



Orientação a Objetos

UML - Agregação e Composição

Prof. Lucas Boaventura



UML - Unified Modeling Language



- Linguagem de modelagem para representar um sistema de maneira padronizada.
- **Origem:**
 - compilação de “melhores práticas” de engenharia na modelagem de sistemas complexos (1995).
- Não é uma metodologia de desenvolvimento de software.
- Mantida pelo OMG - Object Management Group
<http://www.uml.org>

[CERTIFICATION](#)[TRAINING](#)[RESOURCES](#)[VENDORS](#)

UNIFIED CELEBRATING 25 YEARS OF UML 1.1

One of the key figures in the software development community, Grady Booch describes the "grand journey" that led to the development of OMG UML standard.

[WATCH VIDEO](#)[25 YEARS
ANNIVERSARY](#)[WHAT IS
UML?](#)[UML
VENDOR](#)[UML
SPECIFICATIONS](#)[UML
CERTIFICATION](#)[TRAINING
PAGE](#)

UML - Unified Modeling Language



- **Objetivos da UML:**
 - Especificação, documentação e estruturação para sub-visualização e maior visualização lógica do desenvolvimento completo de um sistema de informação.
- **UML:**
 - Modelos (escritos)
 - Diagramas (14 tipos) divididos em 2 grupos
 - Estáticos (estruturais)
 - Dinâmicos (comportamental)

UML - Unified Modeling Language



- **Diagramas Estruturais:**

- Diagrama de classes
- Diagrama de objetos
- Diagrama de componentes
- Diagrama de instalação
- Diagrama de pacotes
- Diagrama de estrutura composta
- Diagrama de perfil

- **Diagramas**

- Comportamentais:**

- Diagrama de caso de uso
- Diagrama de estados
- Diagrama de atividade
- Diagrama de sequência
- Diagrama de interatividade
- Diagrama de colaboração
- Diagrama de tempo

UML - Unified Modeling Language



- Diagramas Estruturais:

- Diagrama de classes
- Diagrama de objetos
- Diagrama de componentes
- Diagrama de instalação
- Diagrama de pacotes
- Diagrama de estrutura composta
- Diagrama de perfil

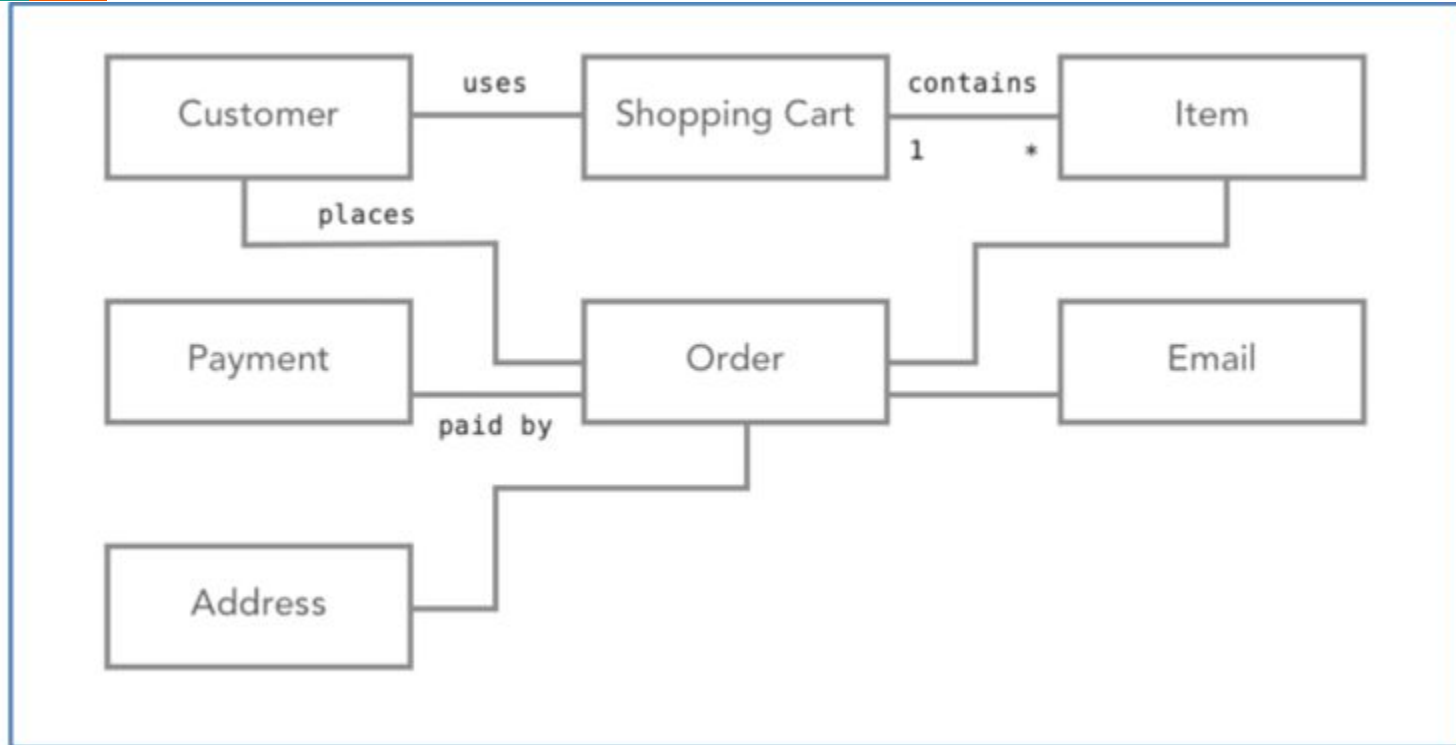
- Diagramas Comportamentais:

- Diagrama de caso de uso
- Diagrama de estados
- Diagrama de atividade
- Diagrama de sequência
- Diagrama de interatividade
- Diagrama de colaboração
- Diagrama de tempo

Exemplo: Diagrama de objetos



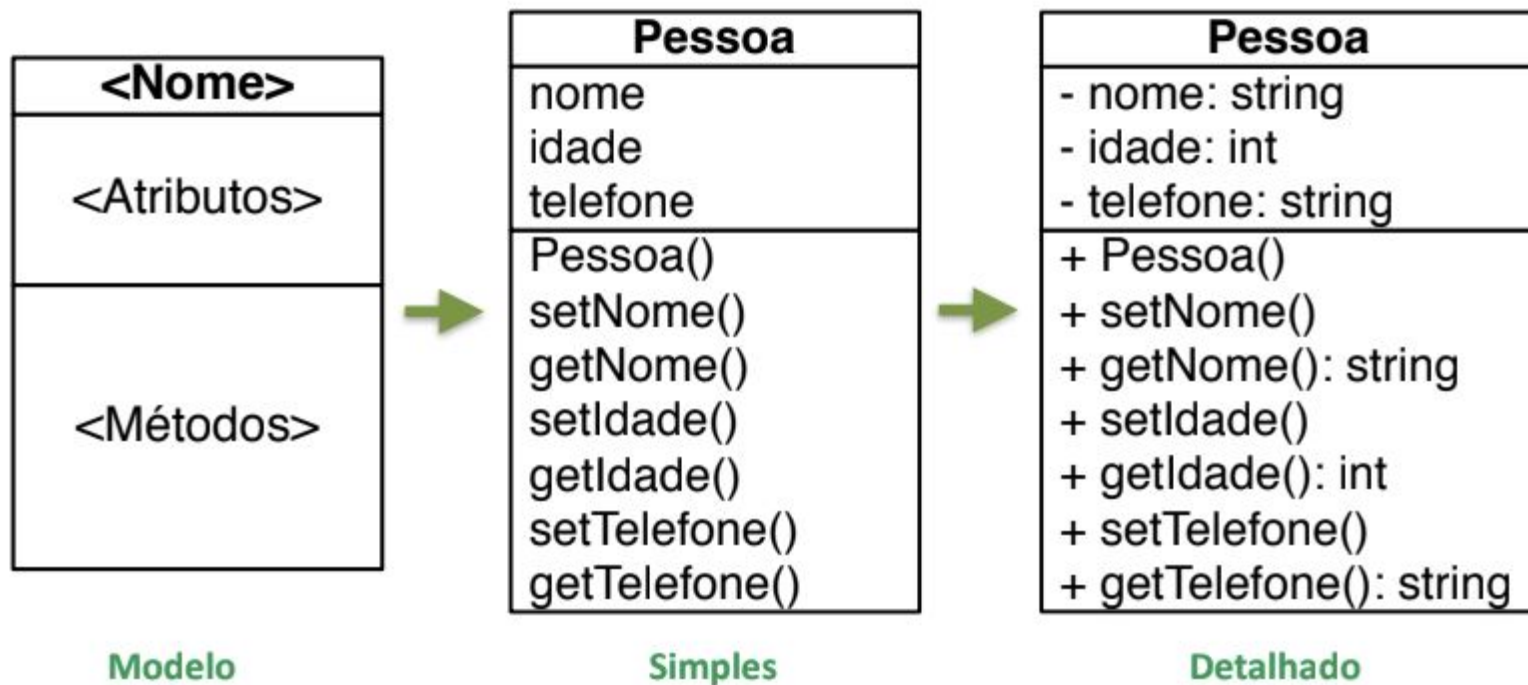
Exemplo: Diagrama de objetos



Exemplo: Diagrama de Classe



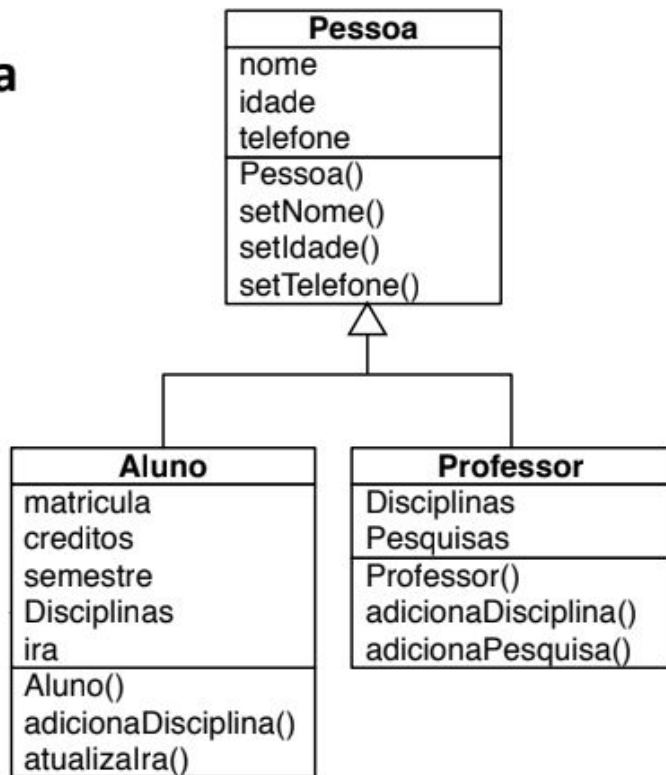
- Classe





Exemplo: Diagrama de Classe

- Classe: Herança



Associação



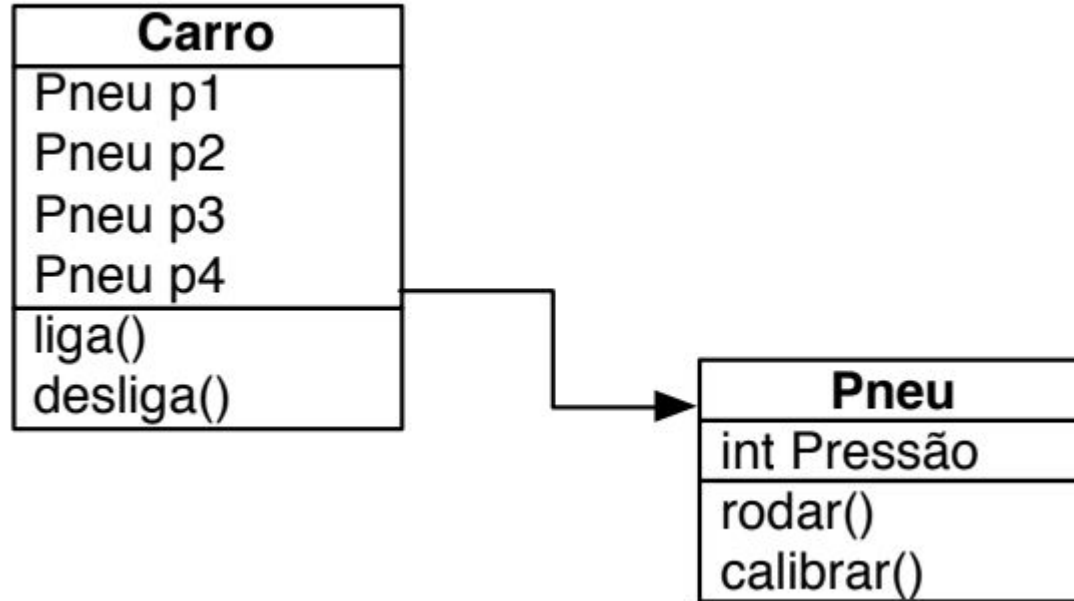
- Associação é um tipo genérico de **relação entre classes**.
- Permite que a instância de um objeto realize uma ação de outro objeto em sua execução.
- Na prática, ocorre quando uma classe possui atributos do tipo de outra classe.

Herança e Associação



- Herança:
 - relação É UM(A) ...
 - extensão
 - Unidirecional (a classe filha é um da classe pai) e não o inverso
- Associação:
 - relação TEM UM(A) ...

Associação



Agregação



- É um tipo de **Associação fraca**.
- Ocorre quando uma classe usa outras classes em suas operações. As classes utilizadas participam da classe principal, mas a classe principal não contém estas classes utilizadas como sendo suas.
- Relação Todo / Parte. Neste caso a parte é compartilhada por outros, portanto existe sem o todo.

Agregação

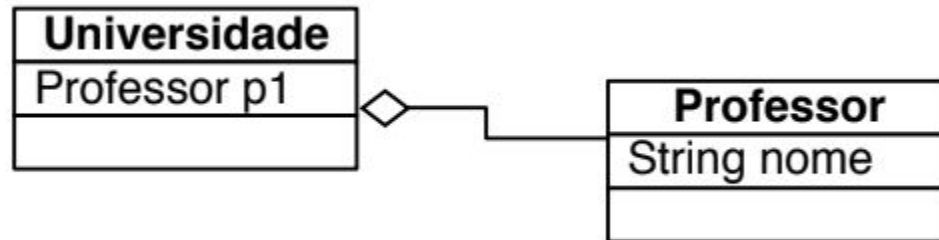


- Na agregação, os objetos são criados fora da classe e são passados como referência para serem usados.
- Um objeto contém uma lista de outros objetos.
- Os objetos contidos podem existir sem serem parte do objeto que os contém.

Agregação



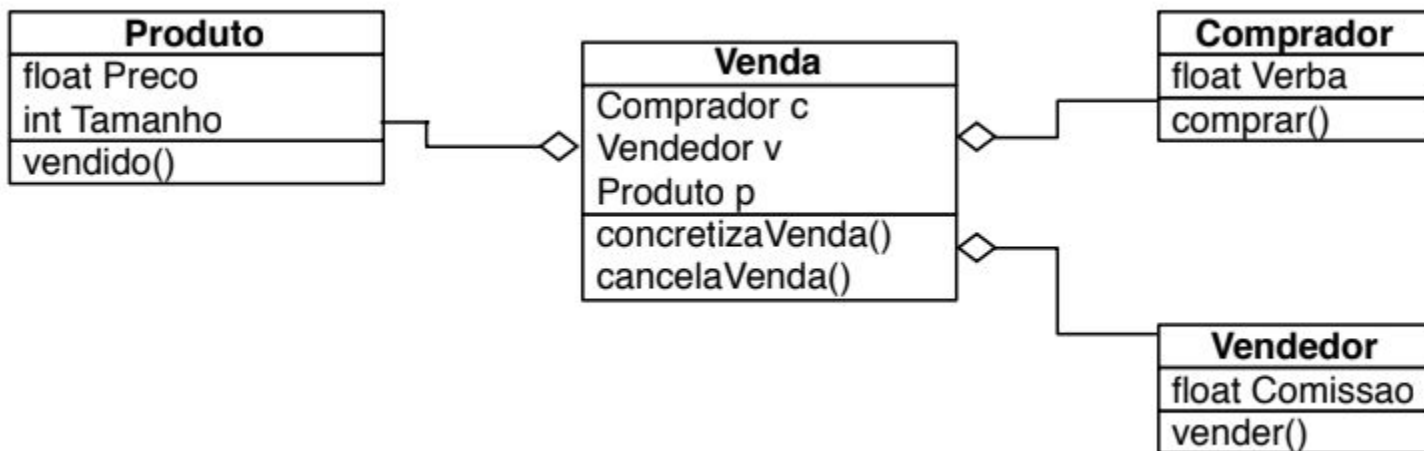
- **Ex 1:** Uma Universidade utiliza ou contém Professores. Caso a universidade deixe de existir, o professor ainda existe e pode ir para outra universidade.
- Portanto, a vida de uma objeto não está necessariamente atrelada ao outro.





Agregação - UML

- Ex 2: Venda (UML)



Agregação

- Ex 2: Venda

```
public class Vendedor {
    float Comissao;
    void vende() {
        System.out.println("Vendido");
    }
}

public class Venda {
    Comprador c;
    Vendedor v;
    Produto p;
    void concretizaVenda() {
        System.out.println("Venda efetuada");
        c.Verba -= p.Preco;
        v.Comissao += p.Preco * 0.1f;
        p.vendido();
    }
    void cancelaVenda() {
        System.out.println("Venda cancelada");
    }
}

public class Comprador {
    float Verba;
    void compra() {
        System.out.println("Comprado");
    }
}

public class Produto {
    float Preco;
    int Tamanho
    void vendido() {
        System.out.println("Vendido");
    }
}
```



Composição

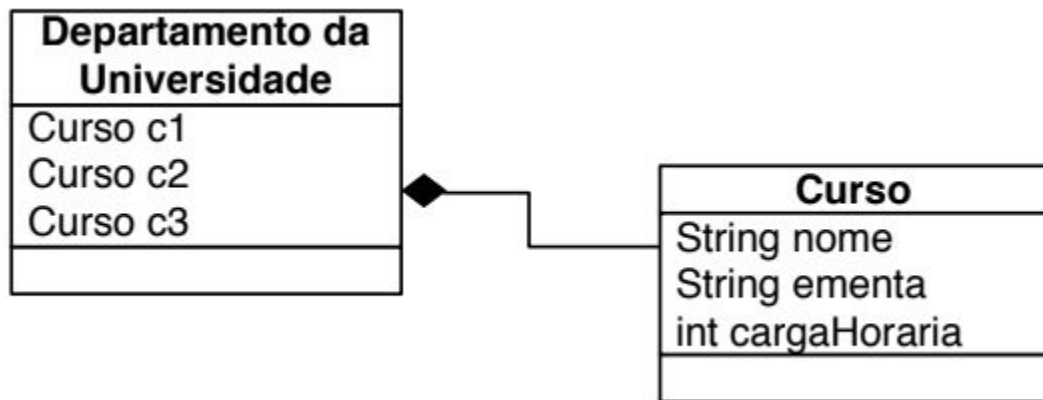


- É um tipo de **Associação forte**.
- Neste caso uma classe se torna dona de outra.
- Sua principal diferença para a agregação é que na composição ocorre que quando o objeto todo deixar de existir os seus objetos partes deverão deixar de existir também.



Composição

- **Ex 1:** Um departamento da universidade contém cursos. Neste caso ele é dono dos cursos. Caso o departamento deixe de existir, os cursos também deixarão de existir.





Composição

- Ex 2: Banco (UML)

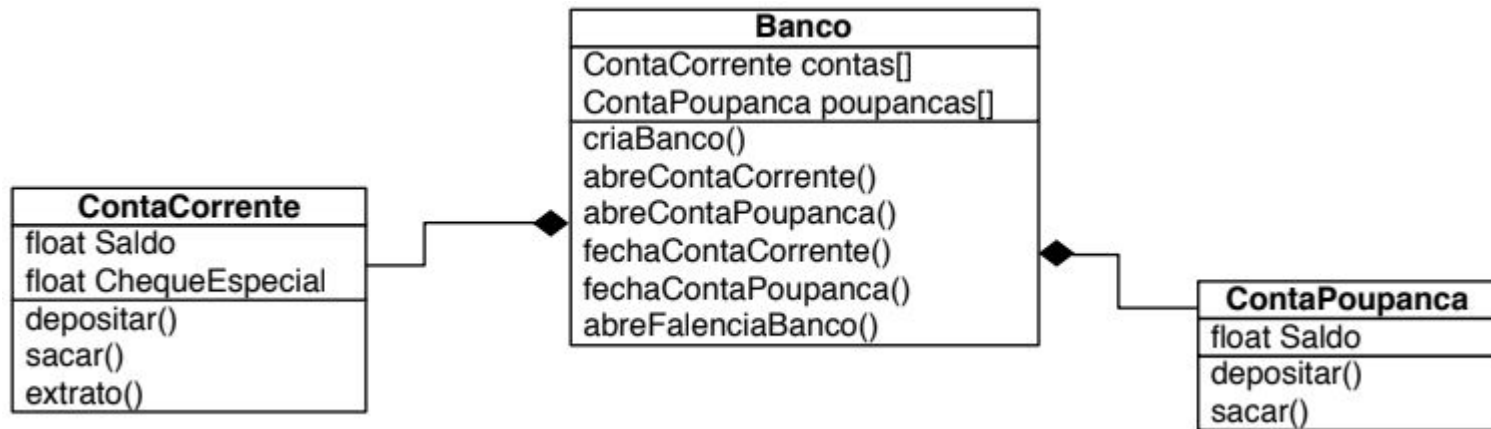


Diagrama de Classe

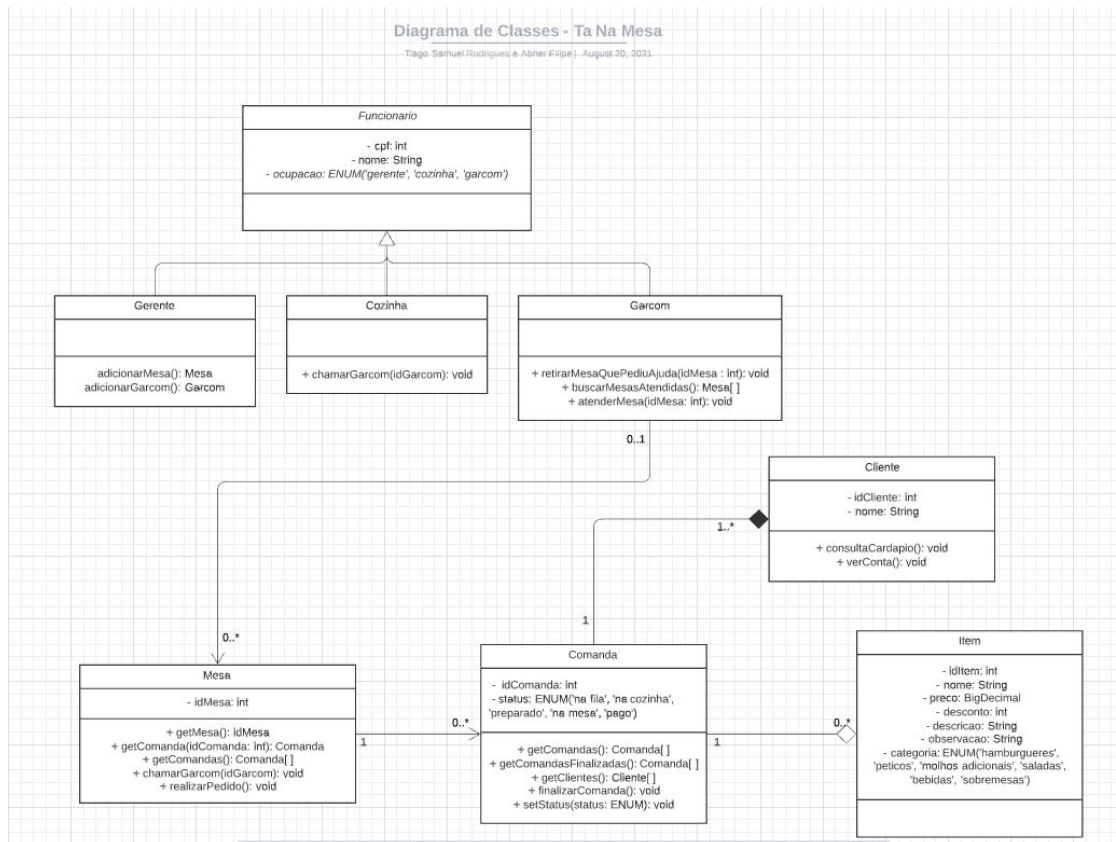


Diagrama de Casos de Uso

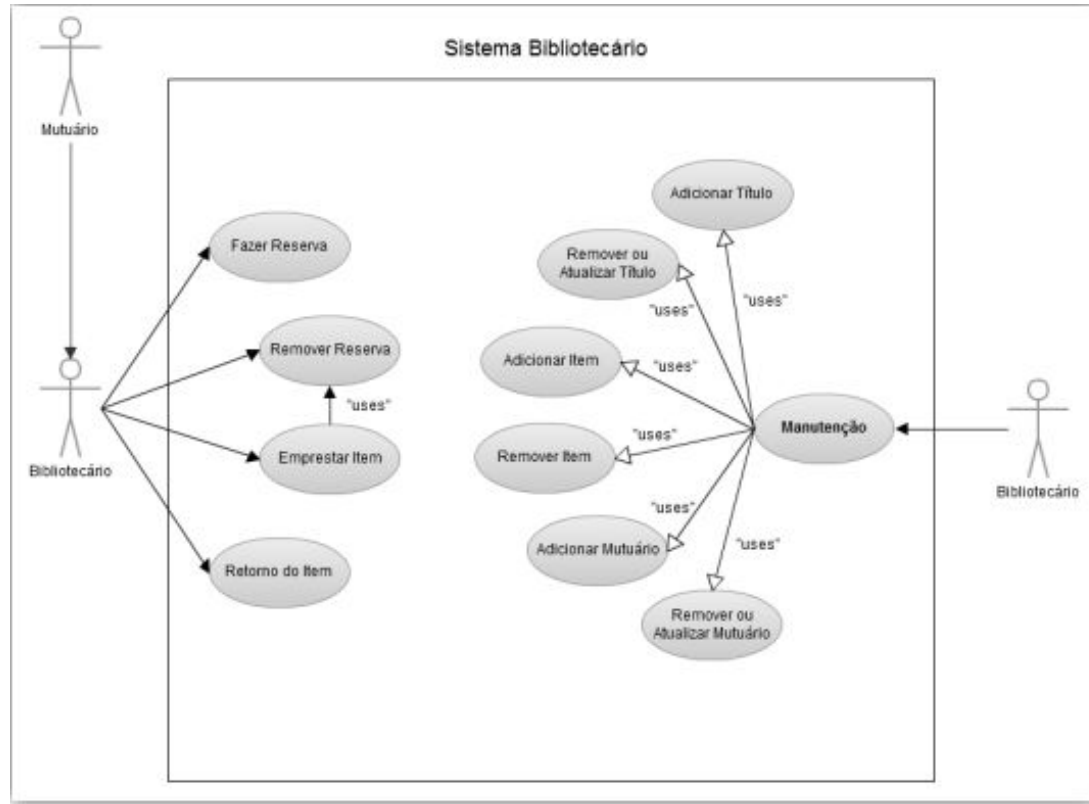
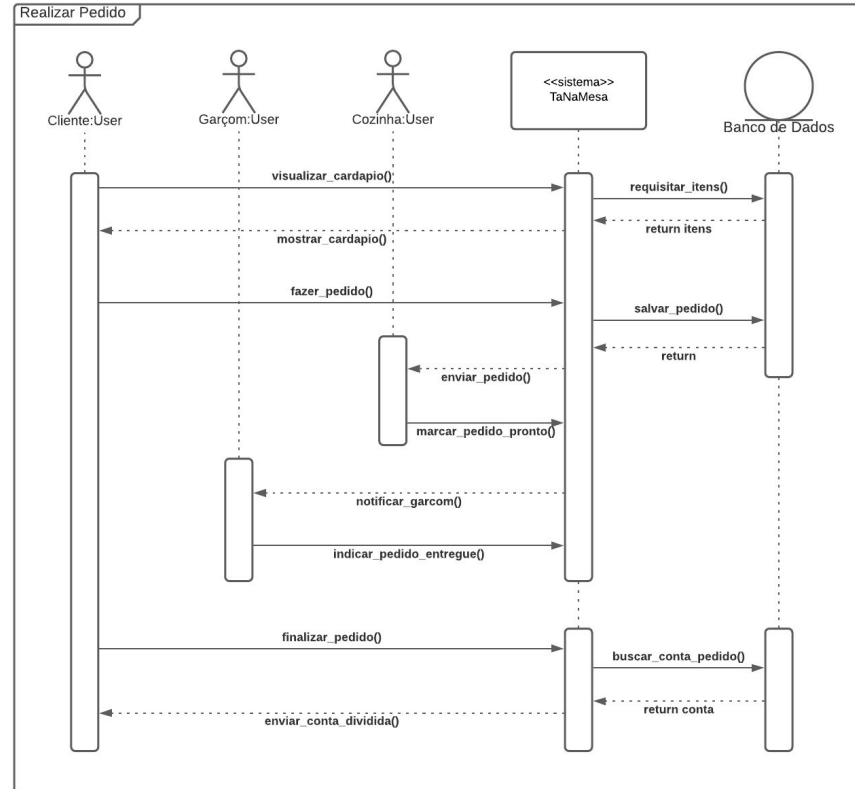


Diagrama de Sequência



UML - Agregação e Composição



Dúvidas?