) | - Número negativo |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 28-A | |
| Descrição | Verifica se o estoque inserido é válido. |
| Método(s) | public void setEstoque(double estoque) |
| Entrada | Cria produto  Seta Estoque = 4. |
| Saída | Estoque do Produto p salvo como o valor fornecido na entrada do metodo setEstoque |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 28-B | |
| Descrição | Verifica se o estoque inserido é válido. |
| Método(s) | public void setEstoque(double estoque) |
| Entrada | Cria produto  Seta Estoque = 0. |
| Saída | Estoque do Produto p salvo como o valor fornecido na entrada do metodo setEstoque |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 28-InvA | |
| Descrição | Verifica se o estoque inserido é válido. |
| Método(s) | public void setEstoque(double estoque) |
| Entrada | Cria produto  Seta Estoque = -1. |
| Saída | RuntimeException de Estoque |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 28-InvB | |
| Descrição | Verifica se o estoque inserido é válido. |
| Método(s) | public void setEstoque(double estoque) |
| Entrada | Cria produto  Seta Estoque = -300. |
| Saída | RuntimeException de Estoque |
| Dependências |  |

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

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 27-A | |
| Descrição | Verifica se o getStatus retorna o parametro Status inserido |
| Método(s) | public String geStatus() |
| Entrada | Cria objeto Cliente c com Status “*1*”.  Execute c. getStatus |
| Saída | “*1*” (sem aspas). |
| Dependências | setStatus |

|  |  |
| --- | --- |
| Caso de Teste 27-B | |
| Descrição | Verifica se o getStatus retorna o parametro Status inserido |
| Método(s) | public String geStatus () |
| Entrada | Cria objeto Cliente c com Status “0”.  Execute c. getStatus |
| Saída | “*0*” (sem aspas). |
| Dependências | setStatus |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| Status | Ativo, Inativo  Inativo apenas se estoque de produto for zero. | - Qualquer coisa diferente dos dois status pré-definidos.  - Inativar produto com estoque maior que zero. |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 30-A | |
| Descrição | Verifica se o Status inserido é válido. |
| Método(s) | public void setStatus(int status) |
| Entrada | Cria produto  Seta Status = 1, ou seja, ativo, demais parametros válidos. |
| Saída | Status do Produto p salvo como o valor fornecido na entrada do metodo setStatus |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 30-B | |
| Descrição | Verifica se o Status inserido é válido. |
| Método(s) | public void setStatus(int status) |
| Entrada | Cria produto  Seta Estoque = 0 e Status = 0 , ou seja, inativo, demais parametros válidos. |
| Saída | Status do Produto p salvo como o valor fornecido na entrada do metodo setStatus |
| Dependências | setEstoque |

|  |  |
| --- | --- |
| Caso de Teste 30-InvA | |
| Descrição | Verifica se o Status inserido é válido. |
| Método(s) | public void setStatus(int status) |
| Entrada | Cria produto  Seta Estoque = 1 e Status = 0 , ou seja, inativo, demais parametros válidos. |
| Saída | RuntimeException de Status |
| Dependências | setEstoque |

|  |  |
| --- | --- |
| Caso de Teste 30-InvB | |
| Descrição | Verifica se o Status inserido é válido. |
| Método(s) | public void setStatus(int status) |
| Entrada | Cria produto  Seta Estoque = 0 e Status = 3, demais parametros válidos. |
| Saída | RuntimeException de Status |
| Dependências | setEstoque |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| Cpf | CPF que nao existe na lista de Clientes | - CPF que existe na lista de Clientes |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 31-A | |
| Descrição | Verifica se String de Cpf é valida |
| Método(s) | private static boolean validaCpfUnico(String cpf) |
| Entrada | Manda o cpf “40114627487” para a funcao. Ele existe na lista de Clientes |
| Saída | false |
| Dependências | getCpf |

|  |  |
| --- | --- |
| Caso de Teste 31-B | |
| Descrição | Verifica se String de Cpf é valida |
| Método(s) | private static boolean validaCpfUnico(String cpf) |
| Entrada | Manda o cpf “55555555555” para a funcao. |
| Saída | true |
| Dependências | getCpf |

|  |  |
| --- | --- |
| Caso de Teste 31-B | |
| Descrição | Verifica se String de Cpf é valida |
| Método(s) | private static boolean validaCpfUnico(String cpf) |
| Entrada | Manda o cpf “40114627487” para a funcao. Ele nao existe na lista de Clientes |
| Saída | true |
| Dependências | getCpf |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| Cpf | CPF que nao existe na lista de Clientes | - CPF que existe na lista de Clientes |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 32-A | |
| Descrição | Verifica se String de Cpf é valida |
| Método(s) | private static boolean validaCpfUnico(String cpf) |
| Entrada | Manda o cpf “40114627487” para a funcao. Ele existe na lista de Clientes |
| Saída | Cliente com respectivo Cpf |
| Dependências | getCpf |

|  |  |
| --- | --- |
| Caso de Teste 32-B | |
| Descrição | Verifica se String de Cpf é valida |
| Método(s) | private static boolean validaCpfUnico(String cpf) |
| Entrada | Manda o cpf “55555555555” para a funcao. |
| Saída | null |
| Dependências | getCpf |

|  |  |
| --- | --- |
| Caso de Teste 32-B | |
| Descrição | Verifica se String de Cpf é valida |
| Método(s) | private static boolean validaCpfUnico(String cpf) |
| Entrada | Manda o cpf “40114627487” para a funcao. Ele nao existe na lista de Clientes |
| Saída | null |
| Dependências | getCpf |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| Mês (int 01 até 12) | Meses anteriores ao atual | - Meses posteriores ao atual  - Numeros que nao representam meses |

### 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 |
| Descrição | Qualquer string |  |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 34-A | |
| Descrição | Verifica saidas |
| Método(s) | public ArrayList<Integer> **consultaProdutos** (String descricao) |
| Entrada | Manda a descricao “Uva” para a funcao. Existem produtos que começam por essa descricao |
| Saída | ArrayLIst<int> p.Codigo dos produtos que tenham descrição começada por “Uva” |
| Dependências | getDescricao |

|  |  |
| --- | --- |
| Caso de Teste 34-B | |
| Descrição | Verifica saidas |
| Método(s) | public ArrayList<Integer> **consultaProdutos** (String descricao) |
| Entrada | Manda a descricao “Uva” para a funcao. Não produtos que começam por essa descricao |
| Saída | -1 |
| Dependências | getDescricao |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| Codigo | Qualquer numero inteiro |  |
| Estoque | Qualquer numero inteiro positivo (incluindo zero) | Numeros negativos |

### 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 |
| Código | Valor = 1 | Valor = 2 |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 34-A | |
| Descrição | Verifica saidas |
| Método(s) | private static boolean consultaProdutoPorcodigo(int Codigo) |
| Entrada | Manda o Codigo “1” para a funcao. Existe produto que tenham esse codigo |
| Saída | Produto de codigo “1” |
| Dependências | getCodigo |

|  |  |
| --- | --- |
| Caso de Teste 34-B | |
| Descrição | Verifica saidas |
| Método(s) | private static boolean consultaProdutoPorcodigo(int Codigo) |
| Entrada | Manda o Codigo “2” para a funcao. Nao existe produto que tenham esse codigo |
| Saída | null |
| Dependências | getCodigo |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| Descrição | Qualquer string |  |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 37-A | |
| Descrição | Verifica saidas |
| Método(s) | private static boolean validaCpfUnico(String cpf) |
| Entrada | Manda a descricao “Uva” para a funcao. Existem produtos que começam por essa descricao |
| Saída | ArrayLIst<int> p.Codigo dos produtos que tenham descrição começada por “Uva” |
| Dependências | getDescricao |

|  |  |
| --- | --- |
| Caso de Teste 37-B | |
| Descrição | Verifica saidas |
| Método(s) | private static boolean validaCpfUnico(String cpf) |
| Entrada | Manda a descricao “Uva” para a funcao. Não produtos que começam por essa descricao |
| Saída | -1 |
| Dependências | getDescricao |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| Código | Qualquer numero inteiro |  |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 38-A | |
| Descrição | Verifica saidas |
| Método(s) | private static boolean validaCodigo(int Codigo) |
| Entrada | Manda o Codigo “1” para a funcao. Existe produto que tenham esse codigo |
| Saída | false |
| Dependências | getCodigo |

|  |  |
| --- | --- |
| Caso de Teste 38-B | |
| Descrição | Verifica saidas |
| Método(s) | private static boolean validaCodigo(int Codigo) |
| Entrada | Manda o Codigo “2” para a funcao. Nao existe produto que tenham esse codigo |
| Saída | true |
| Dependências | getCodigo |

* 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 aceito |
| 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 **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 53-A | |
| Descrição | Verificar o funcionamento do método buscaPorNumero |
| Método(s) | buscaPorNumero(int) |
| Entrada | Registrar um RegistroVenda, buscar seu número por buscaPorNumero(numero) |
| Saída | Retorna o objeto RegistroVenda |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 53-B | |
| Descrição | Verificar o funcionamento do método buscaPorNumero |
| Método(s) | buscaPorNumero(int) |
| Entrada | Registrar um RegistroVenda, buscar seu número por buscaPorNumero(numero) |
| Saída | Retorna null |
| Dependências |  |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| - | Retorna array de registros | - |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 54-A | |
| Descrição | Verifica o método getRegistros() |
| Método(s) | getRegistros() |
| Entrada | Cadastrar alguns registros e usar getRegistros() |
| Saída | Array de RegistroVenda |
| Dependências |  |

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

### Classes de Equivalência

|  |  |  |
| --- | --- | --- |
| Condição de Entrada | Classes Válidas | Classes Inválidas |
| Número do mês | Mês inteiro e entre 0 e 12 | <0 |
| 0 |
| >12 |

### Casos de Teste

|  |  |
| --- | --- |
| Caso de Teste 55-A | |
| Descrição | Verificar se o funcionamento do método getRegistroDoMes |
| Método(s) | getRegistroDoMes(int) |
| Entrada | 1 |
| Saída | Retorna registros do mês de janeiro do ano atual |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 55-InvA | |
| Descrição | Verificar se o funcionamento do método getRegistroDoMes |
| Método(s) | getRegistroDoMes(int) |
| Entrada | -1 |
| Saída | Exceção |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 55-InvB | |
| Descrição | Verificar se o funcionamento do método getRegistroDoMes |
| Método(s) | getRegistroDoMes(int) |
| Entrada | 0 |
| Saída | Exceção |
| Dependências |  |

|  |  |
| --- | --- |
| Caso de Teste 55-InvC | |
| Descrição | Verificar se o funcionamento do método getRegistroDoMes |
| Método(s) | getRegistroDoMes(int) |
| Entrada | 13 |
| Saída | Exceção |
| Dependências |  |