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 void **printRegistro**(int codigo) | Imprime informações de um registro venda, com base em seu código |
| 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 **getNome** – Classe **Ciente**

### Classes de Equivalência

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

### Casos de Teste

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

* 1. Metodo **setNome** – Classe **Ciente**

### Classes de Equivalência

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

### Casos de Teste

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

* 1. Metodo **getCpf** – Classe **Ciente**

### Classes de Equivalência

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

### Casos de Teste

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

* 1. Metodo **setCpf** – Classe **Ciente**

### Classes de Equivalência

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

### Casos de Teste

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

* 1. Metodo **getEndereco** – Classe **Ciente**

### Classes de Equivalência

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

### Casos de Teste

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

* 1. Metodo **setEndereco** – Classe **Ciente**

### Classes de Equivalência

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

### Casos de Teste

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

* 1. Metodo **getTelefone** – Classe **Ciente**

### Classes de Equivalência

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

### Casos de Teste

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

* 1. Metodo **setTelefone** – Classe **Ciente**

### Classes de Equivalência

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

### Casos de Teste

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

* 1. Metodo **getEmail** – Classe **Ciente**

### Classes de Equivalência

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

### Casos de Teste

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

* 1. Metodo **setEmail** – Classe **Ciente**

### Classes de Equivalência

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

### Casos de Teste

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

* 1. Metodo **getStatus** – Classe **Ciente**

### Classes de Equivalência

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

### Casos de Teste

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

* 1. Metodo **setStatus** – Classe **Ciente**

### Classes de Equivalência

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

### Casos de Teste

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

* 1. Metodo **alteraCliente** – Classe **Ciente**

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

* 1. Metodo **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 String.  - Não ser alfabetica  - Ter mais de 128 caracteres.  - Menos de duas palavras.  - Palavras com tamanho menor que 2. |
| CPF | 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. |
| Endereço | String alfanumérica de no máximo 256 caracteres | - Não ser String.  - Não ser alfanumérico.  - Ter mais de 256 caracteres. |
| Telefone | (XX)XXXXXXXXX, com X sendo números | - Não ser String.  - Não conter apenas números e parentheses.  - 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 ser String.  - 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 coisas 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 Cliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Cria um objeto Cliente com Nome *“Robert Richards”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

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

|  |  |
| --- | --- |
| Caso de Teste 01-C1 | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public Cliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Cria um objeto 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 |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

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

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

|  |  |
| --- | --- |
| Caso de Teste 01-F1 | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public Cliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Cria um objeto Cliente com Nome *“Ro Rim”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |
| Caso de Teste 01-G1 | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public Cliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Cria um objeto Cliente com Nome *“Ro Ri Do”* (sem aspas). Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 01-H1 | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public Cliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Cria um objeto 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 |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

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

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

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

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

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

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

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

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

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

|  |  |
| --- | --- |
| Caso de Teste 01-InvJ1 | |
| Descrição | Verifica se o nome inserido é válido. |
| Método(s) | public Cliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Cria um objeto 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 Cliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Cria um objeto Cliente com Cpf “40114627487”. Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 01-InvA2 | |
| Descrição | Verifica se o Cpf inserido é válido. |
| Método(s) | public Cliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Cria um objeto Cliente com Cpf “5555555555”. 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 Cliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Cria um objeto Cliente com Cpf “555555555555”. 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 Cliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Cria um objeto Cliente com Cpf “5555555555555”. 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 Cliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Cria um objeto Cliente com Cpf “11111111111”. 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 Cliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Cria um objeto Cliente com Cpf “48391291828”. Demais parâmetros com suas entradas válidas. |
| Saída | RuntimeException de CPF inválido |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

|  |  |
| --- | --- |
| Caso de Teste 01-A3 | |
| Descrição | Verifica se o Endereço inserido é válido. |
| Método(s) | public Cliente(String nome, String cpf, String endereco, String telefone, String email, int status) |
| Entrada | Cria um objeto Cliente com Endereço “. Demais parâmetros também com suas entradas válidas. |
| Saída | Objeto Cliente criado |
| Dependências | setNome, setCpf, setEndereco, setTelefone, setEmail, setStatus |

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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

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

### Classes de Equivalência

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

### Casos de Teste

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