

5 – O padrão UML

5.1. Domínio “do problema”

5.2. Abordagem / paradigma estrutural (REVISÃO)

5.3. Abordagem / paradigma OO (Concertação)

5.4. A UML e o Diagrama de Classes

5.4.1. O Básico (do básico) Sobre UML

5.4.2. O diagrama de Classes (básico)

5.5. Relacionamentos entre Classes

5.5.1. Associação

5.5.2. Generalização

5.5.3. Agregação

5.5.4. Composição



5.1. Domínio de problema



Figura 3.1: Domínio bancário

5.1. Domínio de problema

- ▶ Suponha que o nosso problema requeira que se desenvolva um Software de gestão duma escola. Como entidades do **domínio desse problema** consideraríamos, alunos, professores, matriculas, avaliações e disciplinas por exemplo.

5.2. Abordagem / paradigma estrutural (RESUMO)

- ▶ Um programa era a entidade dum certo domínio;
- ▶ Um atributo era uma variável que pertencia a uma classe;
- ▶ Os dados dessa classe eram armazenados nos seus atributos.

5.3. Abordagem / paradigma OO

- ▶ Uma aplicação orientada a objectos é composta por objectos;
- ▶ Em geral, um **objecto** representa uma **entidade** do domínio;
- ▶ Um atributo é uma variável que pertence a um objecto;
- ▶ Os dados de um objecto são armazenados nos seus atributos.

5.3. Abordagem / paradigma OO

- ▶ **As tarefas** que um objecto pode realizar são definidas pelos seus **métodos**;
- ▶ Um objeto é composto por atributos e métodos;
- ▶ Os objectos não representam apenas coisas **concretas** como os alunos da escola. Eles também devem ser utilizados para representar coisas **abstractas** como uma matrícula de um aluno ou um serviço que a escola ofereça.

5.3. Abordagem / paradigma OO

► Classe

- É um tipo estruturado que pode conter (membros):
 - Atributos (dados / campos)
 - Métodos (funções / operações)
- A classe também pode prover muitos outros recursos, tais como:
 - Construtores
 - Sobrecarga
 - Encapsulamento
 - Herança
 - Polimorfismo
- Exemplos:
 - Entidades: Produto, Cliente, Triangulo
 - Serviços: ProdutoService, ClienteService, EmailService, StorageService
 - Controladores: ProdutoController, ClienteController
 - Utilitários: Calculadora, Compactador
 - Outros (views, repositórios, gerenciadores, etc.)

5.4. UML e o Diagrama de Classes

- ▶ Básico (do básico) Sobre UML
- ▶ O diagrama de Classes

5.4.1. O Básico (do básico) Sobre UML

- ▶ Basicamente, **UML** (**U**nified **M**odeling **L**anguage) é uma linguagem de notação (um jeito de escrever, ilustrar, comunicar) para uso em projetos de sistemas.
- ▶ Esta linguagem é expressa através de diagramas. Cada **diagrama** é composto por **elementos** (formas gráficas usadas para os desenhos) que **possuem relação** entre si.

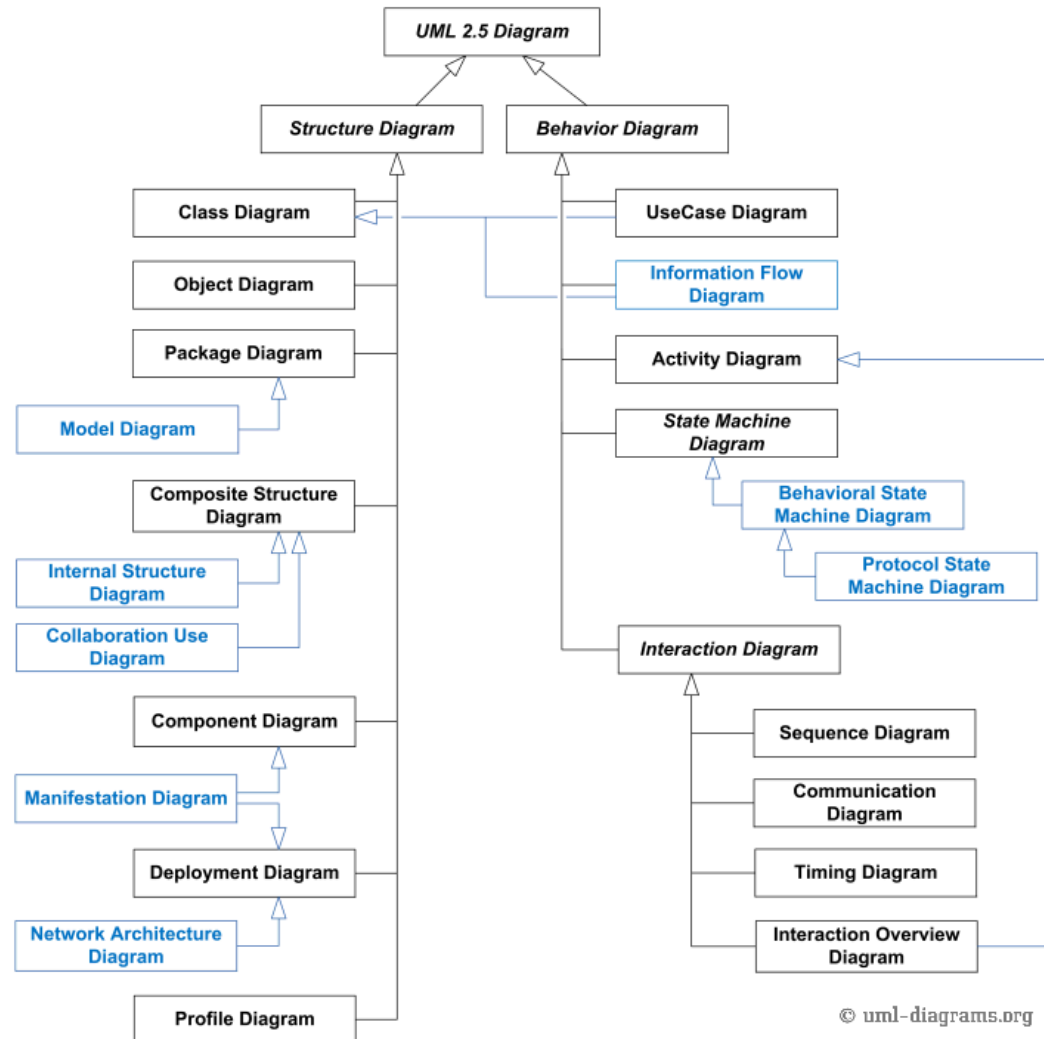
5.4.1. O Básico (do básico) Sobre UML

- ▶ Os diagramas da UML se dividem em dois grandes grupos: diagramas **estruturais** e diagramas **comportamentais**.
- ▶ **Diagramas estruturais** devem ser utilizados para especificar detalhes da estrutura do sistema (parte estática), por exemplo: classes, métodos, interfaces, namespaces, serviços, como componentes devem ser instalados, como deve ser a arquitetura do sistema etc.

5.4.1. O Básico (do básico) Sobre UML

- ▶ **Diagramas comportamentais** devem ser utilizados para especificar detalhes do comportamento do sistema (parte dinâmica), por exemplo: como as funcionalidades devem funcionar, como um processo de negócio deve ser tratado pelo sistema, como componentes estruturais trocam mensagens e como respondem às chamadas etc.

5.4.1. O Básico (do básico) Sobre UML



© uml-diagrams.org

5.4.1. O Básico (do básico) Sobre UML

UML ajuda muito a deixar o escopo claro, pois centraliza numa única visão (o diagrama) um determinado conceito, utilizando uma linguagem que todos os envolvidos no projeto podem facilmente entender.

- ▶ Desde que **utilizada na medida certa**, ou seja, apenas **quando realmente é necessário**.
- ▶ O maior problema na produção de software, a maior dor, em qualquer país do mundo é a **falha na comunicação**.
- ▶ Vejamos um rápido exemplo **didático** de como se dá a comunicação em equipes de produção de software:

5.4.1. O Básico (do básico) Sobre UML

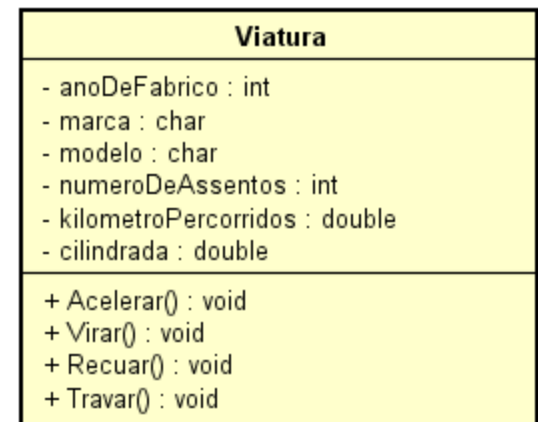
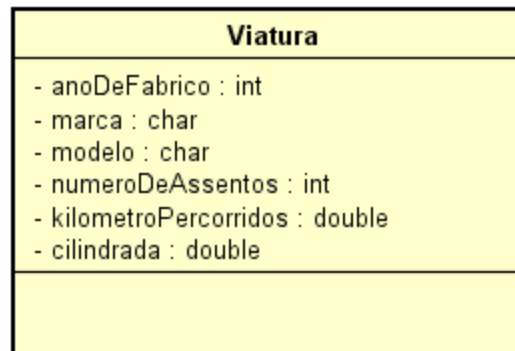
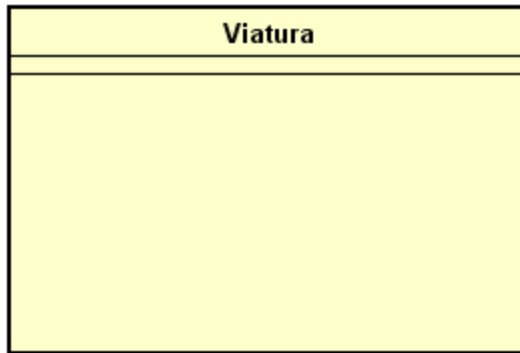
Vejam os um rápido exemplo **didático** de como se dá a comunicação em equipes de produção de software:

”

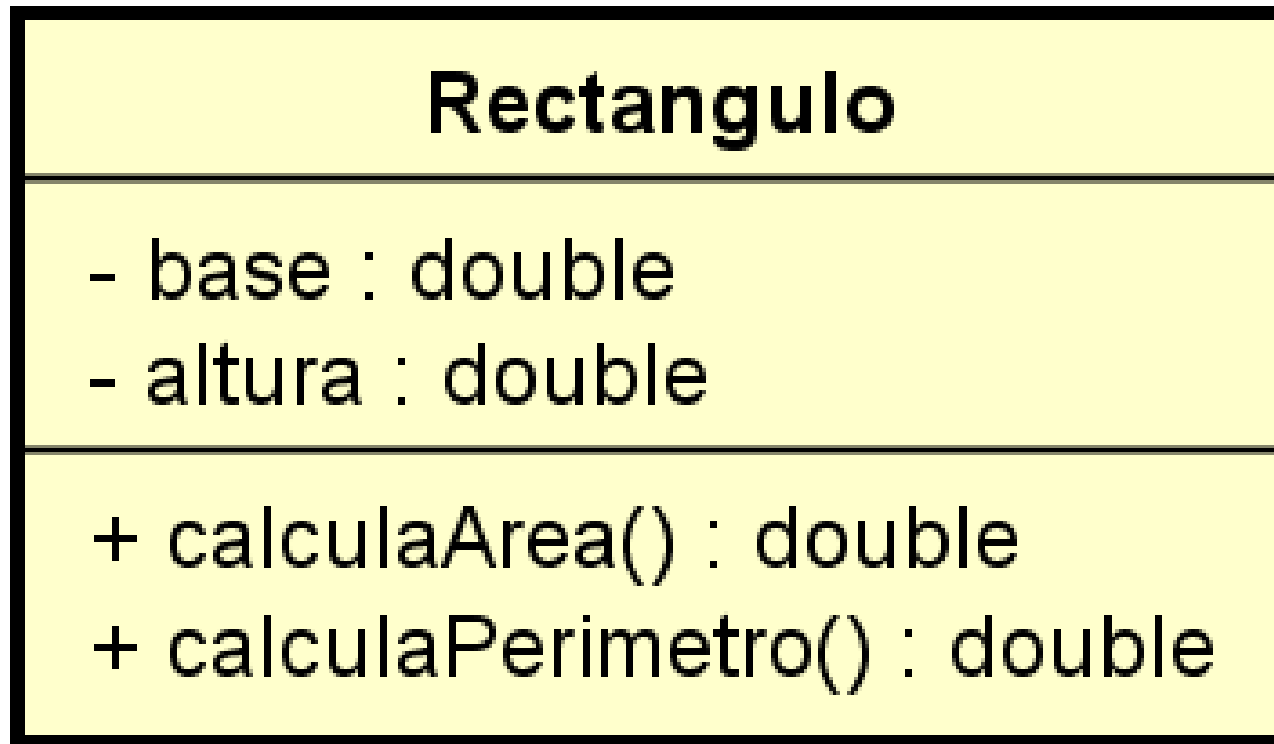
/ João quer A, explica à equipe algo “parecido” com B. Marcos entende que João quer C, e explica para Claudia que é para fazer D. Claudia faz um “D que mais se parece um E”, e entrega um “meio E” para João. E João queria um A... */*

5.4.1. O Básico (do básico) Sobre UML

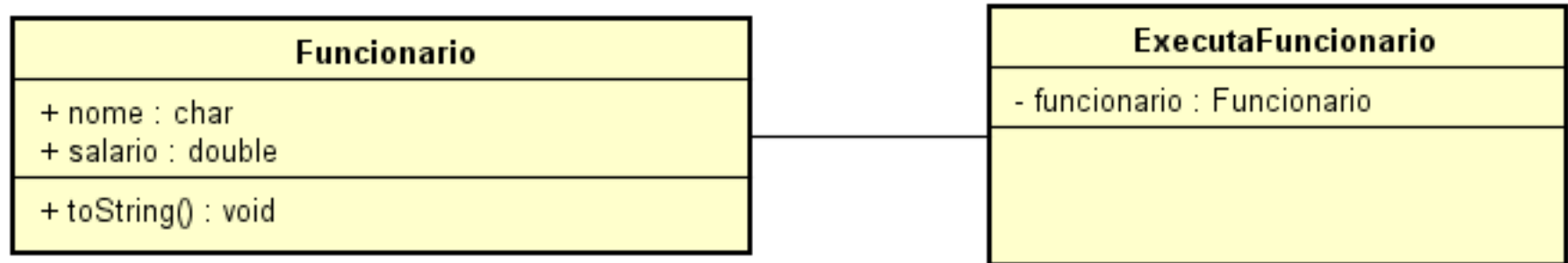
5.4.2. O diagrama de Classes ([astah](#))



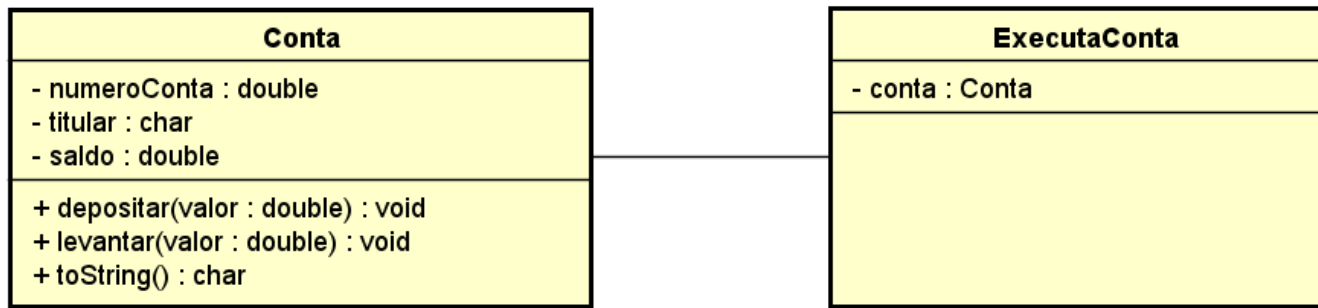
5.4.2. O diagrama de Classes ([astah](#))



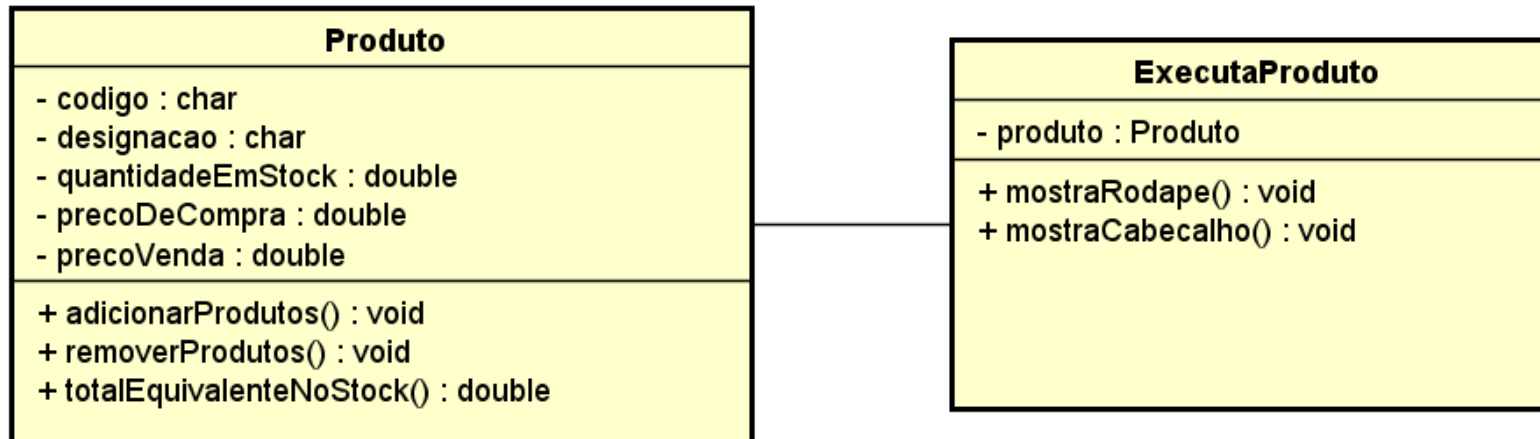
5.4.2. O diagrama de Classes ([astah](#))



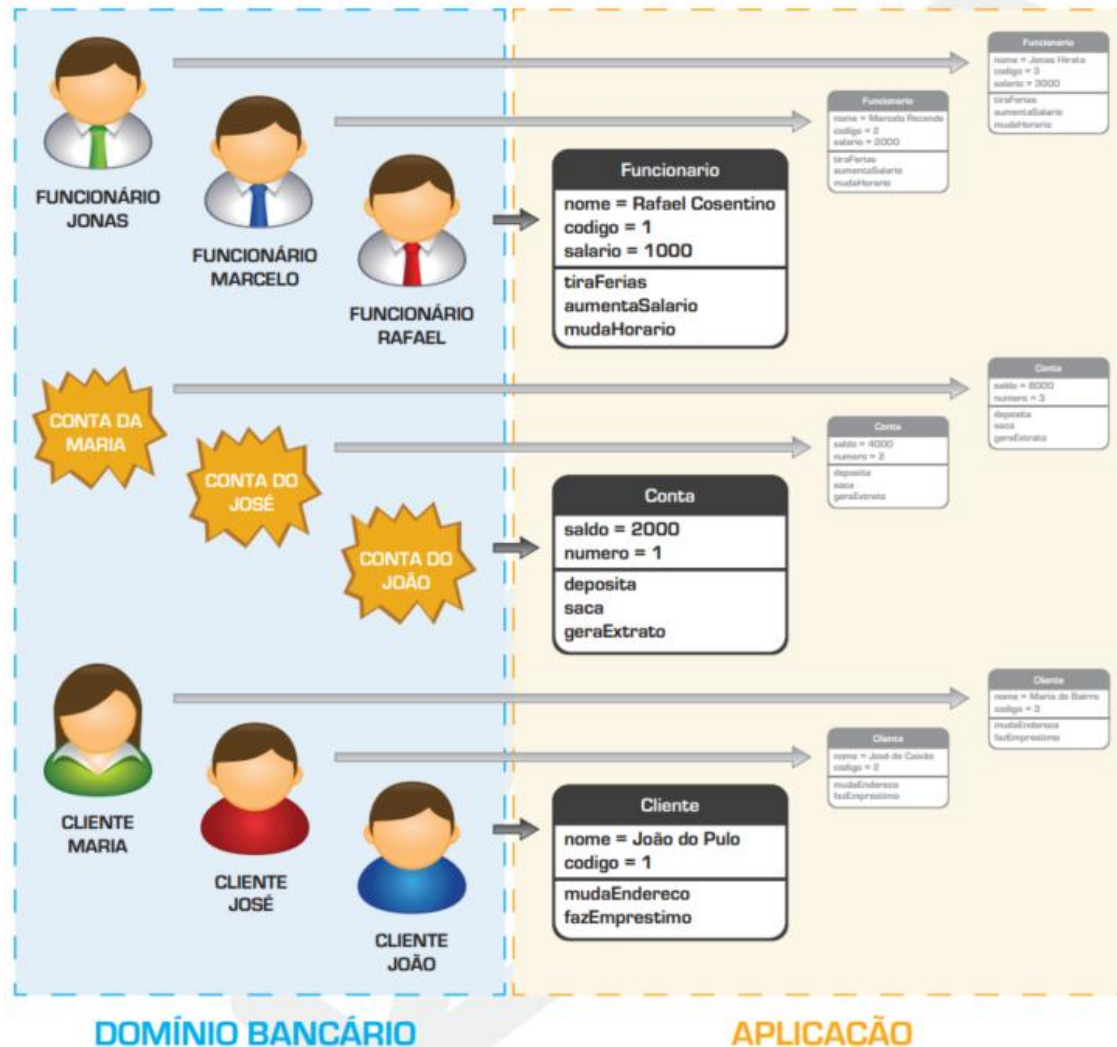
5.4.2. O diagrama de Classes ([astah](#))



5.4.2. O diagrama de Classes ([astah](#))

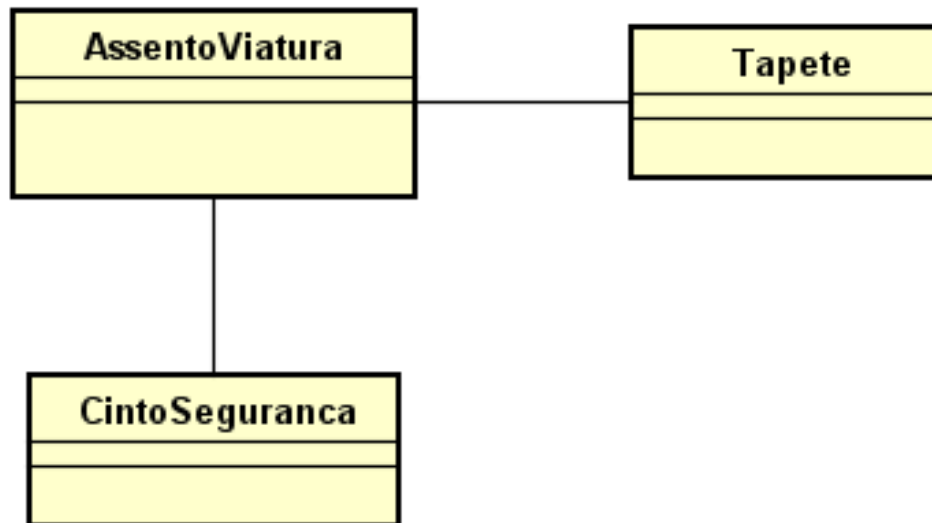


1.4. Mapeamento Domínio - Aplicação

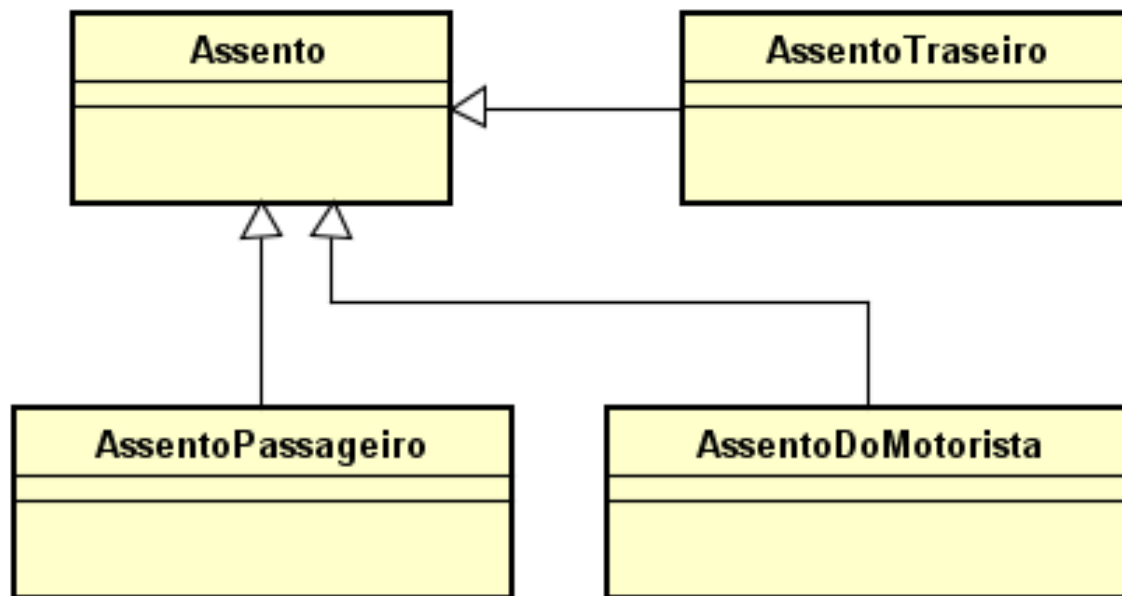


5.5. Relacionamento entre Classes

5.5.1. Associação



5.5.2. Generalização



5.5.3. Agregação

5.5.4. Composição

Aula concluída com Sucesso!



Bibliografia auxiliar – consultada

- ▶ <https://www.ateomomento.com.br/diagramas-uml/>
- ▶ <https://baciotti.silvrback.com/o-analista-de-testes-e-o-analista-de-requisitos>