



INSTITUTO FEDERAL
Rio Grande do Sul



Diagrama de Classes

Análise e Projeto de Software - 3º semestre

Tecnologia em Análise e Desenvolvimento de Sistemas

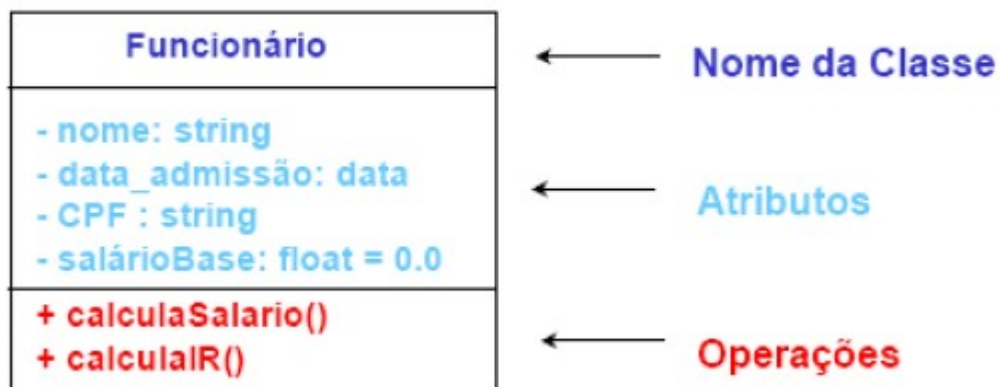
Prof^a.Dr^a. Narúsci Bastos

narusci.bastos@riogrande.ifrs.edu.br

Diagrama de Classes

- Oferecem uma notação gráfica formal para a modelagem de classes e seus relacionamentos
- **Relacionamentos** representam as conexões entre diferentes classes, indicando como elas interagem ou dependem umas das outras
 - Eles ajudam a definir as associações e interdependências que refletem as relações no domínio do problema, proporcionando um entendimento mais claro de como as classes se conectam no sistema

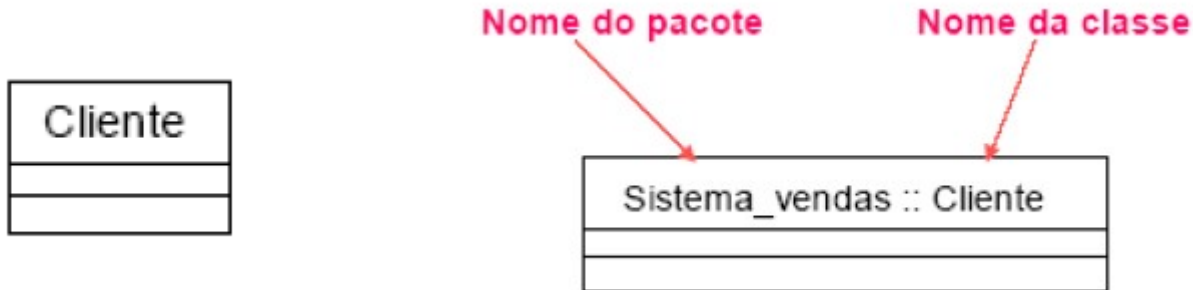
Notação



Nome da Classe

Pode ser:

- Nome simples
- Nome do caminho: o nome da classe é precedido pelo nome do pacote (subsistema) em que a classe existe



Atributos

- Características de uma classe

- **Exemplos:**

idade

- endereço : String

+ CPF : int

nroCorrentista : int

Operações / Métodos

- Uma operação é um serviço que pode ser requisitado a qualquer objeto da classe, afetando seu estado (afetando valor de seus atributos)

Ex: consultar()

Visibilidade

- **Público (+):** o que pode ser visto pelas operações de outras classes
- **Protegido (#):** o que pode ser visto apenas pelas operações da própria classe e por suas classes herdeiras
- **Privado (-):** o que pode ser visto apenas pelas operações da própria classe

Estereótipos

- **Mecanismos de extensão da UML**
 - Podem ser utilizados em diversos diagramas
 - Graficamente, um estereótipo é representado por um nome entre dois sinais de menor e dois sinais de maior (<< >>)
- **Exemplos:**
 - No diagrama de classes, podem ser utilizados para classificar o comportamento dos métodos, por exemplo <<construtor>>

Relacionamentos entre classes

- Os relacionamentos determinam as ligações entre as classes
- **Principais TIPOS:**
 - Associação
 - Agregação e Composição
 - Generalização

Associação simples

- Uma associação é um relacionamento estrutural que descreve uma ligação (conexão entre as classes ou objetos)

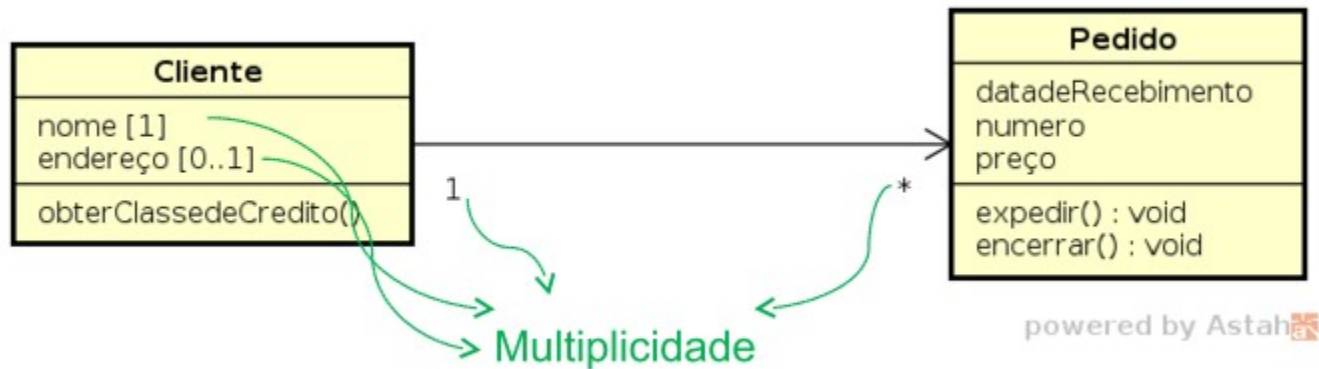


- Navegabilidade



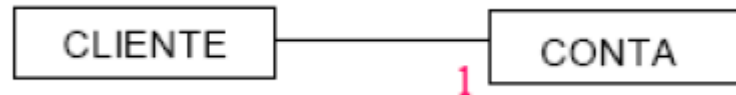
Multiplicidade

- A multiplicidade indica a quantidade de ocorrências de um conceito
- Ela é aplicada para atributos, classes e associações

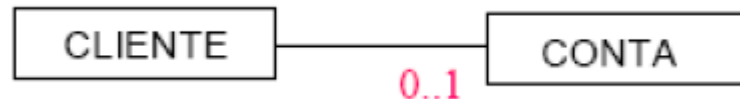


Multiplicidade

- (1-1): cliente tem uma e somente uma conta



- (0-1): cliente pode ter uma conta



- (1-N): cliente tem no mínimo uma conta, mas pode ter mais



- (0-N): cliente pode ter várias contas



Tipos de Associação

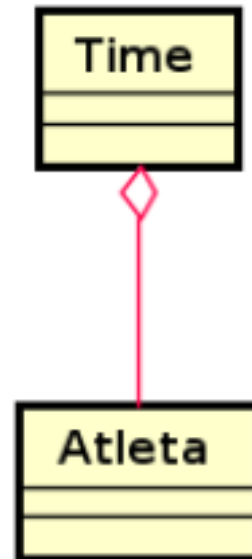
- Agregação
- Composição

→ **Agregação** e **composição** são dois tipos de relacionamentos "todo-parte" nos diagramas de classes, que expressam a relação entre um objeto que representa o "todo" e objetos que representam suas "partes"

→ Embora pareçam semelhantes, cada um representa uma relação com um grau diferente de dependência e vínculo entre os objetos

Agregação

- Agregação é um tipo especial de associação
- Representa uma relação **todo-parte** (fraco)
- Tenta demonstrar que as informações de um objeto todo precisam ser complementadas pelas informações de um objeto parte
→ Quando o todo é eliminado, as partes continuam existindo



Agregação

- Representa uma relação **todo-parte** (fraco)
- Tenta demonstrar que as informações de um objeto todo precisam ser complementadas pelas informações de um objeto parte
→ Quando o todo é eliminado, as partes continuam existindo

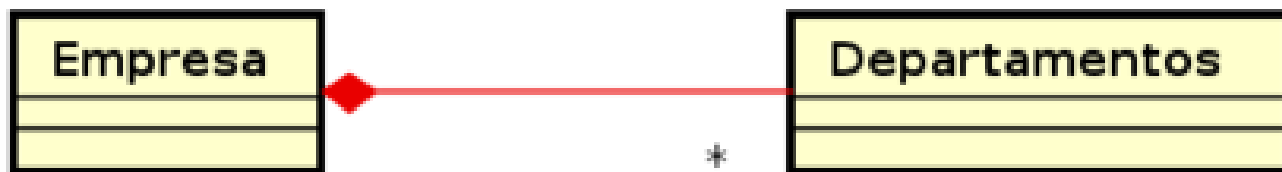
Ex: Uma classe "Biblioteca" e uma classe "Livro". A biblioteca "agrega" livros, pois cada livro existe independentemente da biblioteca. Um livro pode ser transferido de uma biblioteca para outra ou existir sem estar em uma biblioteca específica

– **Biblioteca** → **Todo** / entidade que agrega os livros

– **Livro** → **Parte**

Composição

- É uma variação da Agregação
- Representa uma relação **todo-parte** (forte)
- Quando uma parte é criada, sua existência deve ser coincidente com o todo
- Quando um todo é eliminado, suas partes também devem ser eliminadas



Composição

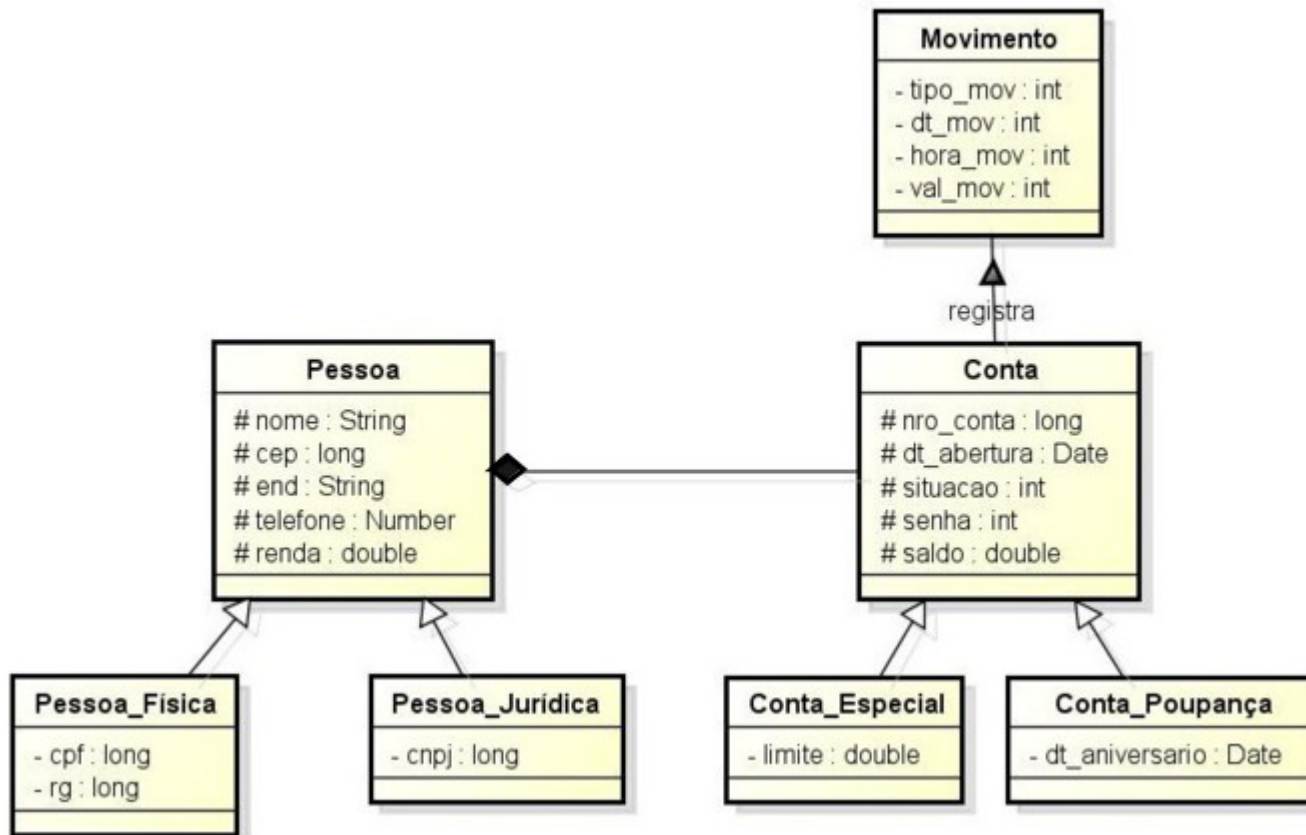
- Quando uma parte é criada, sua existência deve ser coincidente com o todo
- Quando um tudo é eliminado, suas partes também devem ser eliminadas

Ex: Imagine uma classe "Casa" e uma classe "Quarto". A casa é composta por quartos, mas se a casa for demolida, os quartos também deixarão de existir, pois eles dependem diretamente da existência da casa.

Generalização

- A generalização é o compartilhamento de atributos, operações e relacionamentos entre classes com base em um relacionamento hierárquico (**herança**)
 - **Superclasses e subclasses**
 - Subclasses herdam os atributos e métodos da superclasse, permitindo ainda modificações nos mesmos (polimorfismo)
- **Polimorfismo:** Redecaração de métodos herdados por uma classe. Esses métodos, embora semelhantes, diferem de alguma forma da implementação utilizada na super-classe, sendo necessário, portanto, reimplementá-los na sub-classe

Exemplo - Sistema Bancário





INSTITUTO FEDERAL
Rio Grande do Sul



Diagrama de Classes

Análise e Projeto de Software - 3º semestre

Tecnologia em Análise e Desenvolvimento de Sistemas

Prof^a.Dr^a. Narúsci Bastos

narusci.bastos@riogrande.ifrs.edu.br

Conceitos “Extras”

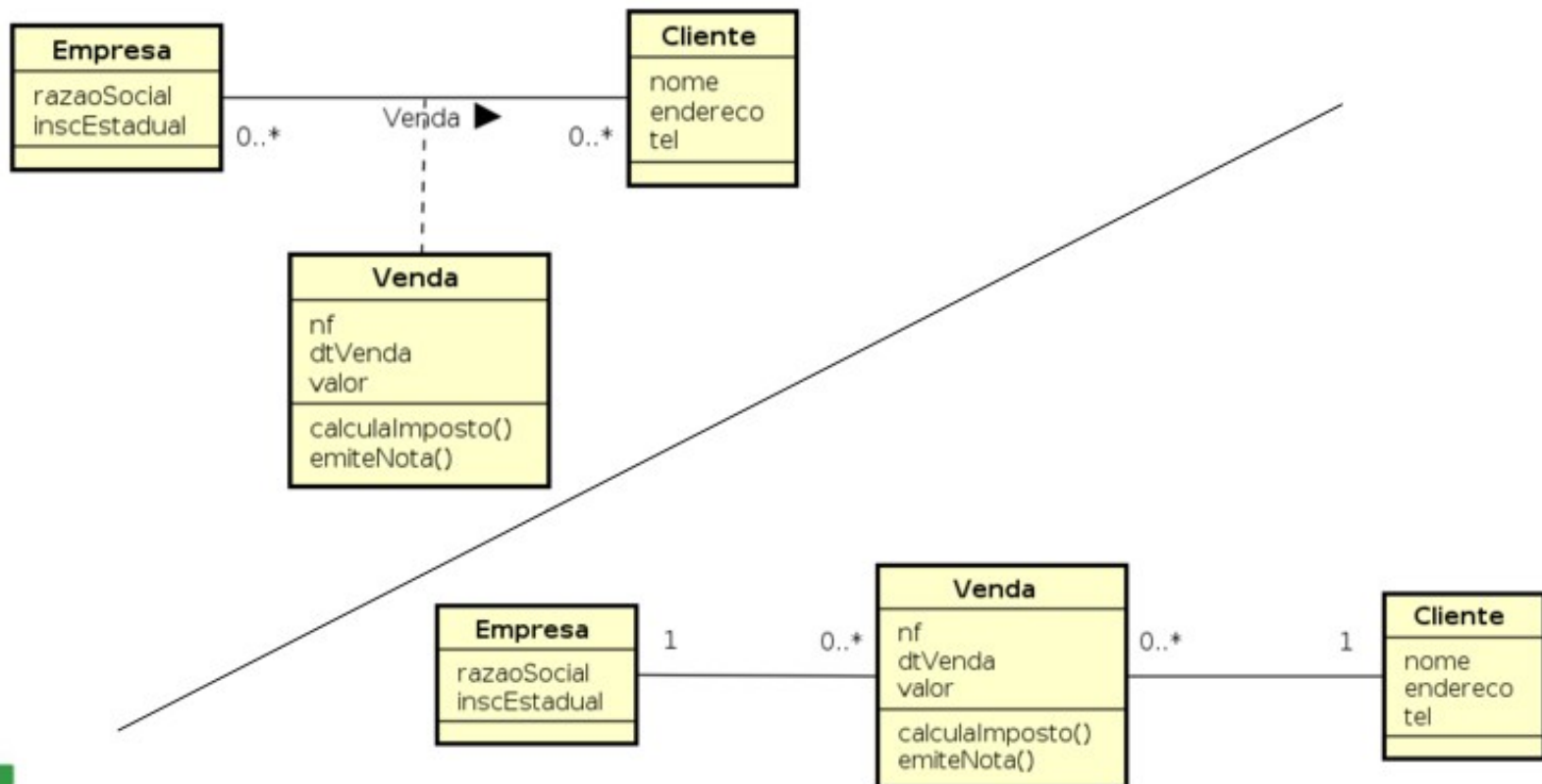
- Classe Associativa
- Classe Concreta x Classe Abstrata
 - Interface

Classe associativa

- Classes associativas são utilizadas quando a associação entre duas classes possui multiplicidade de muitos para muitos (ambas extremidades)
- Esta classe é necessária para armazenar atributos transmitidos pela associação de ambas as classes

Classe associativa

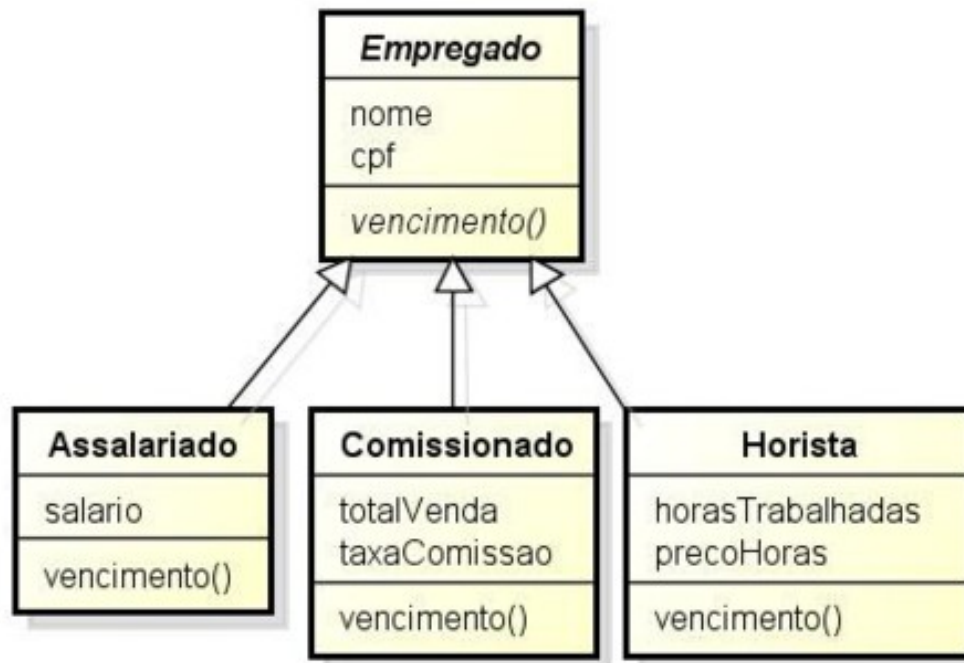
- É uma classe derivada de uma associação



Classe abstrata

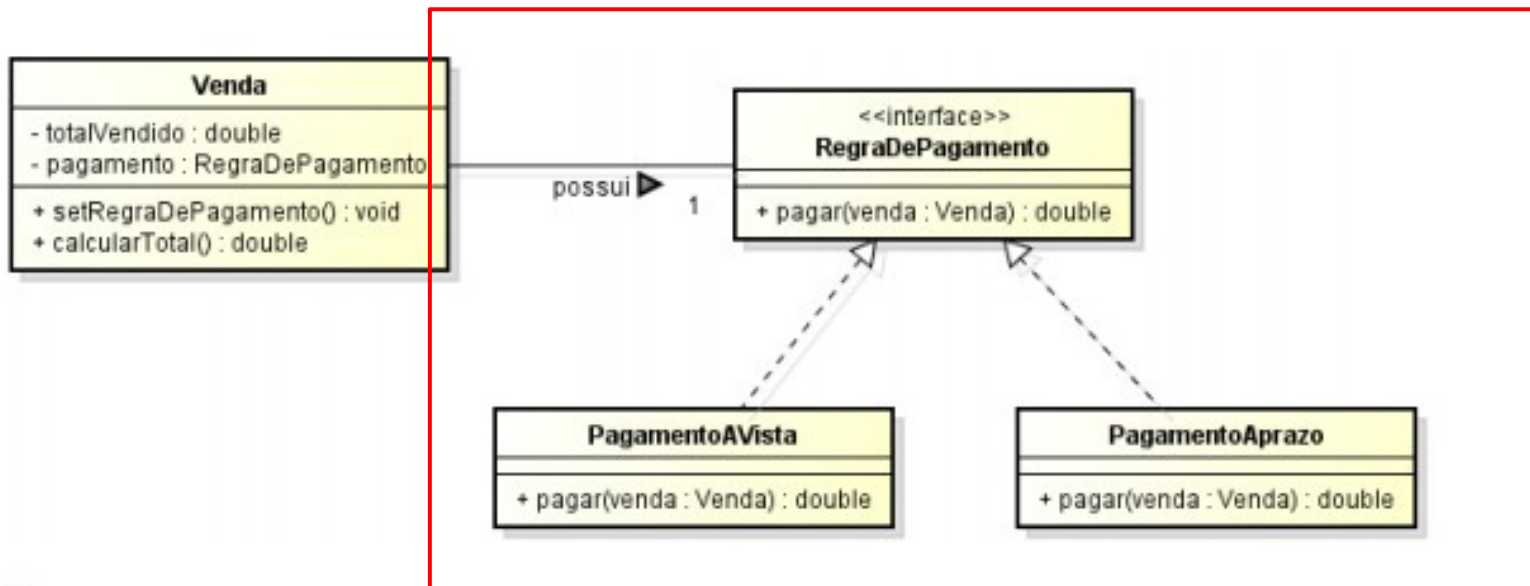
- **Concreta:** pode ter instâncias
- **Abstrata:** não pode ter instâncias
 - Contém uma ou mais operações abstratas:
 - Uma operação abstrata possui apenas a assinatura (cabeçalho) mas não possui implementação
 - A classe abstrata é sempre uma superclasse que não possui instâncias, e só tem sentido se tiver subclasses que implementem as operações abstratas
 - O nome de uma classe abstrata é escrito **em *itálico***, ou possui a restrição {abstract}.

Exemplo - Classe Abstrata



Interface

- Interface é uma “classe abstrata” que não possui atributos e que todas as operações são públicas e abstratas
- Define o contrato que estabelece os métodos obrigatórios
- Notação:





INSTITUTO FEDERAL
Rio Grande do Sul



Diagrama de Classe

Análise e Projeto de Software - 3º semestre

Tecnologia em Análise e Desenvolvimento de Sistemas

Prof^a.Dr^a. Narúsci Bastos

narusci.bastos@riogrande.ifrs.edu.br

Cronograma

- 13/11 - Aula + Exercícios
- 20/11 - Aula + Exercícios
- 27/11 - Mini projeto
- 04/12 - Aula de Revisão
- 11/12 - Atividade Avaliada
- 18/12 - Notas / Revisão exame

Notas:

Exercícios - 3,0

Mini projeto - 3,0

Atividade Avaliada - 4,0