

# Documentação

## Classe Interface

A classe `Interface` é responsável por implementar a interface gráfica do usuário (GUI) da aplicação de cadastro de usuários para uma distribuidora. Utilizando a biblioteca Java Swing, a classe oferece funcionalidades para a entrada de dados do usuário e a exibição de formulários de cadastro.

### Principais Funcionalidades:

- **Tela Inicial:** Exibe um painel de login onde o usuário pode inserir suas credenciais (usuário e senha) e escolher entre CPF e CNPJ.
- **Cadastro de Usuário:** Permite ao usuário acessar um formulário de cadastro onde são solicitados dados como nome, telefone, e-mail e senha.
- **Persistência de Dados:** Integra-se com a classe `UsuarioPersistencia` para salvar os dados do usuário no sistema.
- **Diálogos:** Utiliza `JDialog` para exibir o formulário de cadastro e mensagens de confirmação.

## Classe InterfaceEstoque

A classe `InterfaceEstoque` implementa a interface gráfica do usuário (GUI) para a gestão do estoque de produtos em uma distribuidora. Utilizando a biblioteca Java Swing, a classe permite visualizar produtos e seus respectivos lotes, oferecendo funcionalidades para interação com os dados do estoque.

### Principais Funcionalidades:

- **Visualização de Produtos:** Exibe uma lista de produtos disponíveis no estoque.
- **Visualização de Lotes:** Permite ao usuário selecionar um produto e visualizar os lotes associados a ele.
- **Detalhes do Lote:** Mostra informações detalhadas sobre o lote selecionado, incluindo código, quantidade e validade.

## Classe Arquivo

A classe `Arquivo` fornece métodos utilitários para a leitura e gravação de dados em arquivos. Esta classe simplifica a manipulação de arquivos de texto, permitindo que o usuário leia o conteúdo de um arquivo ou salve novas informações em um arquivo existente.

### Principais Funcionalidades:

- **Leitura de Arquivos:** Permite ler o conteúdo de um arquivo de texto e retornar o conteúdo como uma única string.
- **Gravação em Arquivos:** Permite salvar uma string em um arquivo de texto, sobrescrevendo o conteúdo existente.

## Interface Persistencia

A interface `Persistencia` define um contrato genérico para operações de persistência de dados em uma aplicação. É projetada para ser implementada por classes que desejam fornecer funcionalidades de armazenamento e recuperação de objetos de um tipo específico.

### Principais Funcionalidades:

- **Salvar Itens:** Permite a gravação de uma lista de itens de um tipo genérico.
- **Recuperar Itens:** Permite a recuperação de todos os itens armazenados do tipo especificado.

## Classe PessoaPersistencia

A classe `PessoaPersistencia` é responsável pela persistência de objetos do tipo `Pessoa` em um arquivo JSON. Implementando a interface `Persistencia`, esta classe permite gerenciar a leitura e gravação de uma lista de pessoas, facilitando a manipulação e armazenamento dos dados de forma estruturada.

### Principais Funcionalidades:

- **Salvar Pessoa:** Adiciona uma nova pessoa à lista existente e a grava no arquivo "pessoas.json".
- **Carregar Pessoas:** Lê o arquivo JSON e retorna a lista de pessoas armazenadas, retornando uma lista vazia caso o arquivo não exista.
- **Salvar Lista de Pessoas:** Sobrescreve o conteúdo do arquivo JSON com uma nova lista de pessoas.
- **Encontrar Todas as Pessoas:** Retorna a lista completa de pessoas armazenadas no arquivo JSON.

A classe utiliza a biblioteca `Gson` para converter objetos Java para o formato JSON, garantindo uma manipulação eficiente dos dados.

## Classe Pessoa

A classe `Pessoa` representa um usuário na aplicação, encapsulando informações pessoais como nome, telefone, email, login e endereço. Essa classe é fundamental para gerenciar os dados dos usuários, assegurando a validação adequada de informações sensíveis, como telefone e email, através de exceções personalizadas.

### Principais Funcionalidades:

- **Criação de Pessoa:** Permite instanciar um objeto `Pessoa` com atributos como nome, telefone, login, endereço e email, garantindo que os dados sejam validados durante a criação.
- **Validação de Telefone e Email:** Utiliza classes auxiliares (`Telefone` e `Email`) para garantir que os formatos de telefone e email sejam corretos, lançando exceções apropriadas em caso de erro.
- **Getters e Setters:** Fornece métodos para acessar e modificar os atributos da classe, permitindo a manipulação dos dados do usuário de forma controlada.

Essa estrutura permite que a classe `Pessoa` seja utilizada em diversas partes da aplicação, mantendo a integridade e a validade das informações do usuário.

## Classe Cliente

A classe Cliente estende a classe Pessoa e representa um usuário específico que pode realizar transações em uma aplicação de gestão de distribuidora. Além dos atributos básicos herdados, como nome, telefone, login, endereço e email, a classe Cliente inclui funcionalidades voltadas para a interação com produtos e processos de compra.

### Principais Funcionalidades:

- **Identificação do Cliente:** Atribui um ID único a cada cliente, incrementando automaticamente a cada nova instância criada.
- **Manipulação de Produtos:** Métodos para inserir, retirar e consultar produtos, permitindo que o cliente interaja com o estoque.
- **Finalização de Pedidos:** Método para concluir um pedido, integrando a lógica de compra na aplicação.
- **Pagamento:** Método para processar pagamentos, facilitando a finalização das transações.

Além disso, a classe inclui métodos para gerenciar a senha do cliente e formatar o CPF, garantindo que as informações sejam tratadas de forma consistente e segura. Essa estrutura permite uma gestão eficiente dos clientes dentro do sistema, facilitando a implementação de funcionalidades comerciais.

## Classe Fornecedor

A classe Fornecedor representa uma entidade que fornece produtos ou serviços em um sistema de gestão de distribuidora. Esta classe encapsula informações essenciais sobre o fornecedor, incluindo seu nome social, telefone, email, endereço e CNPJ, garantindo a validação e o armazenamento adequado desses dados.

### Principais Funcionalidades:

- **Informações do Fornecedor:** Armazena dados como nome social, telefone, email e endereço, permitindo fácil acesso a essas informações.
- **Validação do CNPJ:** A classe incorpora a lógica de validação do CNPJ, garantindo que o número fornecido esteja correto antes de ser armazenado.
- **Encapsulamento de Dados:** Utiliza atributos privados para proteger os dados sensíveis, promovendo boas práticas de programação orientada a objetos.

Essa estrutura proporciona uma base sólida para a gestão de fornecedores, facilitando a integração com outras funcionalidades do sistema e assegurando a integridade dos dados coletados.

## Classe Gerente

A classe Gerente representa um usuário do sistema com responsabilidades administrativas em uma distribuidora. Esta classe estende a classe Pessoa, incorporando atributos e métodos que permitem a gestão de informações específicas do gerente, como senha e detalhes de contato.

### Principais Funcionalidades:

- **Gerenciamento de Dados Pessoais:** Armazena informações como nome, telefone, email e endereço, permitindo fácil acesso e modificação.

- **Autenticação Segura:** Inclui um atributo para a senha, proporcionando a funcionalidade de autenticação do gerente no sistema.
- **Controle de Identificação:** Mantém um contador estático para o ID do gerente, garantindo que cada instância do gerente seja única.
- **Métodos de Acesso e Modificação:** Fornece métodos getter e setter para manipular informações de telefone, email e login, promovendo encapsulamento e segurança dos dados.

Essa estrutura permite que o gerente execute funções administrativas e de supervisão, facilitando a gestão eficiente da distribuidora. A classe também está projetada para ser facilmente extensível, permitindo a adição de novos métodos para funcionalidades específicas, como gestão de estoque e geração de relatórios.

## Classe Vendedor

A classe Vendedor representa um usuário do sistema que atua na função de vendas dentro de uma distribuidora. Esta classe estende a classe Pessoa e inclui atributos e métodos específicos para gerenciar as informações e operações de um vendedor.

### Principais Funcionalidades:

- **Informações Pessoais:** Armazena dados como nome, telefone, email e endereço, garantindo que todas as informações do vendedor estejam acessíveis e atualizáveis.
- **Identificação Única:** Possui um contador estático para o ID do vendedor, assegurando que cada instância criada tenha um identificador exclusivo.
- **Autenticação:** Inclui um atributo para a senha, permitindo o gerenciamento seguro do acesso ao sistema.
- **Integração com CPF:** Permite a associação de um CPF ao vendedor, contribuindo para a identificação e a conformidade legal.

A classe Vendedor está estruturada para ser facilmente ampliável, possibilitando a adição de métodos relacionados a operações de venda no futuro, como a realização de vendas e o gerenciamento de pedidos, fortalecendo assim a funcionalidade da aplicação.

## Classe Estoque

A classe Estoque gerencia a coleção de produtos disponíveis em uma distribuidora, permitindo operações de adição, remoção e consulta de itens, bem como a aplicação de cupons e promoções.

### Principais Funcionalidades:

- **Gerenciamento de Produtos:** Mantém uma lista de produtos, permitindo adicionar novos itens e consultar a quantidade disponível de cada um.
- **Cupons e Promoções:** Suporta a criação e verificação de cupons e promoções, facilitando a aplicação de descontos em produtos específicos.
- **Validade de Produtos:** Possui métodos para verificar quais produtos estão próximos da data de validade, ajudando na gestão de estoque e minimização de perdas.
- **Baixa de Estoque:** Permite registrar a retirada de produtos do estoque, tanto por quantidade quanto utilizando cupons, garantindo que o inventário esteja sempre atualizado.

- **Tratamento de Exceções:** Implementa uma série de verificações que garantem que operações inválidas sejam tratadas adequadamente, como busca de produtos e lotes.

A classe Estoque é uma parte fundamental do sistema, proporcionando uma interface robusta para a gestão de inventário e garantindo que todas as operações relacionadas a produtos sejam realizadas de maneira eficiente e segura.

## Classe Produto

A classe Produto representa um item no estoque, gerenciando informações como preço, nome e lotes associados. É responsável por garantir a integridade dos dados do produto e por operações relacionadas ao controle de estoque.

### Principais Funcionalidades:

- **Gerenciamento de Lotes:** Permite cadastrar e acessar lotes associados ao produto, garantindo que as operações sobre os lotes sejam realizadas de forma segura.
- **Quantidade Total:** Fornece métodos para calcular a quantidade total disponível do produto, considerando todos os lotes.
- **Validação de Dados:** Implementa verificações para garantir que o preço e impostos do produto estejam dentro de limites aceitáveis.
- **Manipulação de Preços:** Possui métodos para alterar o preço do produto, assegurando que não seja negativo.
- **Tratamento de Exceções:** Integra um sistema robusto de tratamento de exceções para lidar com situações inválidas, como tentativas de acesso a lotes inexistentes.

Essa classe é fundamental para a gestão de produtos dentro da distribuidora, oferecendo uma interface clara e segura para manipulação de dados relacionados a itens em estoque.

## Classe Lote

A classe Lote representa um agrupamento de itens em estoque, permitindo gerenciar informações como data de aquisição, validade e quantidade disponível. É uma parte essencial da gestão de produtos, garantindo que as operações de controle de estoque sejam realizadas de maneira eficaz.

### Principais Funcionalidades:

- **Validação de Dados:** Implementa verificações rigorosas para assegurar que a validade do lote e a quantidade de itens estejam dentro de limites válidos.
- **Cálculo de Vencimento:** Oferece métodos para calcular quantos dias faltam até o vencimento do lote, utilizando uma data de referência.
- **Remoção de Itens:** Permite a remoção de itens do lote, com validações para evitar remoções negativas ou superiores à quantidade disponível.
- **Armazenamento Estrutural:** Cada lote é identificado por um código único, facilitando a gestão e o rastreamento.

Essa classe é fundamental para garantir a integridade e a rastreabilidade dos produtos em estoque, contribuindo para uma gestão eficiente e precisa.

## Classe Promocao

A classe Promocao representa uma oferta temporária aplicada a um produto específico, permitindo ajustar seu preço para incentivar vendas. É uma ferramenta essencial na gestão de marketing e vendas de produtos em estoque.

### Principais Funcionalidades:

- **Validação de Dados:** Garante que o código do produto e o valor da promoção sejam válidos, evitando códigos nulos ou valores negativos.
- **Armazenamento de Informações:** Mantém o código do produto e o novo valor promocional, facilitando a aplicação da promoção durante as transações de venda.
- **Acesso aos Dados:** Fornece métodos para recuperar o código do produto e o novo valor, permitindo que outros componentes do sistema utilizem essas informações conforme necessário.

Essa classe contribui para a flexibilidade da gestão de preços e promoções, ajudando a aumentar a competitividade e a atratividade dos produtos no mercado.

## Classe Cupom

A classe Cupom é uma extensão da classe Promocao, representando um desconto específico que pode ser aplicado a produtos durante as transações de venda. Esta classe é fundamental para gerenciar ofertas promocionais que envolvem códigos de cupom.

### Principais Funcionalidades:

- **Validação de Dados:** Assegura que o código do cupom e o valor da promoção sejam válidos, prevenindo erros durante o uso de cupons.
- **Herança de Promocao:** Herda características da classe Promocao, mantendo a estrutura de códigos de produtos e novos valores, além de permitir a reutilização de lógica de validação.
- **Acesso ao Código do Cupom:** Oferece um método para acessar o código do cupom, permitindo que sistemas de vendas utilizem essa informação para aplicar descontos durante o checkout.

A classe Cupom é essencial para a implementação de estratégias de marketing e fidelização de clientes, promovendo vendas através de incentivos diretos.

## Classe Data

A classe Data representa uma data específica, armazenando informações sobre o dia, mês e ano. É utilizada em diversas partes do sistema, especialmente para gerenciar prazos de validade e datas de aquisição de produtos.

### Principais Funcionalidades:

- **Validação de Datas:** Garante que os valores de dia, mês e ano sejam válidos, evitando erros de entrada e assegurando a integridade dos dados.
- **Conversão para Dias:** Possui um método `toDays()` que converte a data em um valor numérico, facilitando comparações e operações aritméticas.

- **Comparação de Datas:** Permite a comparação entre duas instâncias de Data, verificando se são iguais e lançando exceções em caso de entradas nulas.

A classe Data é fundamental para o gerenciamento eficiente de informações temporais, garantindo que o sistema funcione corretamente em relação a prazos e datas críticas.

## Classe CPF

A classe CPF representa um Cadastro de Pessoa Física, que é um identificador fiscal utilizado no Brasil. Esta classe é responsável por gerenciar e validar números de CPF, assegurando que apenas dados válidos sejam armazenados no sistema.

### Principais Funcionalidades:

- **Validação do CPF:** Implementa um método que verifica se o CPF informado é válido, considerando sequências repetidas e o cálculo dos dígitos verificadores, conforme as regras oficiais.
- **Formatação do CPF:** Possui um método que formata o CPF para um padrão legível, inserindo pontos e traços, facilitando a visualização.
- **Armazenamento Seguro:** O CPF é armazenado de forma que garante sua integridade, assegurando que sempre esteja no formato correto ao ser acessado.
- **Exceções:** Lança uma exceção específica (`CPFException`) em casos de CPF inválido, proporcionando feedback claro ao usuário.

A classe CPF é essencial para garantir a correta identificação de pessoas no sistema, prevenindo erros de entrada e assegurando que apenas dados válidos sejam processados.

## Classe CNPJ

A classe CNPJ representa o Cadastro Nacional da Pessoa Jurídica, que é um identificador utilizado para empresas no Brasil. Esta classe é responsável por gerenciar e validar números de CNPJ, garantindo que apenas dados válidos sejam armazenados no sistema.

### Principais Funcionalidades:

- **Validação do CNPJ:** Implementa um método que verifica se o CNPJ informado é válido, considerando sequências repetidas e o cálculo dos dígitos verificadores, conforme as regras oficiais.
- **Formatação do CNPJ:** Possui um método que formata o CNPJ em um padrão legível, inserindo pontos e traços, facilitando a visualização.
- **Armazenamento Seguro:** O CNPJ é armazenado de forma a garantir sua integridade, assegurando que sempre esteja no formato correto ao ser acessado.
- **Exceções:** Lança uma exceção específica (`CNPJException`) em casos de CNPJ inválido, proporcionando feedback claro ao usuário.

A classe CNPJ é essencial para garantir a correta identificação de empresas no sistema, prevenindo erros de entrada e assegurando que apenas dados válidos sejam processados.

## Classe Telefone

A classe Telefone é responsável por representar e validar números de telefone, incluindo a verificação do código de área (DDD) e o formato adequado. Essa classe assegura que os dados de contato sejam armazenados corretamente no sistema, evitando entradas inválidas.

### Principais Funcionalidades:

- **Validação do Telefone:** Implementa um método que verifica se o número de telefone está no formato correto, garantindo que ele contenha os caracteres necessários, como parênteses e hífen.
- **Extração de DDD:** A classe é capaz de extrair o DDD do número de telefone, permitindo um tratamento separado para o código de área.
- **Formatação do Telefone:** Armazena o número de telefone de forma formatada, facilitando a apresentação ao usuário e a legibilidade.
- **Exceções:** Lança uma exceção específica (`TelefoneException`) quando um número de telefone inválido é inserido, garantindo que o sistema mantenha dados consistentes.

A classe Telefone é fundamental para garantir que as informações de contato sejam precisas e acessíveis, contribuindo para uma melhor comunicação e interação com os usuários.

## Classe Email

A classe Email é responsável por representar e validar endereços de e-mail, armazenando informações sobre o login e o domínio. Essa classe é crucial para garantir a integridade e a validade dos dados de contato eletrônico no sistema.

### Principais Funcionalidades:

- **Validação de E-mail:** Implementa um método que verifica se o e-mail está no formato correto usando expressões regulares, garantindo que ele siga as convenções padrão.
- **Extração de Login e Domínio:** A classe divide o e-mail em duas partes: o login (parte antes do símbolo "@") e o domínio (parte após o "@"), permitindo acesso fácil a essas informações.
- **Exceções:** Lança uma exceção específica (`EmailException`) quando um endereço de e-mail inválido é fornecido, garantindo que apenas e-mails válidos sejam armazenados.
- **Representação de String:** Sobrescreve o método `toString()` para fornecer uma representação legível do e-mail, facilitando sua visualização e uso no sistema.

A classe Email é fundamental para o gerenciamento adequado de comunicações eletrônicas, assegurando que as informações sejam precisas e utilizáveis.



## Classe CPFException

A classe `CPFException` é uma exceção personalizada que representa um erro relacionado à validação de CPF (Cadastro de Pessoa Física). Esta exceção é fundamental para o tratamento de erros no sistema, garantindo que os dados do CPF sejam válidos antes de serem processados.

## Classe CNPJException

A classe `CNPJException` é uma exceção personalizada que representa um erro relacionado à validação de CNPJ (Cadastro Nacional da Pessoa Jurídica). Esta classe é crucial para garantir que apenas CNPJs válidos sejam processados no sistema.

## Classe TelefoneException

A classe `TelefoneException` é uma exceção personalizada que lida com erros relacionados à validação de números de telefone. Esta classe é essencial para garantir que apenas telefones válidos sejam aceitos no sistema.

## Classe EmailException

A classe `EmailException` é uma exceção personalizada utilizada para tratar erros relacionados à validação de endereços de e-mail. Ela é importante para assegurar que somente e-mails válidos sejam processados pelo sistema.

## Classe Distribuidora

A classe `Distribuidora` é o ponto de entrada do aplicativo, responsável por inicializar e exibir a interface do usuário. Ela atua como o controlador principal, conectando a lógica de apresentação com as funcionalidades do sistema.

### Principais Funcionalidades:

- **Inicialização da Interface:** Cria uma instância da classe `Interface` e invoca o método `desenha()`, que é responsável por renderizar a interface gráfica do usuário.
- **Organização do Código:** Centraliza a execução do aplicativo, facilitando a manutenção e a compreensão do fluxo principal da aplicação.
- **Integração com Modelos:** Embora a classe em si não contenha lógica de negócios, ela está preparada para interagir com outros componentes do sistema, como a modelagem de usuários e dados.

A classe `Distribuidora` é fundamental para dar início à operação do sistema, garantindo que a interface do usuário seja exibida corretamente e pronta para interagir com as funcionalidades implementadas.