

# Relatório Explicativo - Sistema de Gerenciamento de Entregas

## 1. Descrição do Cenário

O sistema desenvolvido tem como finalidade gerenciar entregas de mercadorias realizadas por motoristas utilizando veículos específicos, para clientes cadastrados. O sistema contempla o cadastro de três entidades principais: **Clientes**, **Motoristas** e **Veículos**.

A operação central é o registro de uma entrega, onde são vinculados:

Um cliente (destinatário),

Um motorista (responsável pela entrega),

Um veículo (utilizado no transporte),

A distância a ser percorrida,

O valor por quilômetro.

O sistema calcula automaticamente o valor total do frete com base nos dados informados.

## 2. Diagrama de Classes UML

A imagem com o diagrama de classes será exibida em instantes assim que finalizada a geração.

O diagrama inclui as seguintes classes:

**Pessoa** (abstrata)

**Cliente** (herda de **Pessoa**)

**Motorista** (herda de **Pessoa**)

**Veiculo**

**Frete** (interface)

**Entrega** (implementa **Frete**)

### 3. Justificativa para as Escolhas de Modelagem

#### Classe Abstrata **Pessoa**

Criada para representar características comuns a **Cliente** e **Motorista**.

Define os atributos **nome**, **cpf** e **telefone**, além do método abstrato **exibirInformacoes()** forçando subclasses a implementarem sua própria forma de exibição.

#### Subclasses **Cliente** e **Motorista**

**Cliente**: adiciona o atributo **endereco**, fundamental para a entrega.

**Motorista**: inclui a **cnh**, essencial para identificar legalmente o condutor.

#### Classe **Veiculo**

Contém dados necessários para identificação e capacidade logística.

Inclui validação para impedir capacidades inválidas ( $\leq 0$ ).

#### Interface **Frete**

Cria um contrato para qualquer entidade que represente um frete.

Permite futura expansão para outros tipos de frete com regras diferentes.

#### Classe **Entrega**

Centraliza os dados de uma entrega.

Relaciona-se diretamente com `Cliente`, `Motorista` e `Veiculo`, garantindo coesão e reuso.

Implementa `Frete` com o método `calcularValor`, simplificando a lógica de custo.

### **Classe `Main`**

Responsável pela interação com o usuário.

Controla os fluxos de cadastro e listagem com menus simples no terminal.