Diagrama de Classe

Alunos:

Ana Carolina Carvalho da Silva- 15/0116519

Arthur Barbosa Diniz - 15/0118457

Guilherme Augusto Nunes Silva -15/0128134

Ícaro Pereira de Oliveira - 15/0129807

Weyler Almeida Gomes - 14/0053298

Agenda

- O que é UML e porque usar?
- Diagrama de Classe
- O que é Diagrama de Classe?
- Qual a sua utilidade?
- As três perspectivas básicas
- Implementação
- Estrutura da Classe
- Relacionamentos

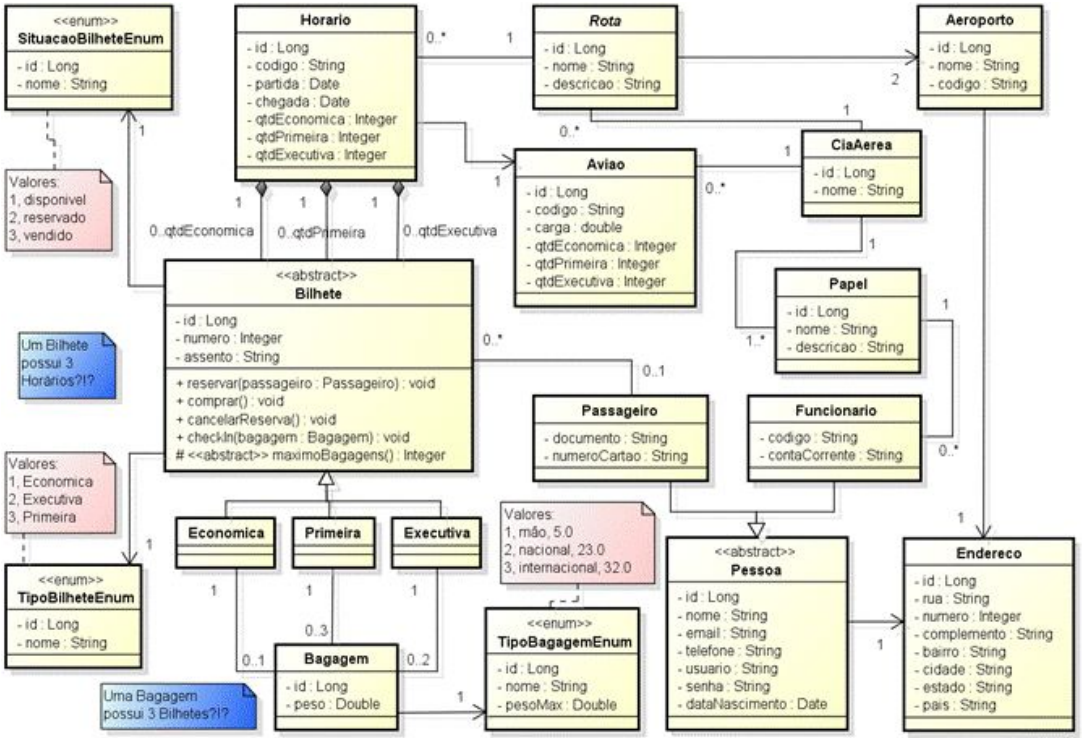
O que é UML e porque usar?

- UML - Unified Modeling Language
- Definição

> "É uma família de notações gráficas, apoiada por um metamodelo único, que ajuda na descrição e no projeto de sistemas de software, particularmente daqueles construídos utilizando o estilo orientado a objetos"

- Por que usar UML?

Diagrama de Classe



O que é Diagrama de Classe?

- Um **diagrama de classes** é uma representação da estrutura e relações das classes que servem de modelo para objetos.

- É uma modelagem muito útil para o desenvolvimento de sistemas, pois define todas as classes que o sistema necessita possuir e é a base para a construção dos diagramas de comunicação, sequência e estados.

Fonte: https://pt.wikipedia.org/wiki/Diagrama_de_classes

Qual a sua utilidade?

- Define todas as classes que o sistema necessita possuir

- É a base para a construção dos diagramas de outros diagramas, como os de comunicação, sequência e estados.

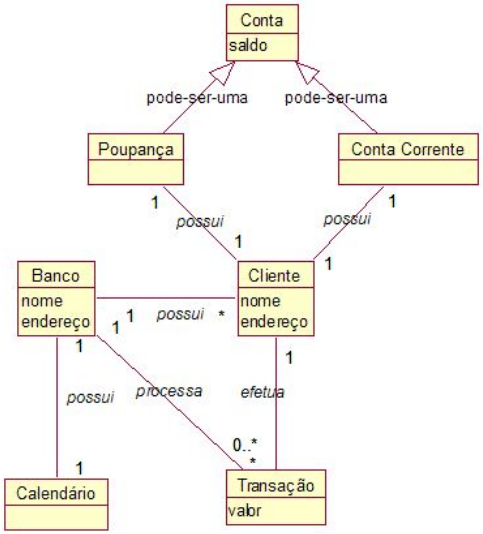
As três perspectivas básicas

- Conceitual
- Especificação
- Implementação

Conceitual

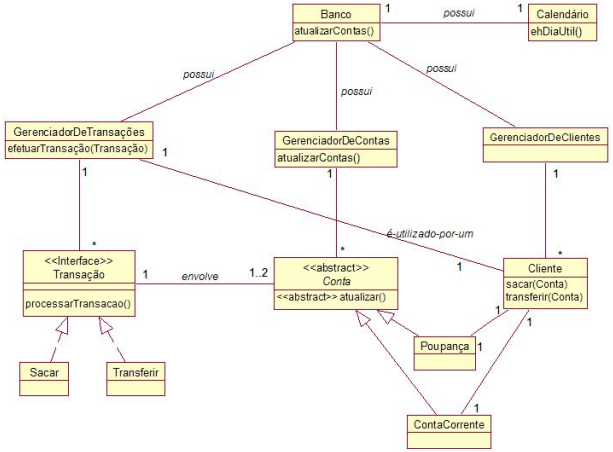
- Representa conceitos do domínio em estudo

- Perspectiva ao cliente



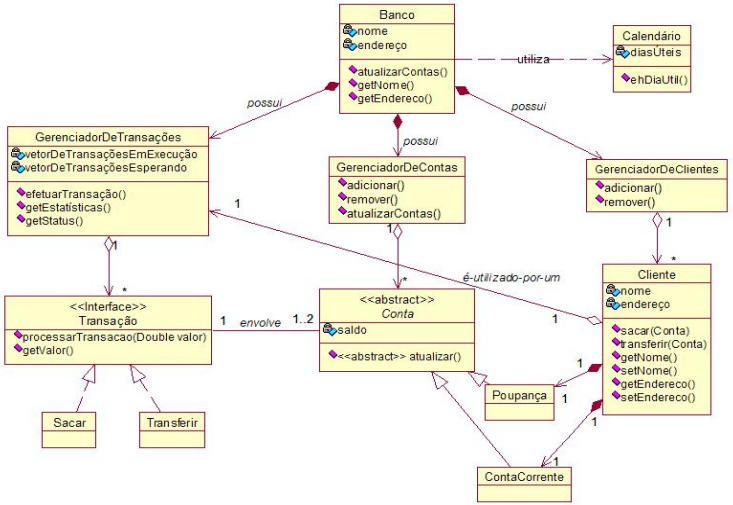
Especificação

- Foco nas interfaces da arquitetura, nos principais métodos, e não como eles irão ser implementados.
- Perspectiva destinada às pessoas que não precisam saber detalhes de desenvolvimento



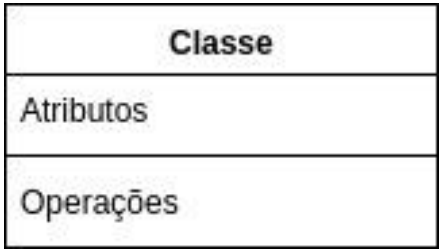
Implementação

- Aborda vários detalhes de implementação, tais como navegabilidade, *tipo* dos atributos, etc.
- Perspectiva destinada ao time de desenvolvimento.



Estrutura da Classe

- Nome Classe
- Atributos
- Operações
- Visibilidade

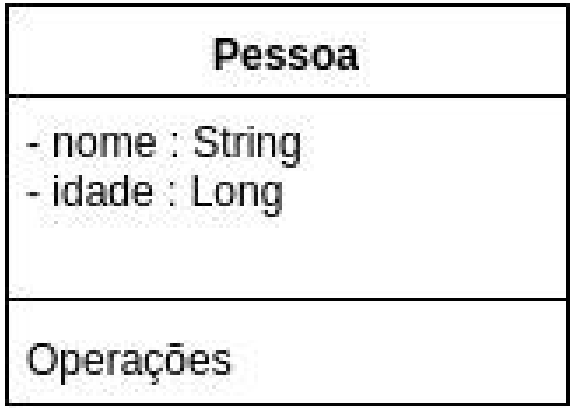


OU

Atributos

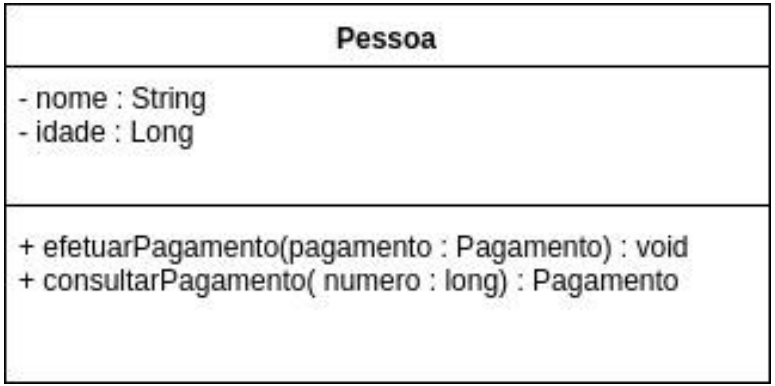
- Um atributo é formado por:

  visibilidade nome : tipo[multiplicidade] = valor inicial { propriedades }



Operações

- Uma operação é formada por:
- visibilidade nome(parâmetros) : tipo de retorno {propriedades}



Relacionamentos

- Associação
- Agregação
- Composição
- Composição x Agregação
- Classe de associação

Associação

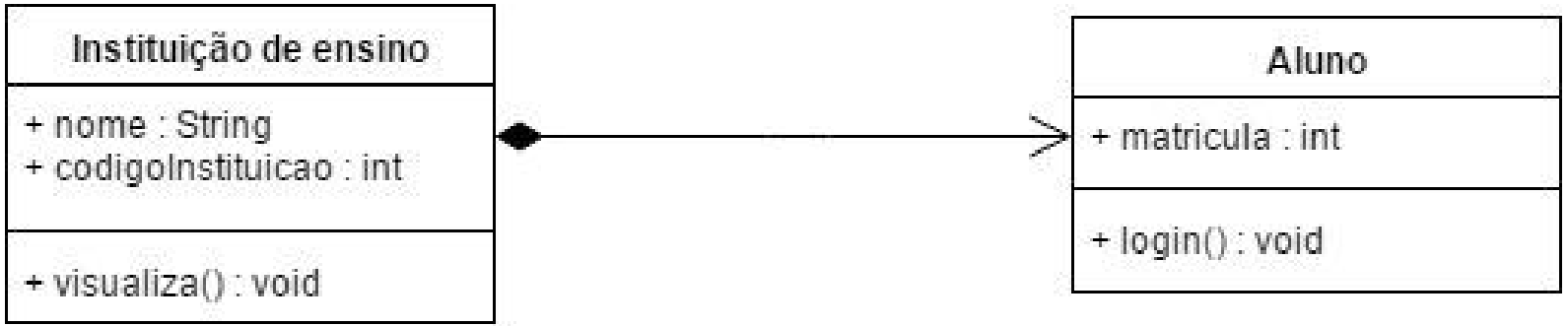
- Relacionamento simples entre duas classes:

Agregação

- Informa que uma classe faz parte de outra classe, mas não de forma exclusiva.

Composição

- Informa que uma classe faz parte de outra classe de forma exclusiva.



Agregação x Composição

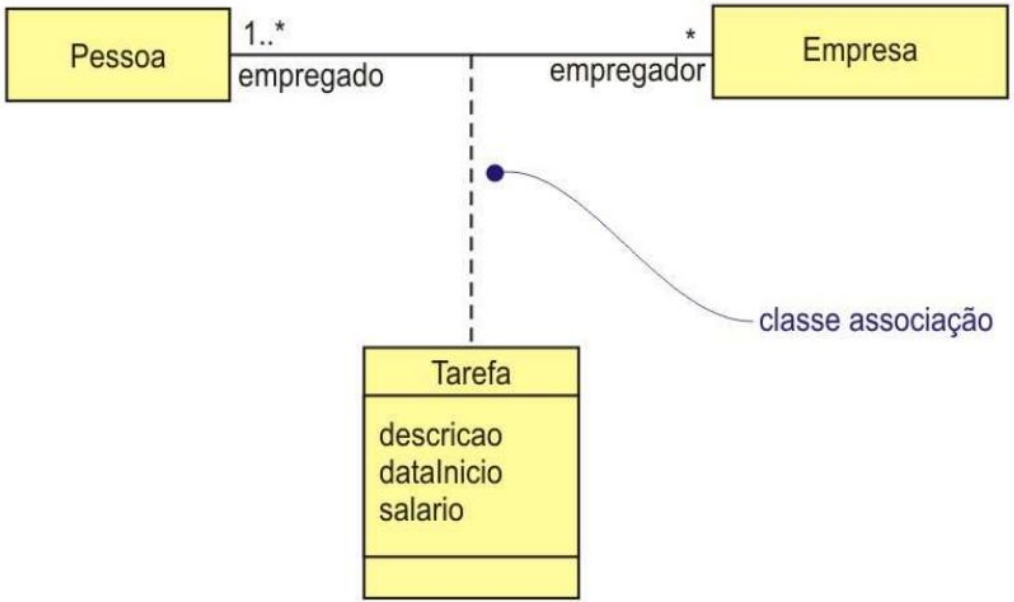
- A diferença entre eles é:

    - Quando, na agregação, a classe responsável pelo relacionamento é excluída, não deve excluir a classe que ele possui relacionamento.

    - Na composição, se a classe responsável pelo relacionamento for excluída, então deve-se excluir a classe que ele possui relacionamento.

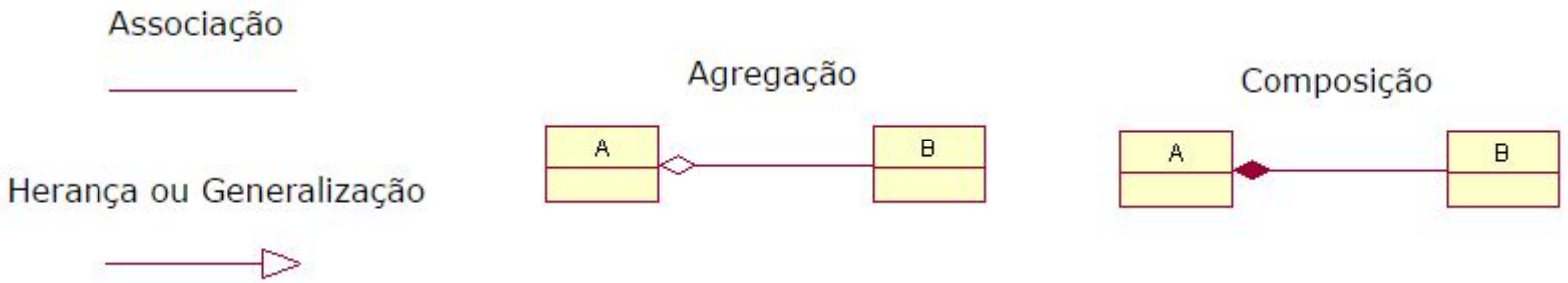
Classe de Associação

- Uma relação de associação entre classes pode ter seus próprios atributos, e quando isso ocorre, a associação deve ser modelada como uma classe.



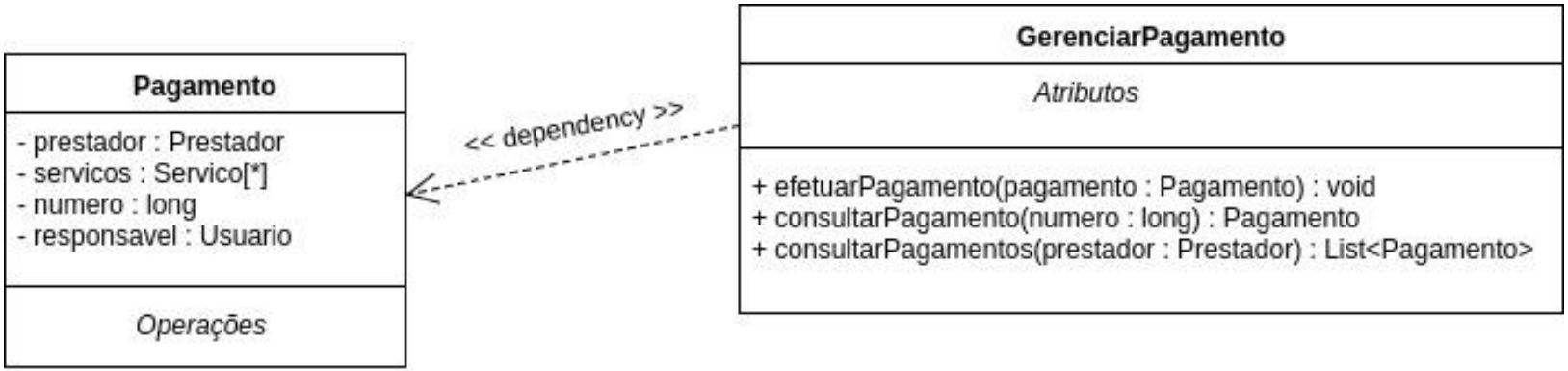
Relacionamentos - Representação Gráfica

- Para cada tipo de relacionamento existe uma representação gráfica.
- Estas são as representações dos relacionamentos mais comuns.



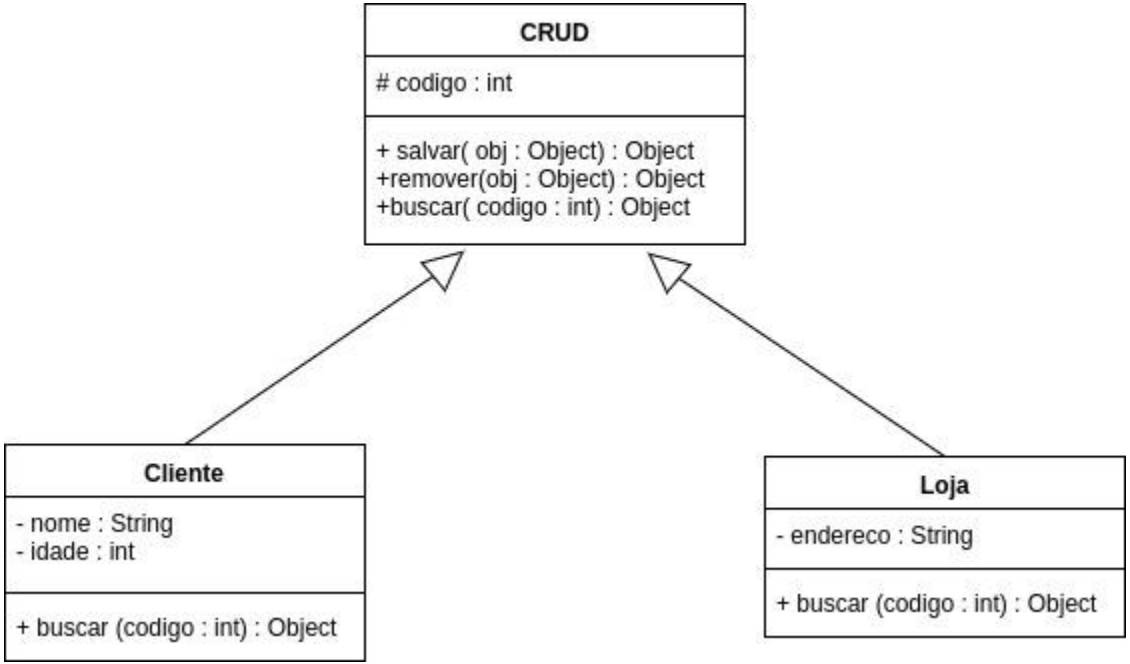
Dependência

- Utilizado quando em um relacionamento entre classes, uma depende da outra para realizar alguma operação.



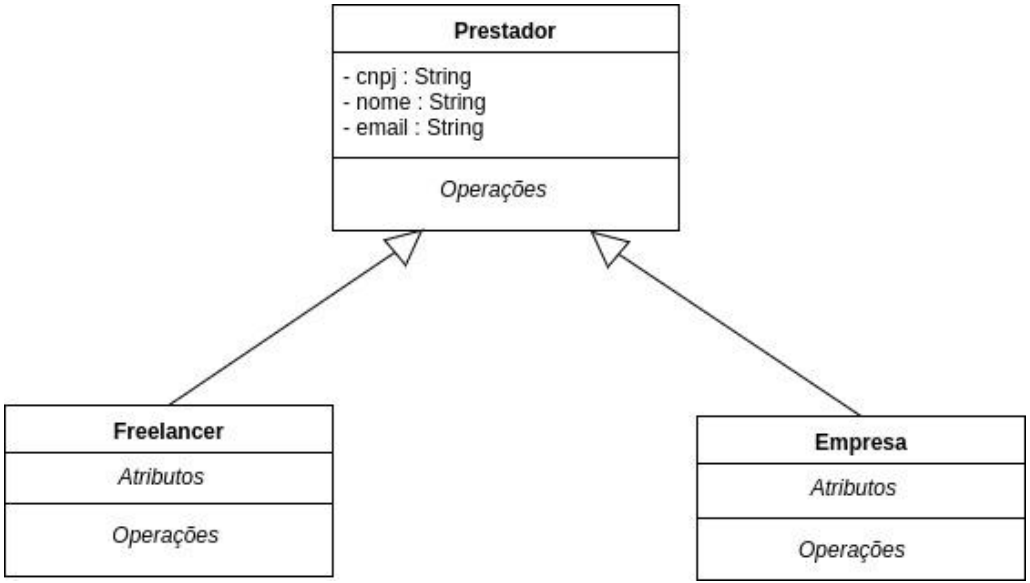
Classe Abstrata

- Utilizada para informar que uma classe não implementa todos os seus métodos.



Herança

- Utilizamos quando queremos declarar subclasses, permitindo a reutilização de códigos já declarados na superclasse.



Como explicar melhor possível meu projeto?

- O que colocar em um Diagrama de Classes

- O que não colocar em um Diagrama de Classes

O que colocar em um Diagrama de Classes

- O necessário para que as pessoas envolvidas consigam compreender o projeto.
- Manter notações simples.
- Gerar um diagrama flexível, facilitando futuras atualizações.
- Desenvolver diagrama com base na ideia adotada, podendo ser um diagrama específico por área de um sistema ou um diagrama com todas as classes envolvidas no sistema.

O que não colocar em um Diagrama de Classes

- Classes que aumentem a complexidade de um diagrama, sendo normalmente:
  - Classes que representam telas;
  - Classes de conexão e acesso ao banco de dados;
  - Classes de API's da linguagem ou de terceiros.

- Desenho de modelos para tudo, a menos que seja necessário.

Referências Bibliográficas

SAKURAI, Rafael; CASCARROLHO, Rodrigo. **UML - Criando Diagrama de Classes Eficientes**, Disponível em: <https://pt.slideshare.net/rodrigocasca/uml-criando-diagramas-eficientes>

WIKIPEDIA. **Diagrama de Classes,** Disponível em: <https://pt.wikipedia.org/wiki/Diagrama_de_classes>

SAUVÉ, Jacques Philippe. **Diagramas - Construindo um diagrama UML**, Disponível em: <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/diagramas/diagramas.htm>