

# Diagrama de Classes

O **Diagrama de Classes** é uma das principais representações na **UML (Unified Modeling Language)**, que modela a estrutura de um sistema orientado a objetos. Ele descreve as classes, seus atributos, métodos e os relacionamentos entre elas, ajudando a visualizar a arquitetura do software e suas interações.

## 1. Elementos de um Diagrama de Classes

Um diagrama de classes é composto por:

- **Classes:** Representadas por um retângulo dividido em três seções:
  - **Nome da Classe:** A parte superior contém o nome da classe.
  - **Atributos:** A parte do meio descreve os atributos (variáveis de instância) da classe.
  - **Métodos:** A parte inferior lista os métodos (comportamentos) da classe.
- **Relacionamentos:** Os tipos mais comuns de relacionamento entre as classes são:
  - **Associação:** Representa a relação entre duas ou mais classes, mostrando que uma instância de uma classe pode se referir a outra.
  - **Agregação:** É uma forma de associação em que uma classe contém uma referência a outra, mas a existência de uma não depende da outra.
  - **Composição:** Tipo mais forte de agregação, onde a classe "container" controla a existência dos objetos.
  - **Herança (Generalização):** Mostra que uma classe filha herda de uma classe pai.
  - **Dependência:** Uma classe depende de outra para realizar suas operações.
  - **Interface:** Conjunto de métodos que uma classe deve implementar. Classes podem implementar várias interfaces.

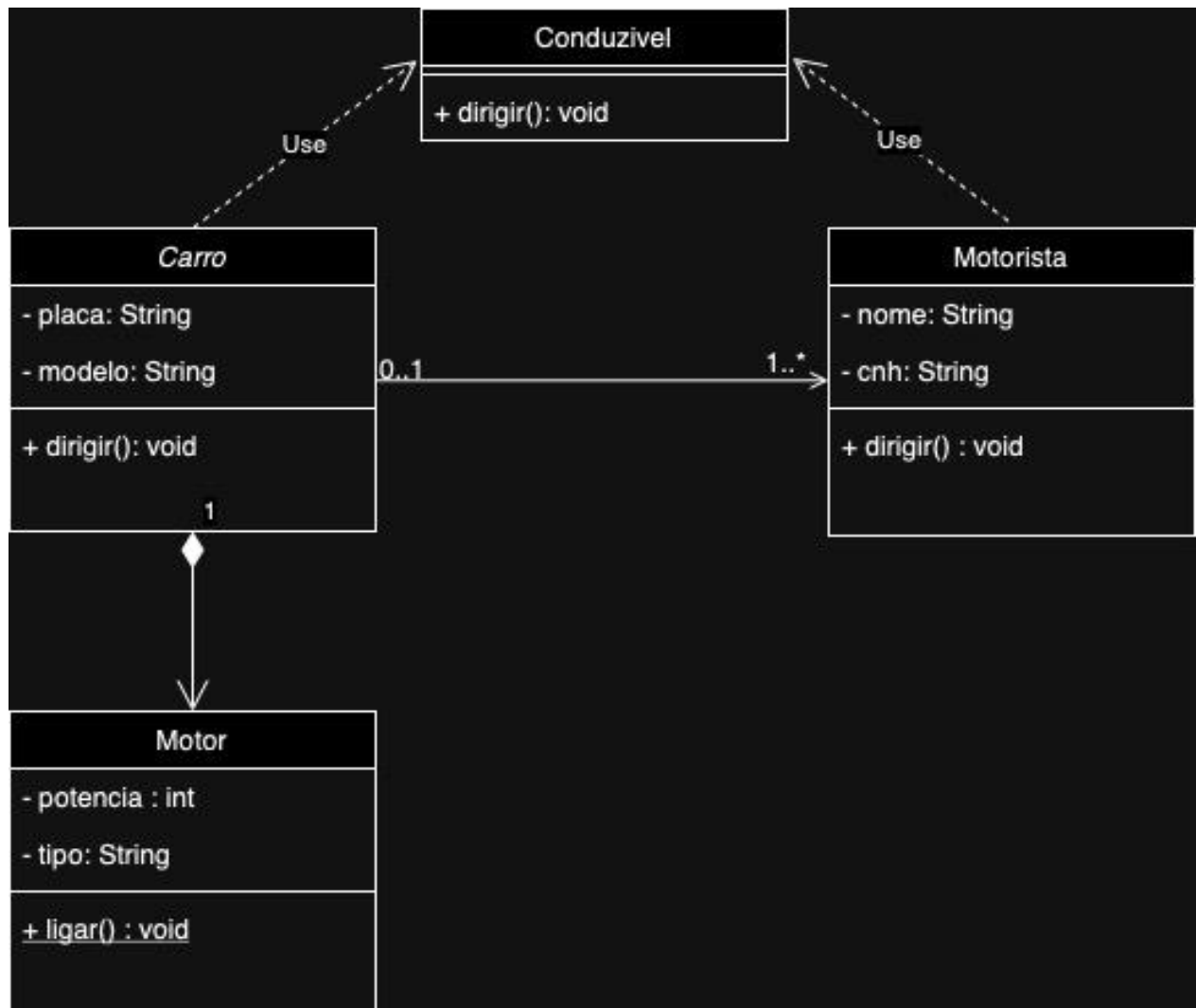
## 2. Componentes de um Diagrama de Classes

- **Visibilidade dos atributos e métodos:** Mostra o acesso permitido para o atributo/método.
  - **+ público:** Acesso permitido a todas as classes.
  - **# protegido:** Acesso permitido à classe, suas subclasses e classes no mesmo pacote.
  - **- privado:** Acesso permitido apenas à própria classe.
- **Tipos de Relacionamentos:**
  - **Associação:** Relacionamento direto entre duas classes.
  - **Agregação:** Representada por um losango vazio, indicando uma relação "tem-um".
  - **Composição:** Um losango cheio, onde uma classe "possui" e controla a existência de outra.
  - **Generalização:** Linha sólida com uma seta vazada, representando herança.
  - **Realização:** Representa a implementação de uma interface.

### 3. Exemplo de Diagrama de Classes

Vamos modelar um sistema de gestão de veículos e motoristas. O diagrama terá as classes **Carro**, **Motorista**, **Motor**, e a interface **Conduzível**. O relacionamento principal será uma composição entre o **Carro** e o **Motor** (já que o motor não existe sem o carro), além de uma agregação entre o **Carro** e o **Motorista** (um motorista pode conduzir vários carros, mas existe independentemente dos carros).

#### Diagrama de Classes



- **Carro:** Tem atributos como placa e modelo, e possui um motor por composição.
- **Motor:** Não pode existir sem um carro (composição).
- **Motorista:** Pode dirigir vários carros (agregação), mas também pode existir sem um carro.
- **Conduzível:** Interface que define o comportamento "dirigir" que tanto **Motorista** quanto **Carro** devem implementar.

## 4. Como Usar Diagramas de Classes

**Projeto de Software:** Diagramas de classes ajudam a visualizar a arquitetura do sistema antes da implementação, definindo os principais componentes e como eles interagem.

**Comunicação:** Eles são uma ferramenta essencial para facilitar a comunicação entre desenvolvedores e outros stakeholders sobre a estrutura do software.

**Documentação:** Servem como uma documentação visual do código, útil para a manutenção do sistema.

**Refatoração:** Quando um sistema precisa de melhorias, o diagrama ajuda a identificar áreas que necessitam de ajustes.