TECNOLOGIA DA INFORMAÇÃO:

AGENDA 11

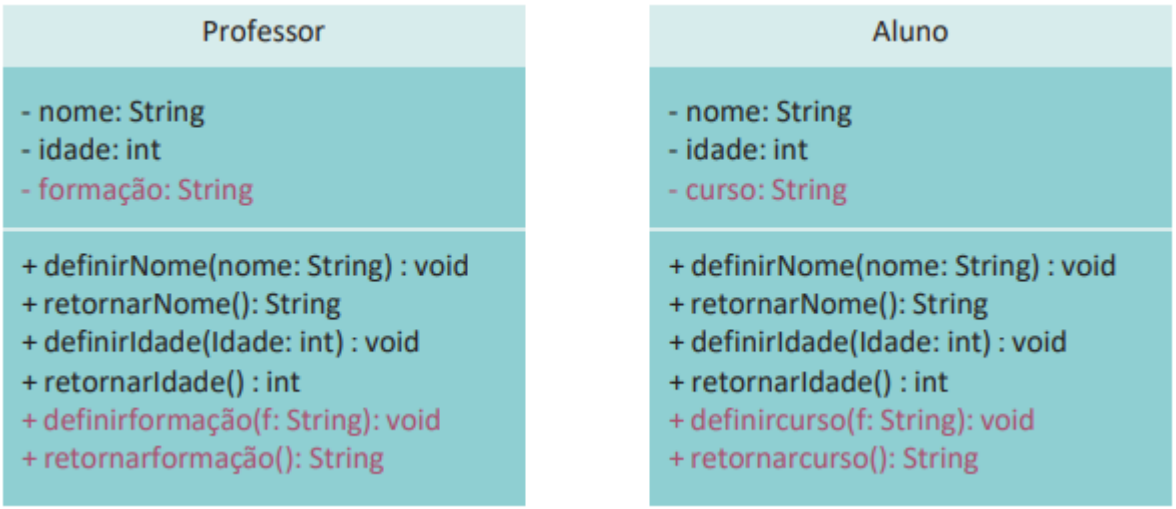
MODELAGEM ORIENTADA A OBJETO – Banco de dados orientado a partir de objetos do mundo real, ou seja, identificação de entidades existentes que devem constar no BD, depois disso são detalhadas suas características e ações. Parte-se das necessidades do usuário e da situação em que ele se encontra.

ABSTRAÇÃO: Identificar um item do mundo real e transformá-lo em uma classe. Classe **Alunos** (possuem características em comum).

CLASSE: É um conjunto de objetos distintos, porém com as mesmas características e comportamentos. Classe é uma abstração de entidades existentes no mundo real. Exemplos de classes: Um molde de biscoito de natal que poderá ser personalizado. A planta de uma casa com as especificações. Toda classe é composta por **atributos** (características) e **métodos** (ações).

OBJETO: Instância / elemento derivado de uma classe. É a representação do mundo real que será manipulado ou armazenado pelo sistema.Todo objeto é algo que existe, uma coisa concetra, já a classe é um modelo ou projeto de objeto. **Atributos** definem propriedades dos objetos. **Métodos** são ações ou procedimentos dos objetos. Boas práticas de métodos (nomes declarados como verbos) acaoVoltar, resgatarValor, pesquisaNomes, cadastrarSalarios, calcularMedias.

HERANÇA: Permite que as classes compartilhem atributos (por herança), métodos e outros membros das classes. Herança é um conceito muito importante em POO pois permite que classes herdem modelos / características em comum **evitando duplicidade de código.** Assim duas classes podem herdar os mesmos métodos e atributos e, ainda sim, reagirem de formas diferentes. Esses atributos / métodos comuns a mais de uma classe são **generalizações,** enquanto os que são pertinentes somente a uma determinada classe são conhecidos como **especificações.**



https://www.devmedia.com.br/introducao-a-programacao-orientada-a-objetos-em-java/26452 (devmedia - blog)

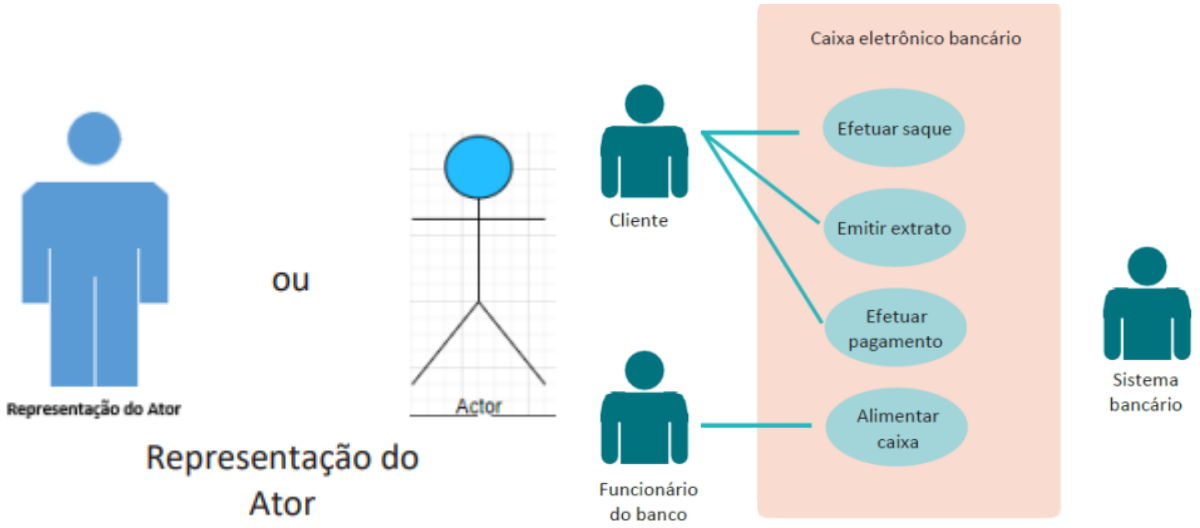
POLIMORFISMO: (Múltiplas formas). É quando a **mesma mensagem / requisição** recebida por **objetos diferentes** causam ações complementares **diferentes.** O comportamento muda de acordo com a classe que receber a mensagem / requisição. É possível chamar o mesmo método e fazê-lo reagir de formas diferentes para diferentes classes, para evitar a repetição de código.

UML (Unified Modeling Language): Linguagem Unificada de Modelagem. Diagramas de Casos de Uso e Diagrama de Classe da UML. Diagramas estruturais (classes, atributos, operações) – comportamentais (funcionamento, diagrama de casos de uso) – interação (interações entre objetos) – implementação. Modelagem seria desenhar um software, antes mesmo de começar a codificar (artifícios visuais – diagramas).

https://www.devmedia.com.br/modelagem-de-sistemas-atraves-de-uml-uma-visao-geral/27913 (devmedia blog UML)

https://www.youtube.com/watch?v=2dSq0Vu1GFo&ab_channel=DevMedia

DIAGRAMA DE CASO DE USO – Diagrama para levantar requisitos de um sistema, é constituído de: casos de uso, atores (que interagem, pessoa, dispositivo de hardware ou outro sistema) e seus relacionamentos.



< > INCLUDE: Quando um caso de uso A deve incluir um comportamento especificado para caso de uso B. Exemplo você vai acessar uma conta (A) que pede para um autenticador (B) um código de verificação

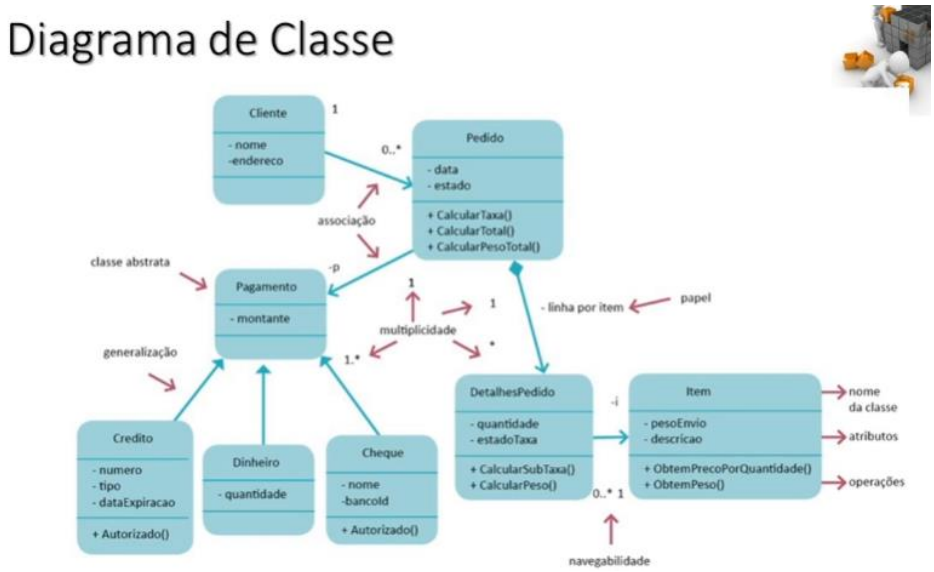
< > EXTENDS: Relação de extensão entre caso de uso A e um caso B.

https://www.youtube.com/watch?v=6kmDjHlK760&ab_channel=GEEaDCPS (POO GEEaD).

ESCOLHA DE MODELAGEM DE DADOS – A escolha do modelo ajuda a visualizar o sistema como ele deve ser na prática. Modelos orientados a objetos permitem construir sistemas mais próximos da realidade.

Exemplo: sistema de gerenciamento acadêmico (onde se mantém informações sobre alunos, cursos, professores e dados pertinentes à escola) cada uma dessas informações é armazenada na forma de objetos. Esses objetos de dados são extraídos do mundo real.

DIAGRAMA DE CLASSES



https://www.youtube.com/watch?v=A9Dy60t3MXs&ab_channel=GEEaDCPS **(**GEEaD)

https://www.youtube.com/watch?v=yhEqroz32Nk&ab_channel=UNIVESP (Univesp – POO)

ORIENTAÇÃO A OBJETOS – Modelo poderoso e conveniente para necessidades de modelagem da vida real. **Organizar o mundo real como uma coleção de objetos que incorporam estrutura de dados e um conjunto de operações que manipulam estes dados.**

Exemplo: Sistema de concessionário de veículos, **objetos:** clientes, veículos, seguros, documentações, vendas, despesas, faturamento, comissões. Cada um desses itens é armazenado no BD na forma de **objetos** (uma representação do mundo real que está registrada no sistema e deve possuir informações importantes).

- **Objeto cliente:** nome, telefone, endereço; No contexto de compra / venda: pagamento, vistoria, documentação; Essas ações são chamadas de **métodos.**

PROGRAMAÇÃO ORIENTADA A OBJETOS (POO) – Casa (objeto) > Planta com especificações (Classe) >

Objeto == representação real de uma classe, ou seja, a casa em si;

Classe == é a abstração de um objeto real que aponta as características do objeto, ou seja, a planta;

Nome da classe

Atributos

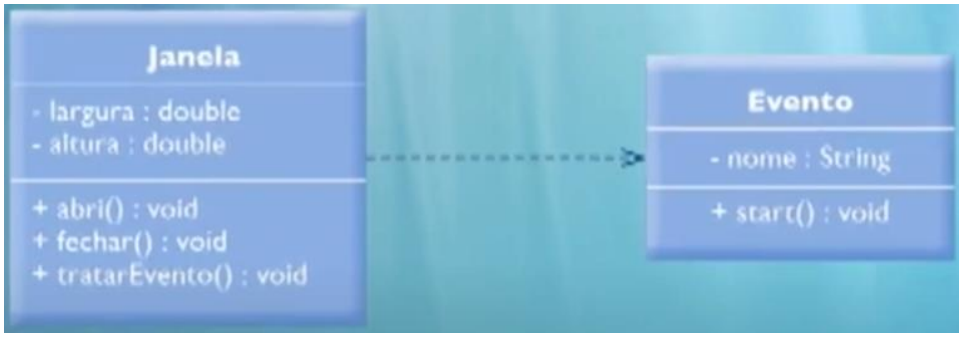
Métodos

Responsabilidades

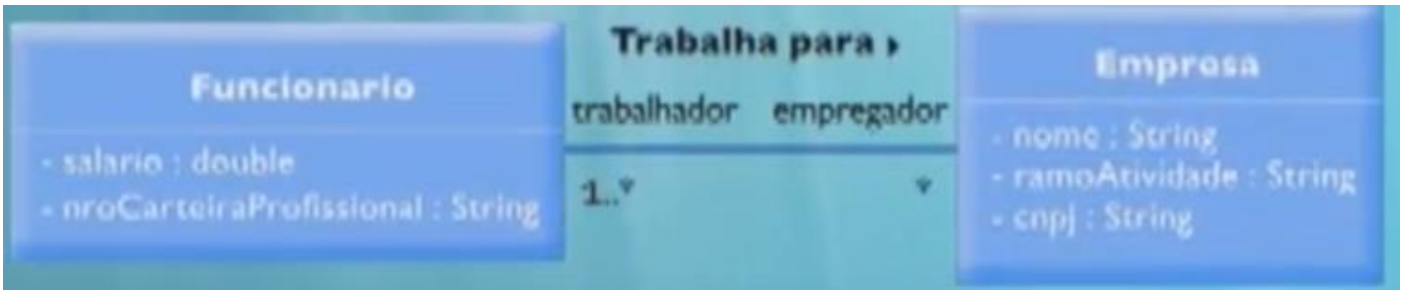
A partir do levantamento das características das classes é que se constroem os objetos.

RELACIONAMENTO ENTRE CLASSES

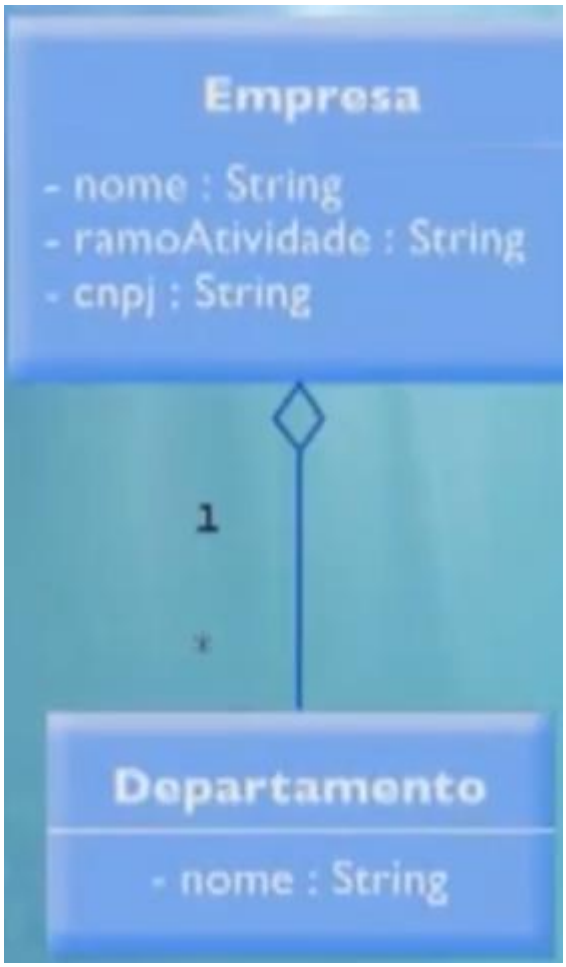
1. **DEPENDÊNCIA:** Relacionamento de utilização. Indicada por linha tracejada com seta que indica o sentido da dependência.



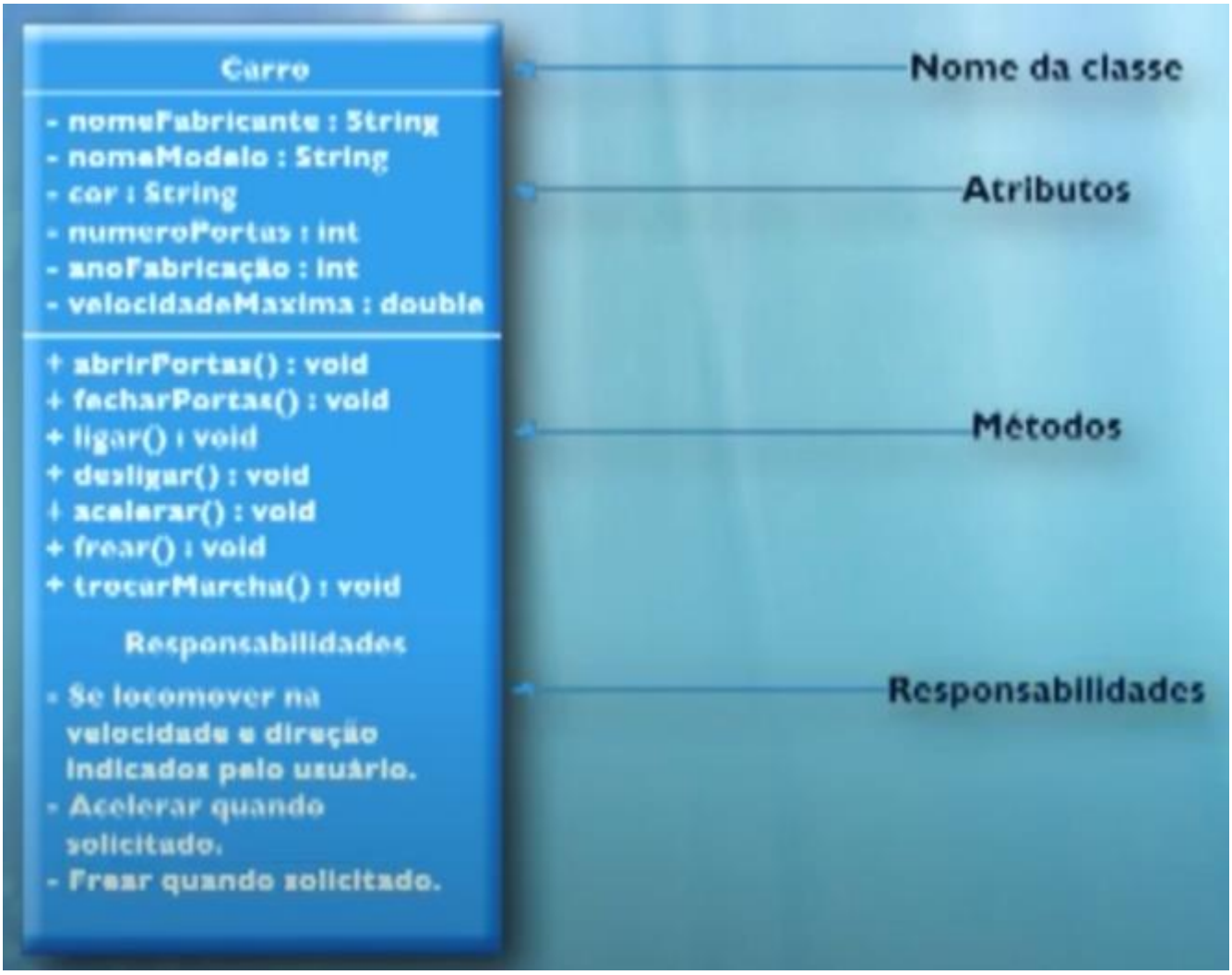
2. **ASSOCIAÇÃO:** Relacionamento estrutural. Conecta objetos de uma classe à objetos de outra classe. É representada por uma linha interligando duas classes.



3. **HERANÇA:** Usa rotinas de dados já existentes para modelar relacionamentos semelhantes.



CARACTERÍSTICAS DAS CLASSES – ENCAPSULAMENTO: Implementação sem explicação. **POLIMORFISMO** (Muitas formas) enviar mesma mensagem a diferentes a vários tipos objetos e cada um deles se comporta de forma diferente para atender a solicitação.



https://www.youtube.com/watch?v=JS6PQy4PMT8&ab_channel=GEEaDCPS (CInformática M2 Agenda07 - GEEaD)