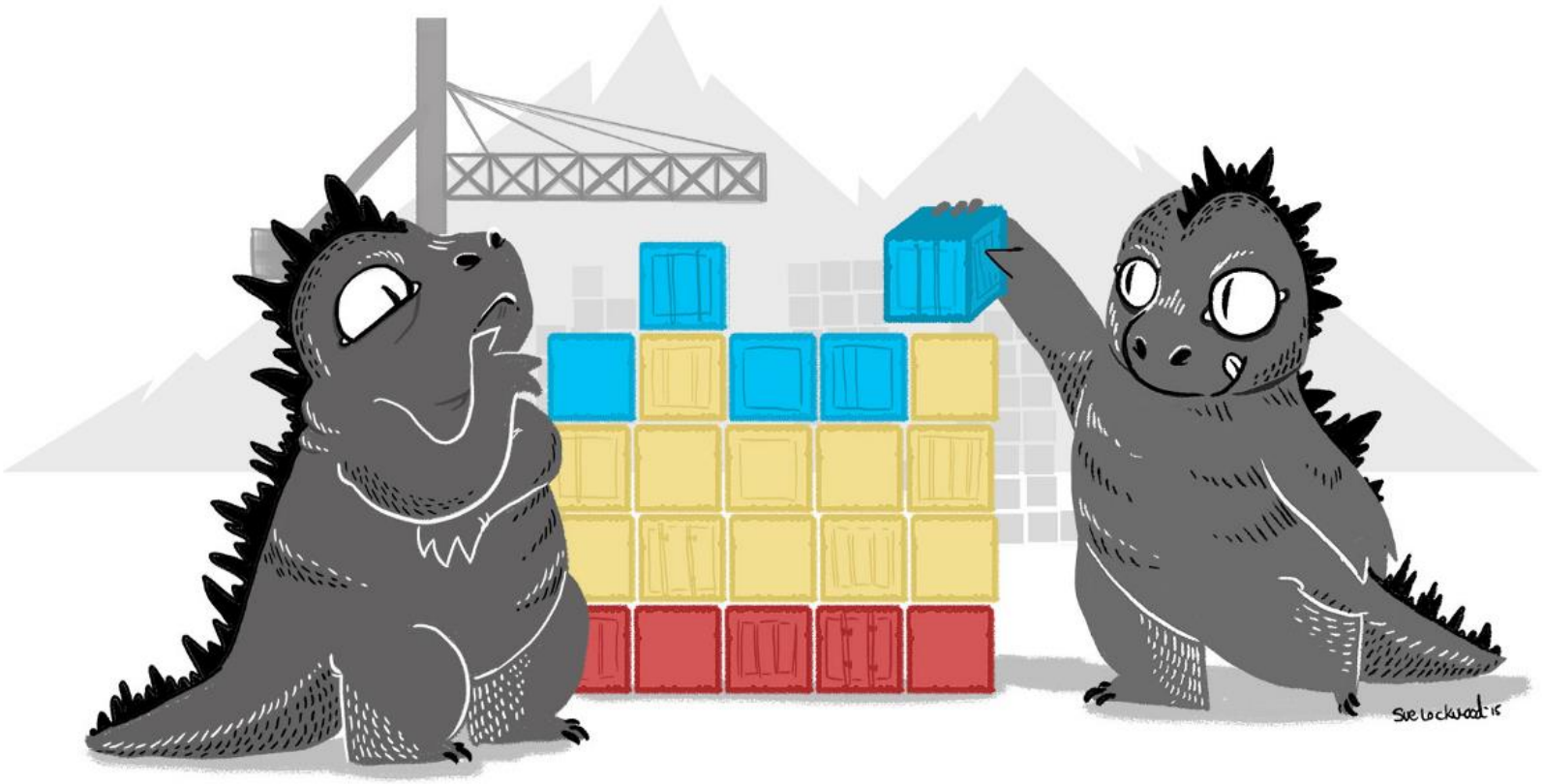


Curso: Análise e Desenvolvimento de Sistemas

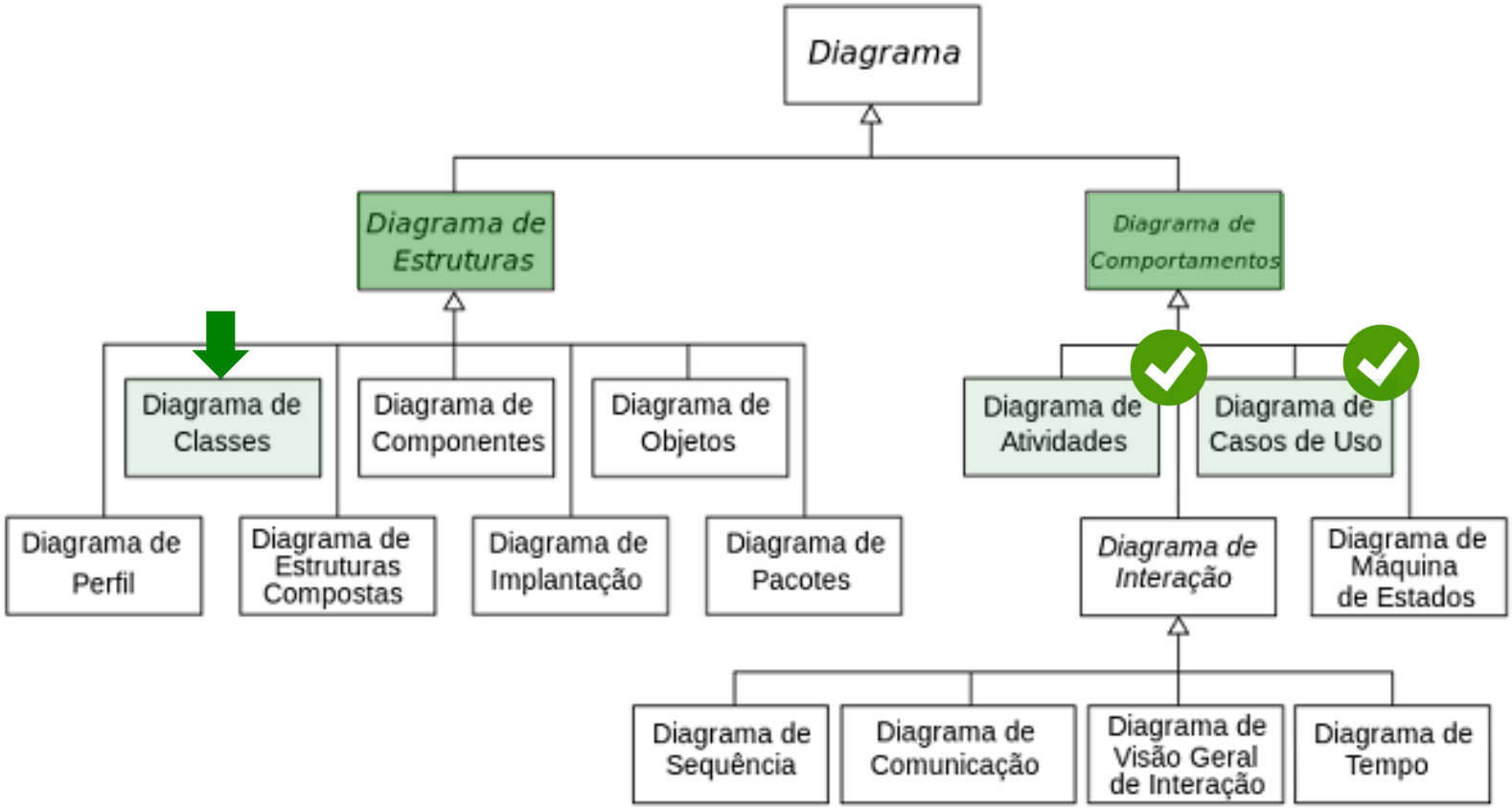
Disciplina: Análise e Projeto Orientado a Objetos

Profª Drª Marcia Cassitas Hino

1º Semestre/2024

Universidade Tuiuti do Paraná

ONDE ESTAMOS



Categoria	Diagramas	Descrição
Diagramas Dinâmicos	Casos de uso	Expressam a funcionalidade de um sistema
	Atividades	Representam o fluxo de atividades dos processos de negócio
	Interatividade	Apresenta um fluxo de atividades, mostrando como elas trabalham em uma sequência de eventos
	Sequência	Define a ordem e a troca das mensagens entre objetos
	Comunicação	Representa o diagrama anterior de colaboração
	Máquina de estados	Representa as ações ocorridas em resposta ao recebimento de eventos
	Temporal	Mostra mudança de estado de um objeto
Diagramas Estruturais	Classes	Apresenta elementos conectados por relacionamentos
	Objetos	Apresenta objetos e valores de dados
	Componentes	Mostra dependências entre componentes de software
	Pacotes	Usado para organizar elementos de modelos e mostrar dependências entre eles
	Implantação	Mostra a arquitetura do sistema em tempo de execução, as plataformas de software, etc.
	Estrutura composta	Usado para mostrar a composição de uma estrutura complexa

CLASSE

- É uma estrutura de dados para declarar variáveis. Representa as características de um objeto e serve para defini-lo.
- Uma instância de uma classe é chamada de objeto, pois atribui valor às variáveis
- Definir uma classe não cria um objeto, assim como um tipo de variável não é uma variável.

- Representa um conjunto de objetos, com suas características próprias.

- O comportamento de **objetos** é definido em uma classe por meio de **métodos** e estados que ele pode possuir. Tudo isso é suportado por **atributos**.

- Uma **classe** refere-se a descrição de um conjunto de objetos que compartilham os mesmos atributos, operações e relacionamentos.

OBJETO

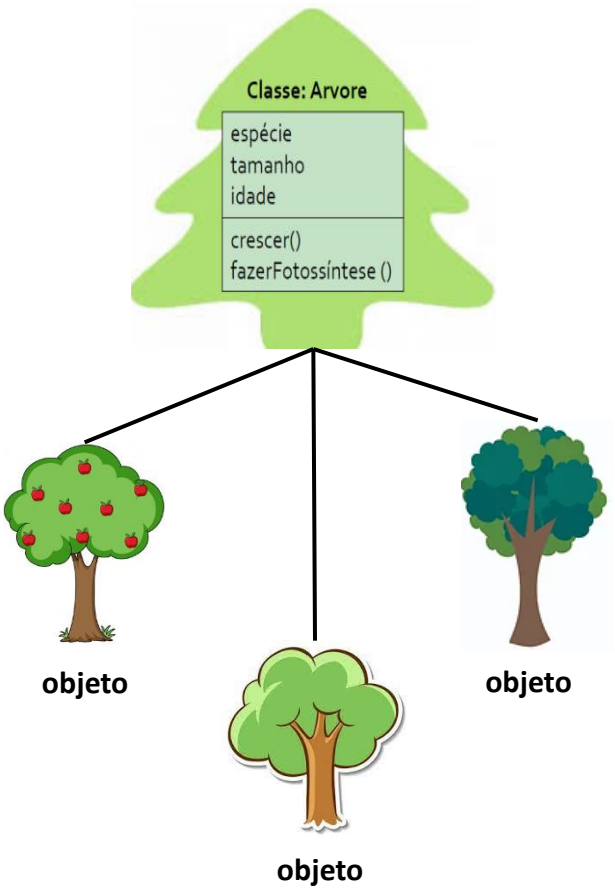
Algo concreto que pode ser percebido pelos sentidos e representado por dados e métodos.

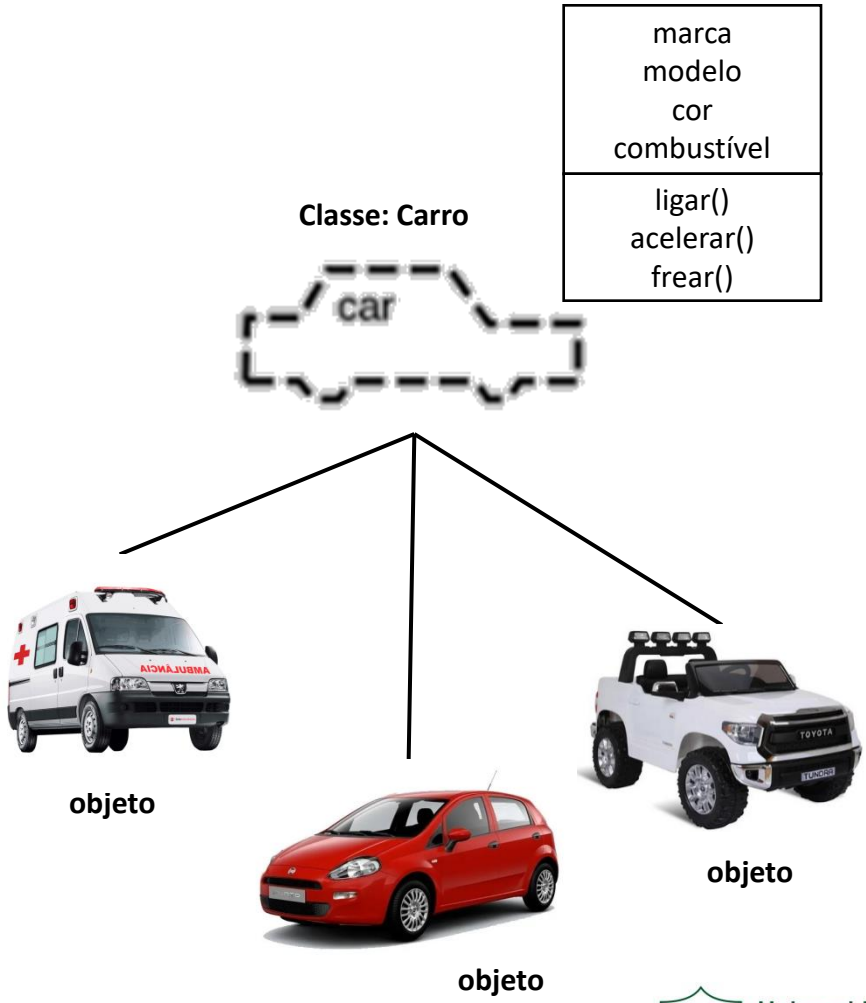
CLASSE

É uma estrutura de informações que descrevem um objeto.

nome data nascimento sexo profissão
Andar() Falar() Ouvir() Comer() Dormir() Vestir()

EXEMPLOS





Universidade Tuiuti do Paraná

Importante entender que Classes não são Objetos, mas a estrutura de um Objeto.

Quais seriam os atributos da uma classe CARRO?

E os atributos da classe HUMANOS?

Quais são exemplos de objetos da classe CARRO?

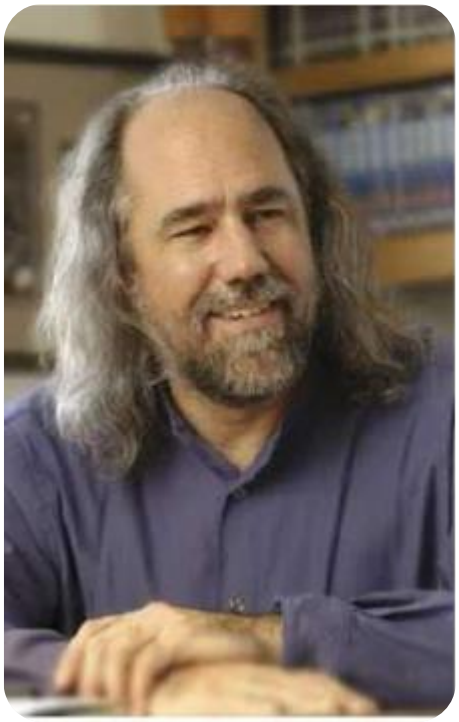
E objetos da classe HUMANOS?

Diagrama de classe

- O mais importante e o mais utilizado diagrama da UML.
- Permite a visualização das classes que compõem o sistema e seus relacionamentos.
- Em um diagrama de classes são definidas as classes de objetos, suas operações, atributos e as relações entre classes.

- A dificuldade é que não há roteiro ou receita para escolher as classes de um sistema. Isso é uma tarefa que depende em grande parte da experiência do desenvolvedor.

- Nas fases iniciais do projeto, as classes são chamadas de **classes candidatas** ou de análise, pois há grande possibilidade que mudem ao longo do projeto.



Grady Booch

James Rumbaugh

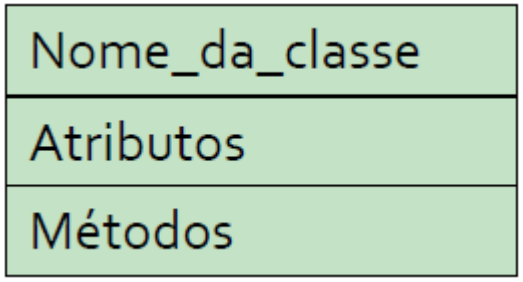
Ivar Jacobson

Os diagramas de classes são os diagramas encontrados com maior frequência na modelagem de sistemas orientados a objetos. Eles são a base para outros diagramas.

Universidade Tuiuti do Paraná

- Na UML todos esses itens são modelados como classes → abstração de itens que fazem parte de seu vocabulário.

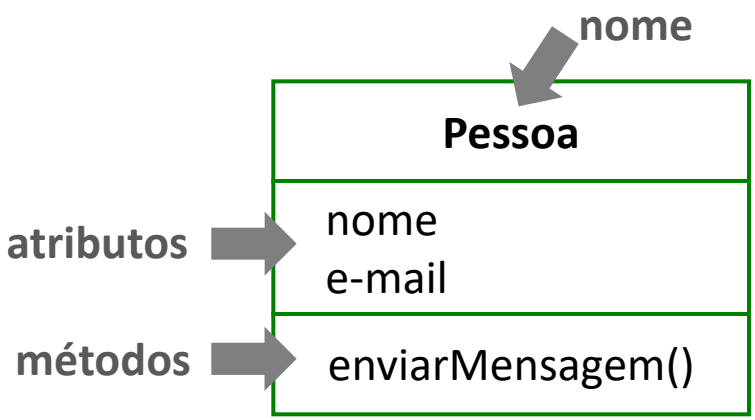
- A UML permite a representação gráfica de classes, tal notação permite visualizar uma abstração independente de qualquer linguagem de programação, dando ênfase às partes mais importantes de uma abstração: seu **nome, atributos e métodos.**



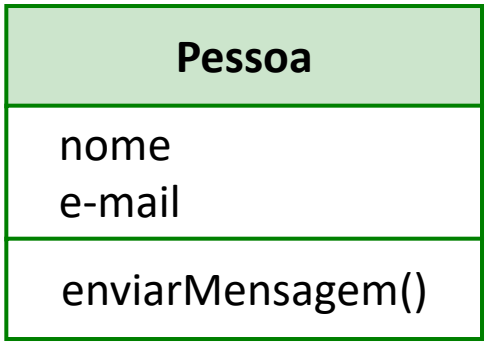
Estrutura de uma classe

Uma classe é representada por um retângulo com três divisões:

- Nome da classe
- Atributos da classe (dados ou características)
- Métodos da classe (operações)

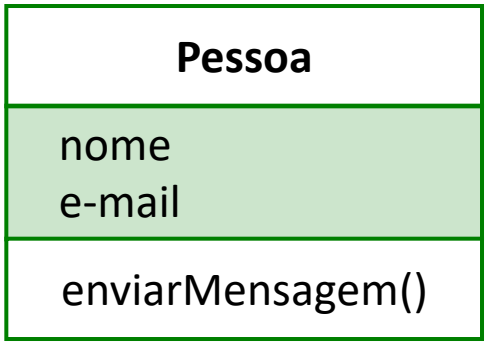


NOME DA CLASSE



- Cada classe deve ter um nome que a **diferencie** de outras classes.
- O nome de uma classe pode ser um texto composto por qualquer número de caracteres.
- Classes devem ter nomes com **substantivos.**
  - Exemplo: Cliente, PaginaWiki, ContaCliente, Endereco...
- **Evite palavras genéricas** como Processador, Dados, Info,....

ATRIBUTOS DA CLASSE



- Um atributo é uma propriedade nomeada de um classe, que descreve um intervalo de valores aos quais as instâncias da propriedade podem apresentar.

- Cada atributo tem um **nome** e um **tipo.**

- O **nome** do atributo pode ser um texto, como o nome das classes deve ser um substantivo.
  - Exemplos: nome, endereço, telefone, dataNascimento, ...

Visibilidade do atributo

- Pública (+)
  Qualquer classe do sistema pode manipular (ler e alterar) o atributo.

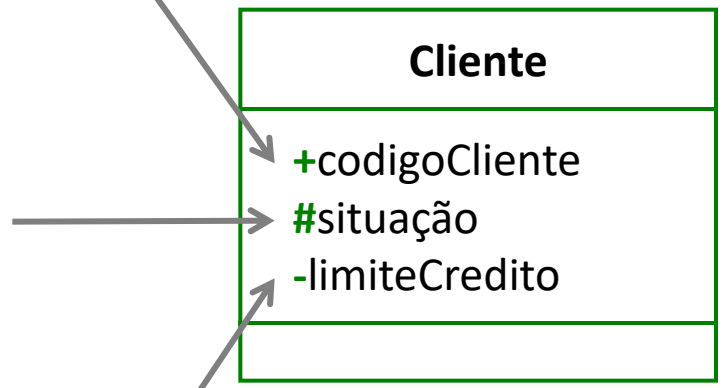
- Protegida (#)
  Qualquer descendente/subclasse pode manipular (ler e alterar) o atributo.

- Privada (-)
  Somente a própria classe pode manipular (ler ou alterar) o atributo.

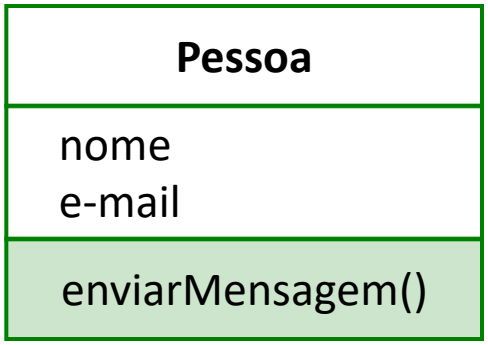
Visibilidade pública
Qualquer classe do sistema pode solicitar seu conteúdo e alterá-lo.

Visibilidade protegida
Somente as subclasses do cliente podem ler e alterar seu conteúdo.

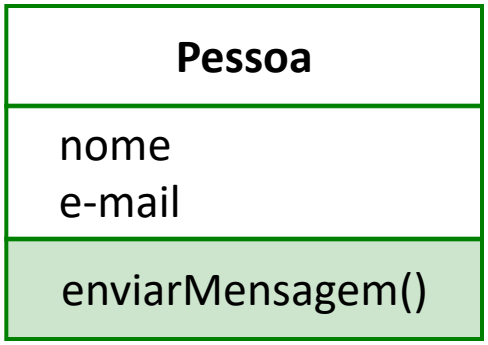
Visibilidade privada
Somente a classe cliente tem acesso a essa informação e pode alterá-la.



MÉTODOS DA CLASSE



- São as funções de uma classe, também chamados de operações.
- Um método ou uma operação é a definição de um serviço que pode ser solicitado por algum objeto da classe para modificar o comportamento.
- Uma classe pode ter **diversos métodos** ou até **nenhum método.**



- O nome de um método pode ser um texto, na prática ele é representado por um **verbo.**
- Se identifica apenas parâmetros e o retorno, ou seja, se especifica a assinatura do método, que contém o nome, o tipo de todos dos parâmetros e o tipo de retorno.
  - Exemplo: RelizarLogon(userID: string, password:string): boolean
- O métodos são declarados **apenas** neste diagrama.

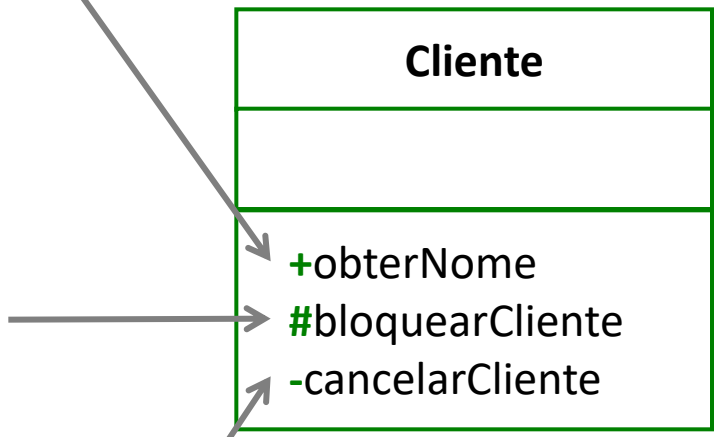
Visibilidade do método

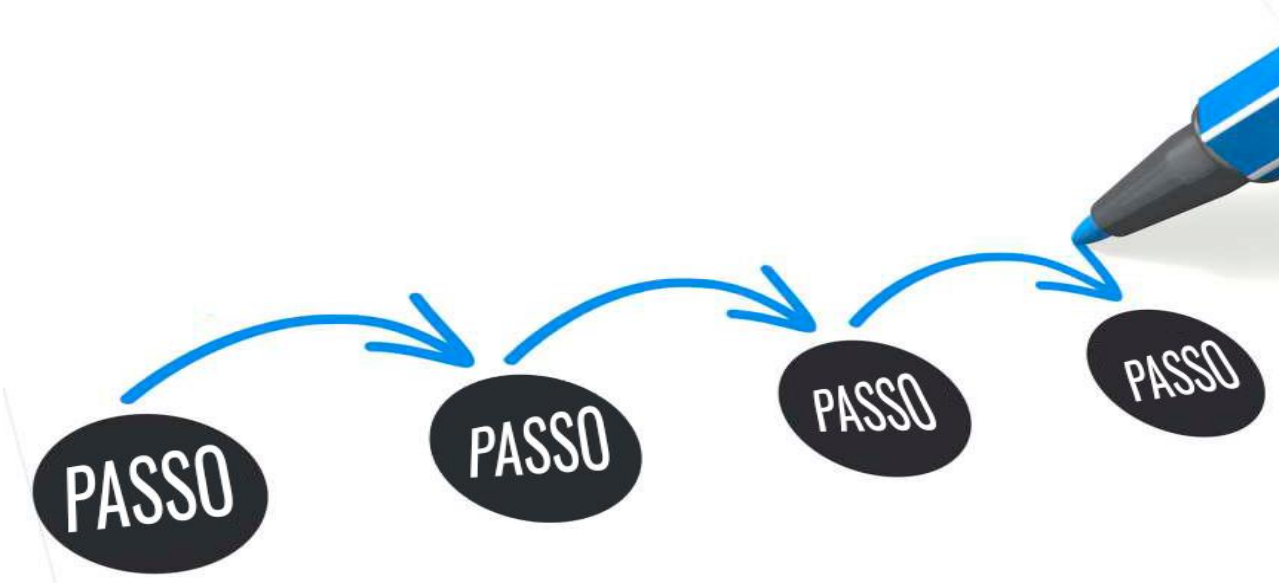
- Público (+)
  Método visível por todas as classes do sistema.
- Protegido (#)
  Método que precisa ser estendido por outras classes.
- Privado (-)
  Funcionalidade que apoia somente a própria classe.

Visibilidade pública
Pode ser chamado por qualquer classe do sistema.

Visibilidade protegida
Somente as subclasses do cliente podem fazer uso dessa funcionalidade.

Visibilidade privada
Somente a classe cliente tem acesso a essa função do cliente.





Conta Bancária

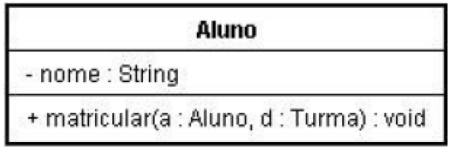
Conta Bancária
identificação saldo dataAbertura

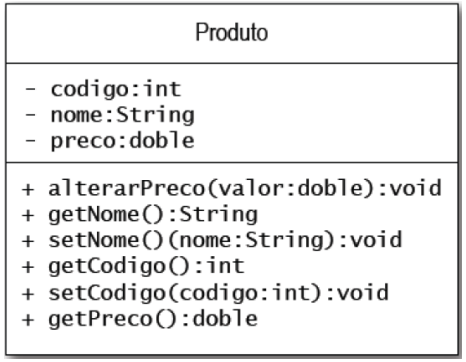
Conta Bancária
criar() bloquear() desbloquear() creditar() debitar()

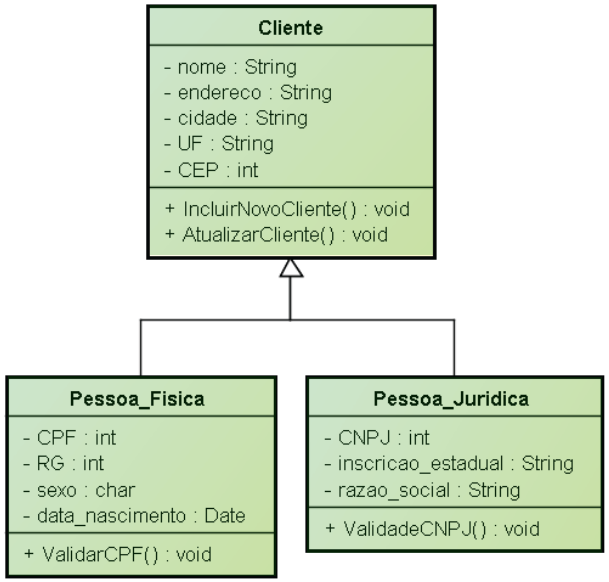
Conta Bancária
identificação saldo dataAbertura
criar() bloquear() desbloquear() creditar() debitar()

Conta Bancária
identificação: String saldo: Quantia dataAbertura: Date
+criar() +bloquear() +desbloquear() +creditar(valor: Quantia) +debitar(valor: Quantia)

Exemplo de classes

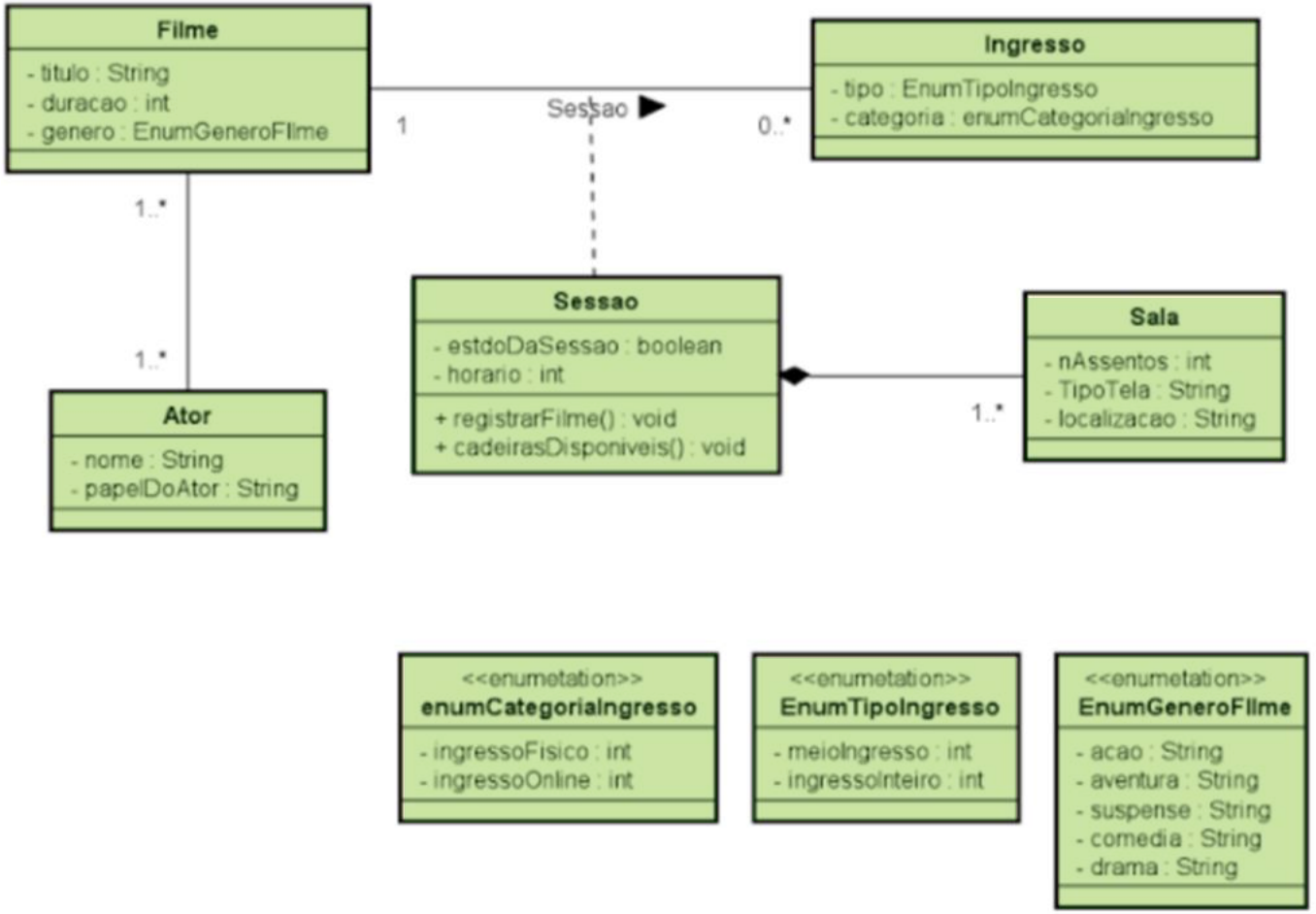


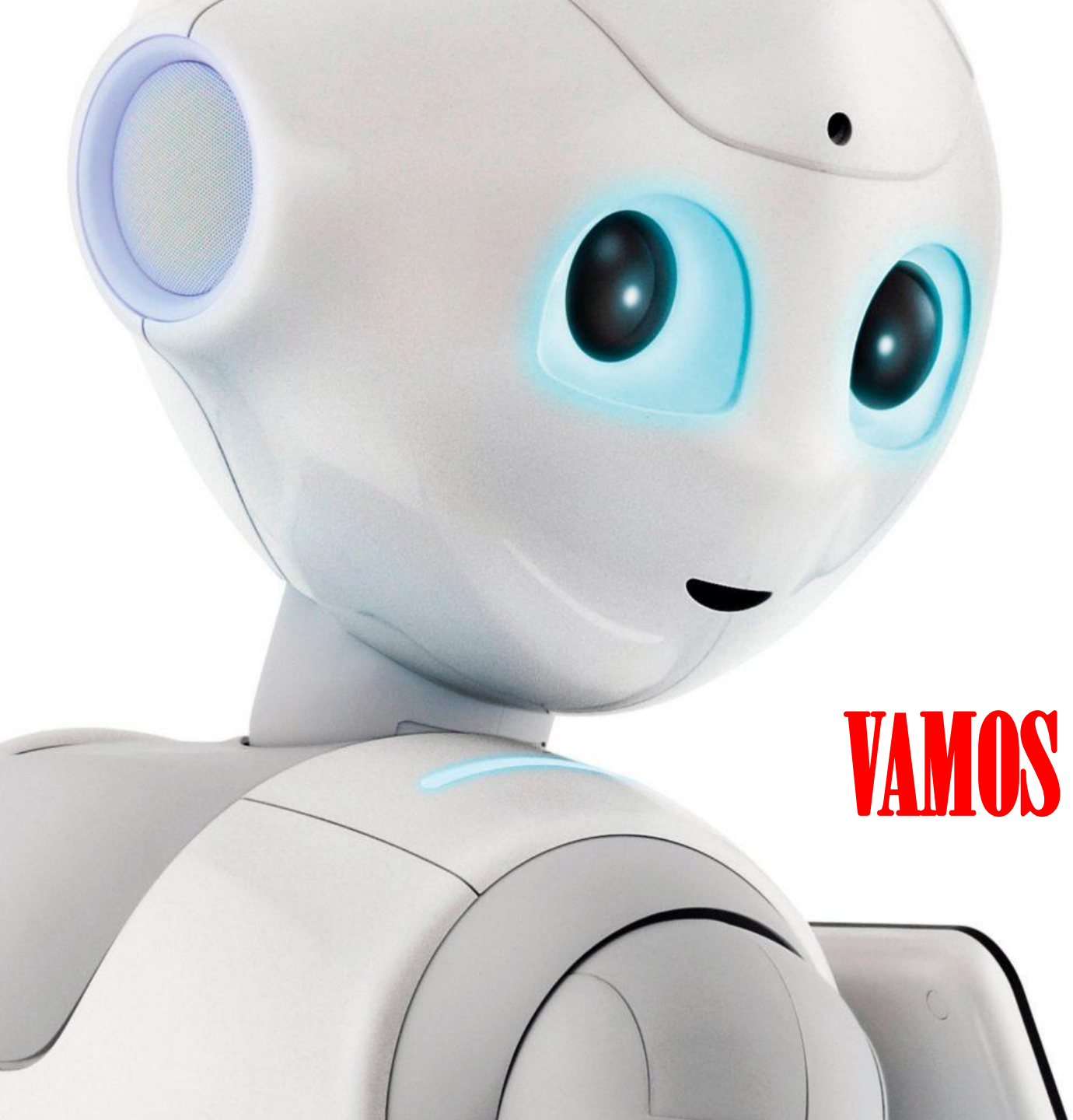




Universidade Tuiuti do Paraná

Exemplo de diagrama de classes





VAMOS PRATICAR?

Universidade Tuiuti do Paraná

EXERCÍCIO 1:
DIAGRAMA BÁSICO DE CLASSES

Construa as classes necessárias para atender as situações abaixo:

Ana faz o controle de seus gastos diários em uma planilha eletrônica com as informações:

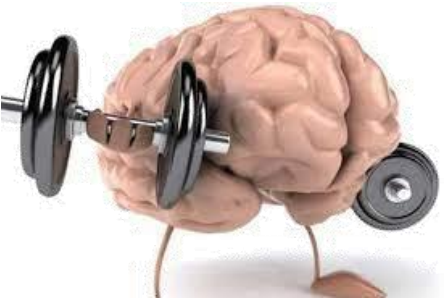
- Para cada gasto ela cadastra: o tipo do gasto (remédio, roupa, refeição, ...), a data do gasto, o valor gasto e a forma de pagamento (dinheiro, cartão de débito ou cartão de crédito)
- No final do mês, ela lista o total dos gastos mensais agrupados por tipo de gasto e, exibe o quanto foi gasto por tipo de forma de pagamento

RESPOSTA EXERCÍCIO 1

TipoGasto
+ descriçãoTipo: string
+ cadastrarTipo (descriçãoTipo: string): boolean + listarTipos(): List

formaPgto
+ descriçãoForma: string
+ cadastrarForma (descriçãoForma: string): boolean + listarForma(): List

Gasto
- tipoGasto: tipoGasto - dataGasto: date - valorGasto: float - formaPgto: formaPagamento
+ cadastrarGasto (tipoGasto: tipoGasto, dataGasto: date, valorGasto: float, formaPgto: formaPgto): boolean + gerarRelatorioMensal (mes: int, ano: int): List



Universidade Tuiuti do Paraná

RELACIONAMENTOS DE CLASSES

- Os objetos tem relações entre eles:
  - um professor ministra uma disciplina para alunos numa sala
  - um cliente faz uma reserva de alguns lugares para uma data
- Essas relações são representadas também no diagrama de classe
- A UML reconhece três tipos como os mais importantes: associação, generalização (ou herança) e dependência

Universidade Tuiuti do Paraná

Componentes de um relacionamento:

- **Representação:** tipo da linha

- **Nome**
  escrito junto à linha que representa a associação, normalmente um verbo

- **Multiplicidades**
  cada associação possui 2 multiplicidades, uma em cada extremo da linha de associação.

- **Navegabilidade ou direção de leitura:**
  Indica como a associação deve ser lida

Indicadores de Multiplicidade

- É a definição de um intervalo de inteiros (limite mais baixo e limite mais alto) para especificar a cardinalidade na criação de um objeto.

Simbologia	Descrição
1	Exatamente um
1..*	Um ou mais
0..*	Zero ou mais
*	Muitos (vários objetos estão envolvidos)
0..1	Zero ou um
3...5	No mínimo três e no máximo cinco.

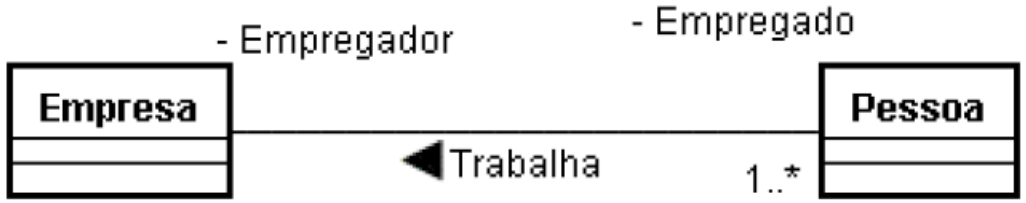
A UML possui cinco tipos de relacionamento, mas reconhece três como os mais importantes:

- associação
- generalização (ou herança) e
- dependência

RELACIONAMENTO: ASSOCIAÇÃO

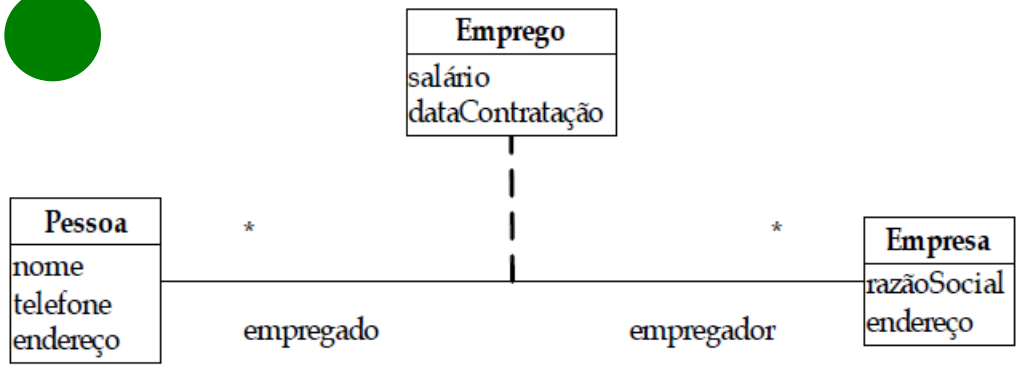
- Esse tipo de relacionamento é utilizado para representar o fato de que objetos podem se relacionar uns com os outros
- Associação representa que duas classes possuem uma ligação como, por exemplo, que elas "conhecem uma a outra”
- Representado por uma linha contínua.

Exemplos de utilização do relacionamento tipo associação:



CLASSE ASSOCIATIVA

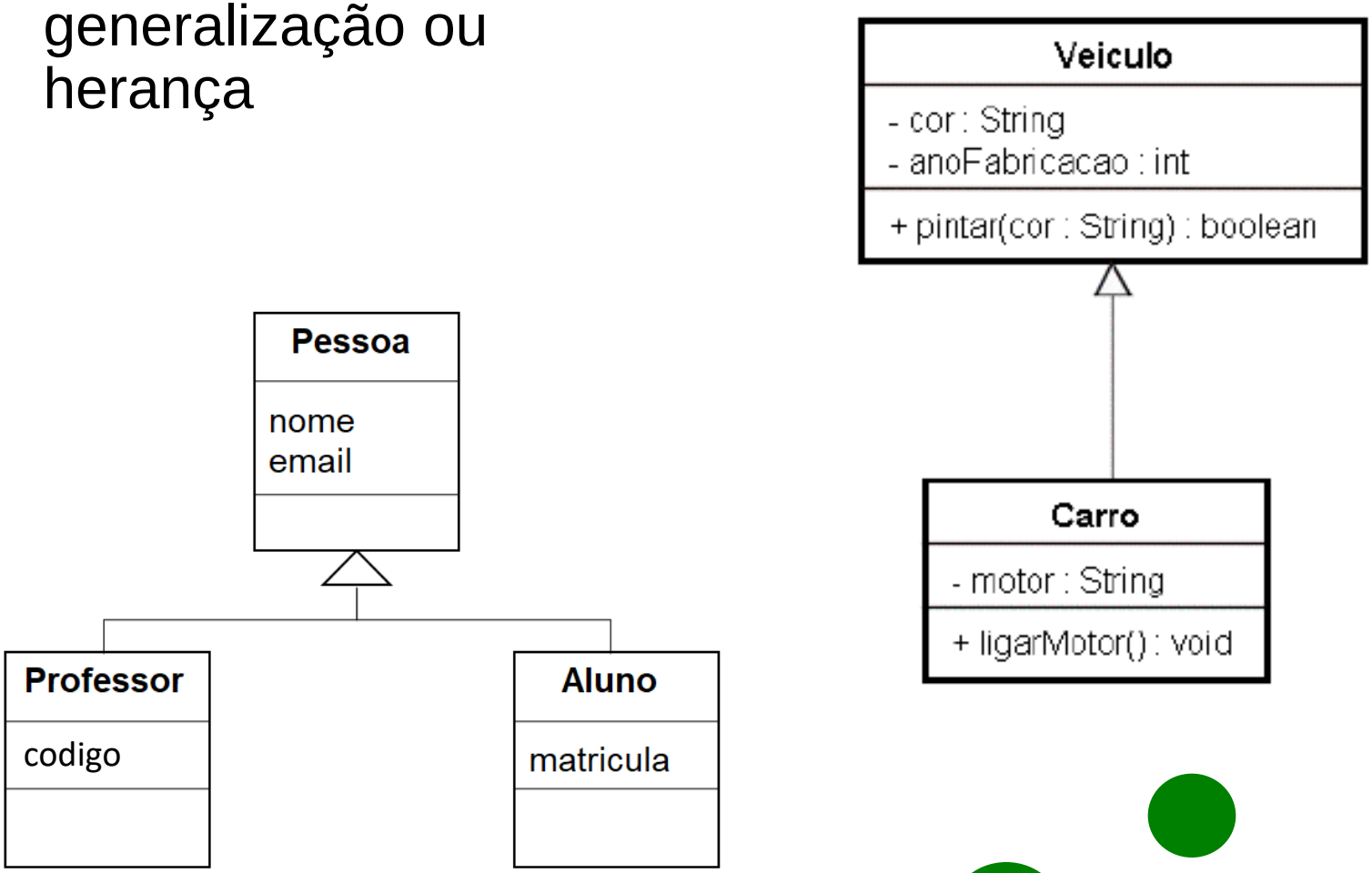
- Classe que está ligada a uma associação, ao invés de estar ligada a outras classes
- Necessária quando duas ou mais classes estão associadas, e é preciso registrar informações sobre esta associação



RELACIONAMENTO: GENERALIZAÇÃO / HERANÇA

- É um tipo de relacionamento similar à associação de mesmo nome em um diagrama de casos de uso
- Seu objetivo é identificar classes-mãe, chamadas gerais e classes-filhas, chamadas especializadas
- Significa ser capaz de incorporar os atributos e métodos de uma outra classe previamente definida

Exemplo de utilização do relacionamento tipo generalização ou herança



RELACIONAMENTO: DEPENDÊNCIA

- São relacionamentos de utilização, determinando as modificações na especificação de um item, ou seja, uma mudança na especificação de um elemento pode alterar a especificação do elemento dependente
- A dependência entre classes indica que os **objetos de uma classe usam serviços dos objetos de outra classe**
- Representada por uma linha tracejada apontando o item do qual o outro depende

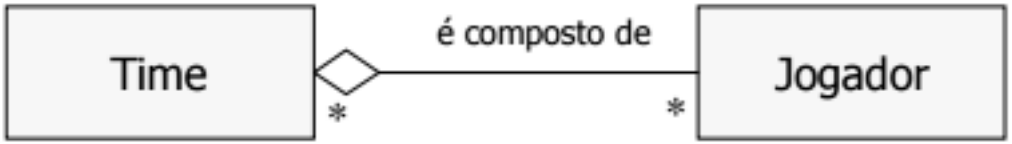
Exemplo de utilização do relacionamento tipo dependência

No exemplo só pode existir um item do pedido se existir o pedido

RELACIONAMENTO: AGREGAÇÃO

- É um caso especial de associação utilizada para representar conexões que guardam uma relação **todo-parte** entre si
- Em uma agregação, um objeto está contido no outro, ao contrário de uma associação
- Onde se puder usar uma agregação, uma associação também poderá ser utilizada
- A destruição de um objeto todo não implica necessariamente a destruição de suas partes
- Representada por uma linha contínua, com um losango na classe que representa o todo

Exemplos de utilização do relacionamento tipo agregação



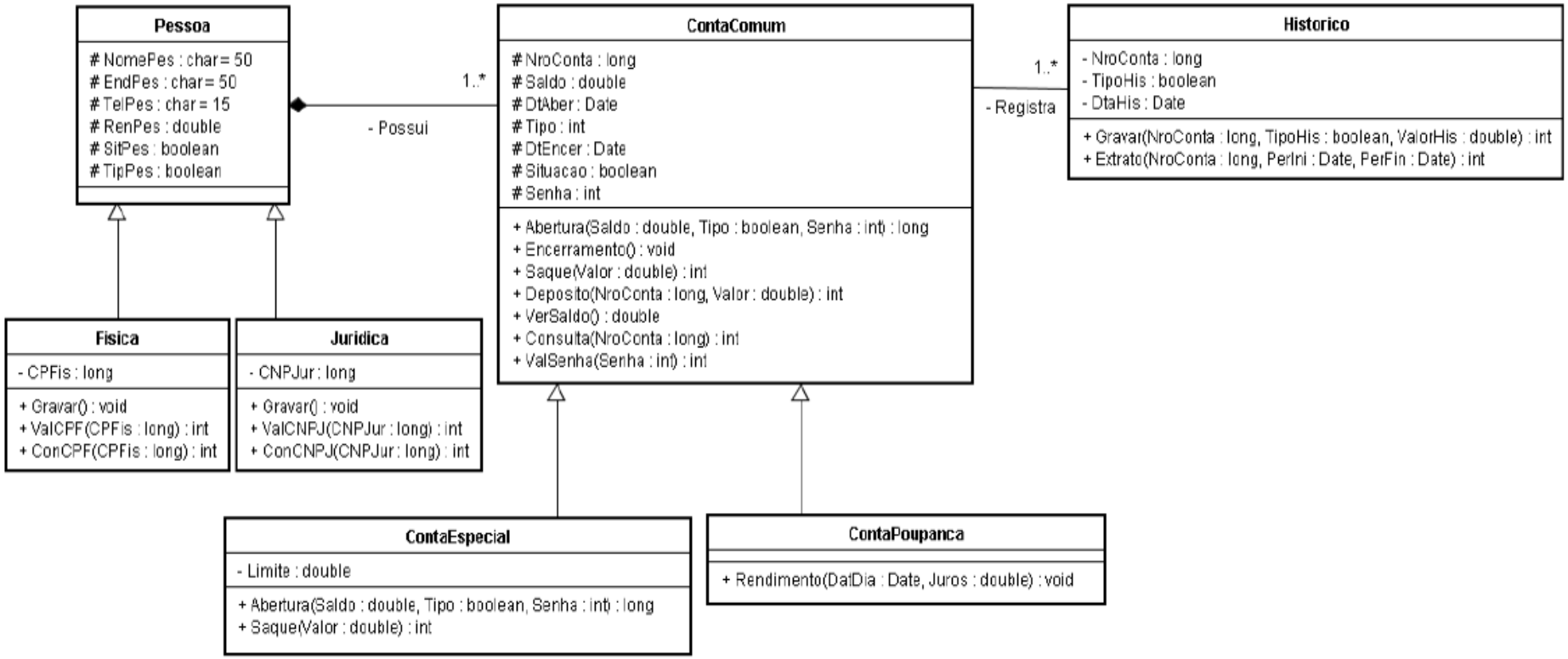
RELACIONAMENTO: COMPOSIÇÃO

- Uma variação do tipo agregação.
- Representa um vínculo mais forte entre objetos-todo e objetos-parte.
- Objetos-parte **têm** que pertencer há um único objeto-todo.
- Representado por uma linha contínua com um losango preenchido na classe que representa o todo.

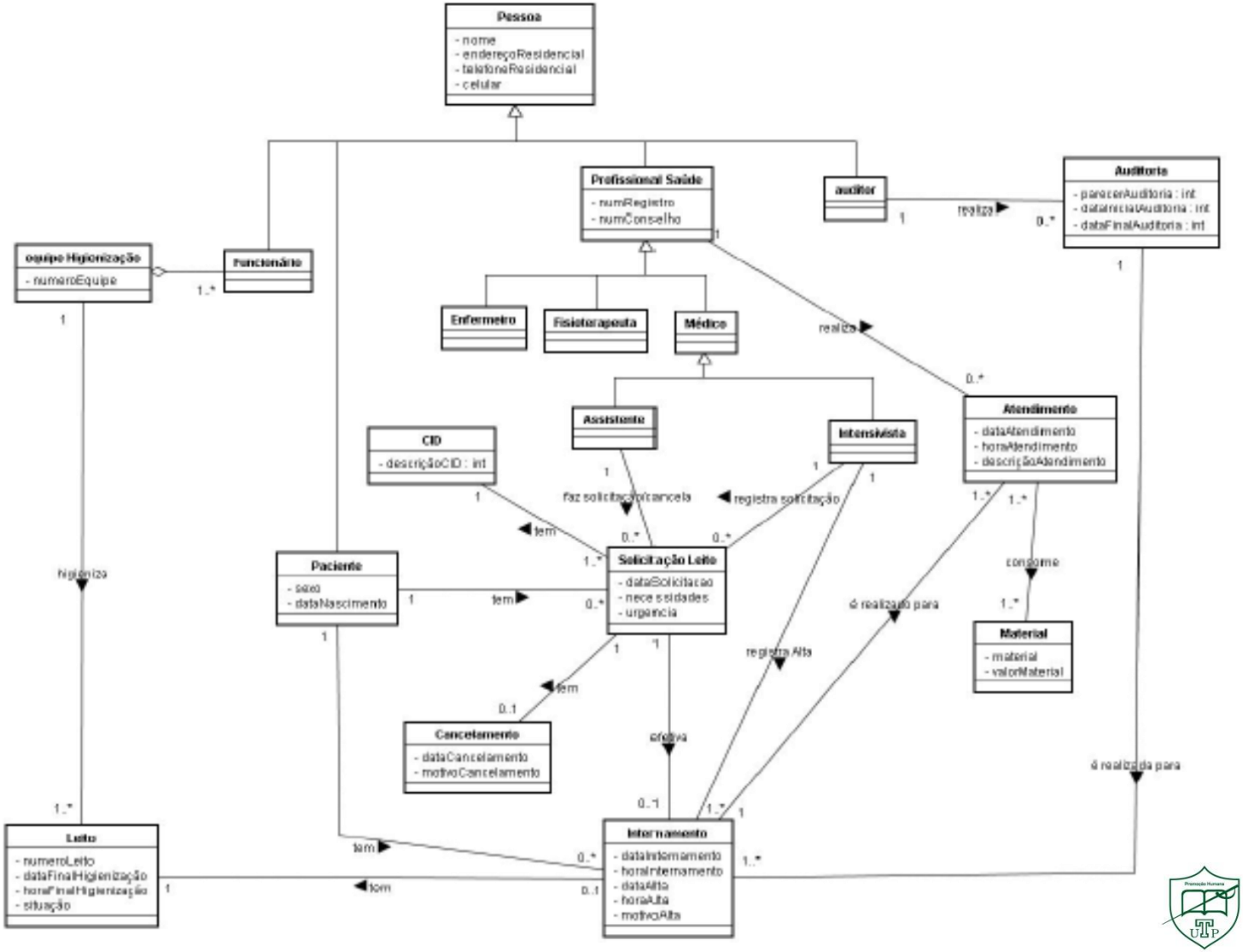
Exemplos de utilização do relacionamento tipo composição

O relacionamento é mais forte e exclusivo, significando que se o objeto-todo (livro) for excluído/eliminado, o objeto-parte (capítulo) também serão excluídos/eliminados.

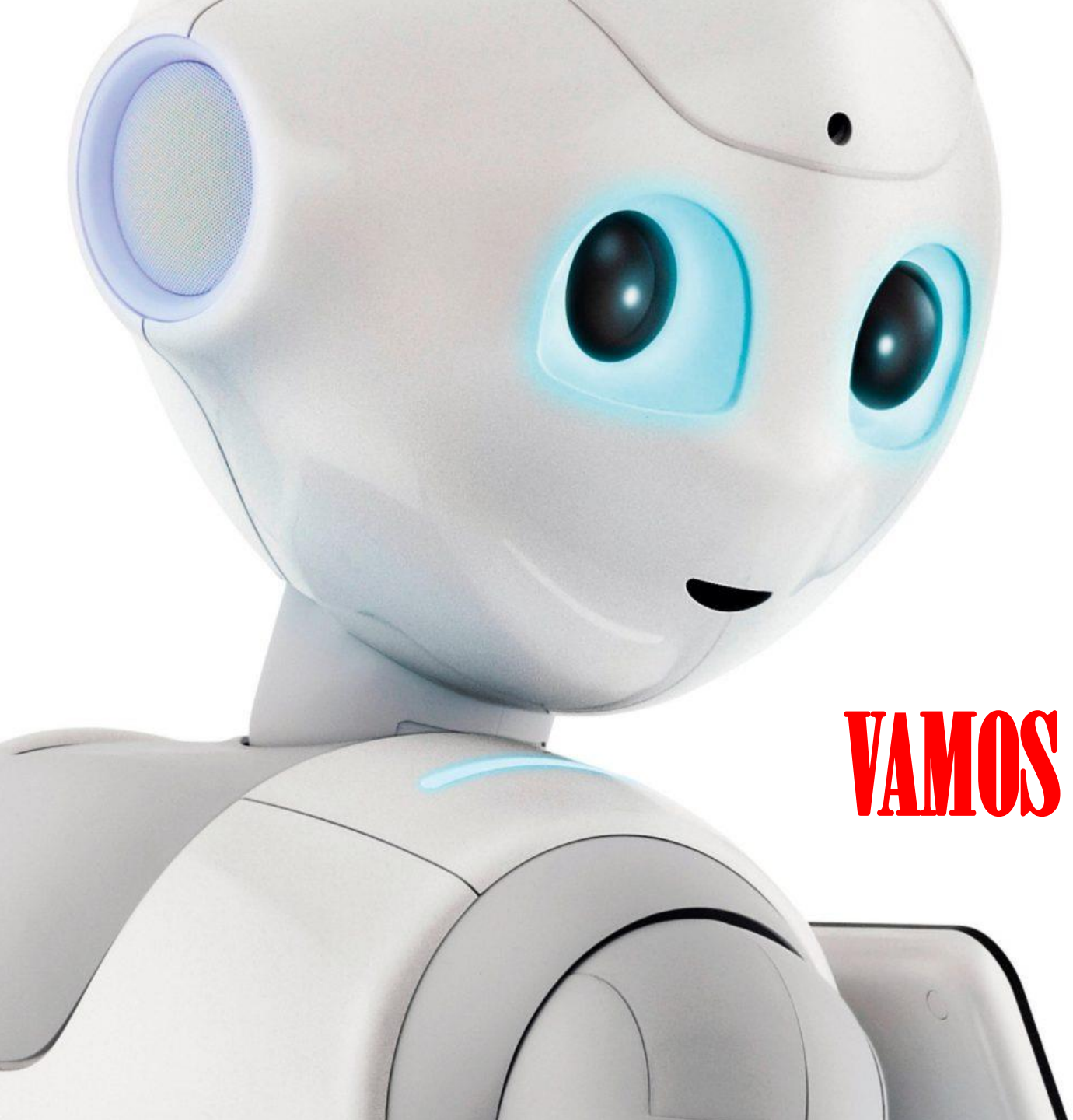
Exemplo de diagrama de classes



Exemplo de diagrama de classes



Universidade Tuiuti do Paraná



VAMOS PRATICAR?

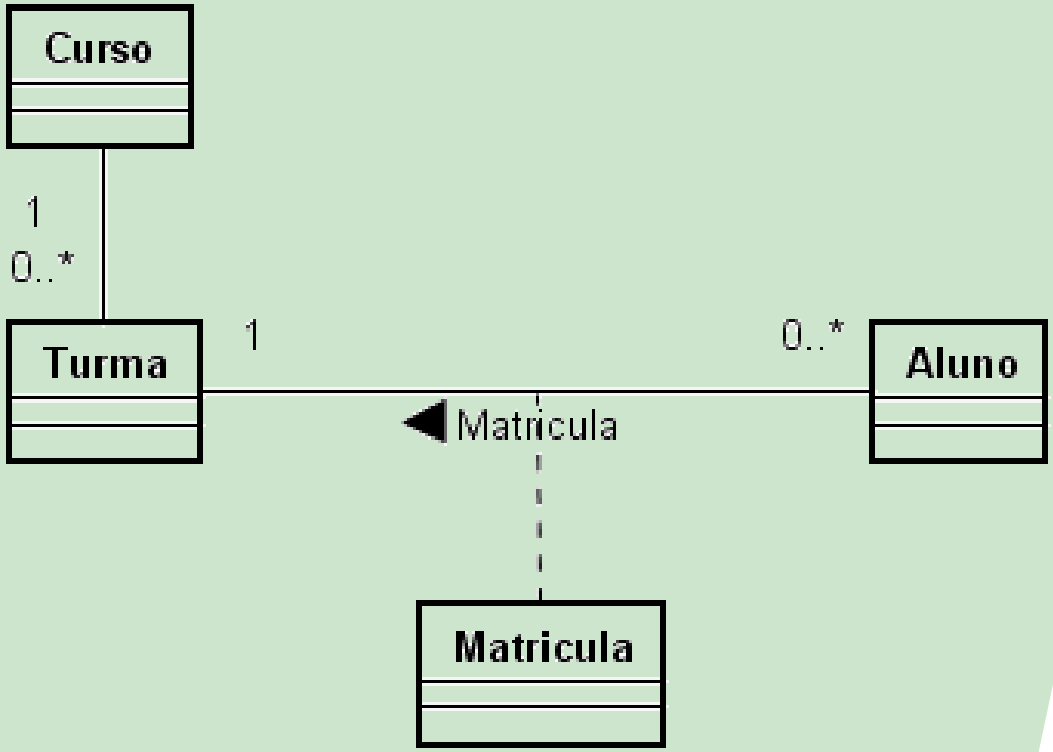
Universidade Tuiuti do Paraná

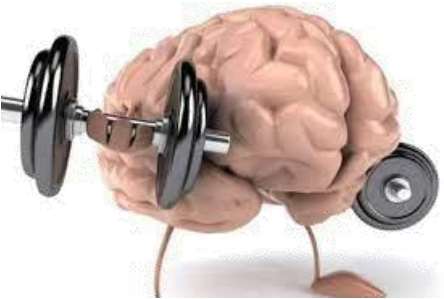
EXERCÍCIO 1: CONTROLE DE CURSO

Construa o digrama de classes (classes e relacionamentos) para atender as situações abaixo:

- Um curso pode ter muitas turmas, porém, uma turma se relaciona exclusivamente com um único curso
- Uma turma pode ter diversos alunos matriculados, no entanto, uma matrícula refere-se exclusivamente a uma determinada turma
- Um aluno pode realizar muitas matrículas, mas cada matrícula refere-se exclusivamente a uma turma específica e a um único aluno

RESPOSTA EXERCÍCIO 1





Obrigada!