

RAFAEL DESIDERIO

Formação Acadêmica

Técnico em Logística

Programação de Jogos Digitais

Graduado em Análise e Desenvolvimento de Sistemas

Pós Graduação MBA em Arquitetura de Software

Experiência Profissional

10 anos na área de Tecnologia (T.I)

Monitor de Laboratório

Workshop em Eventos de Inovações Tecnológicas

Analista de Infra

Desenvolvedor de sistemas

Professor de graduação FIAP

profrafael.desiderio@fiap.com.br





FIAP



| Conteúdo Programático (anual)

- ☐ Programação Orientada a Objetos
- ☐ Diagrama de Classe (UML)
- ☐ Classes, Atributos, Tipos Primitivos e Operações
- ☐ Plataforma JAVA / IDE IntelliJ
- ☐ JAVA, Modelos, Pacotes
- ☐ Encapsulamento, Herança e Polimorfismo
- ☐ Vetor de Objetos
- ☐ ArrayList/Collections Framework;
- ☐ Design Pattern BackEnd
- ☐ JDBC
- ☐ DTO,s
- ☐ Interfaces
- ☐ Maven Project
- ☐ Tratamentos de Erros/Exceções (Try / Catch)
- ☐ Consumo de APIs externas
- ☐ Integração com front end (API Restful)
- ☐ Revisão de Todo o Conteúdo Programático

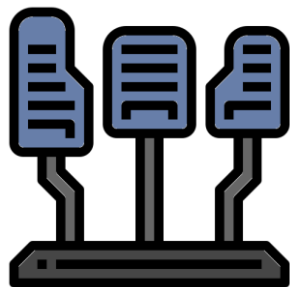
DOMAIN DRIVEN DESIGN

Prof. Rafael Desiderio

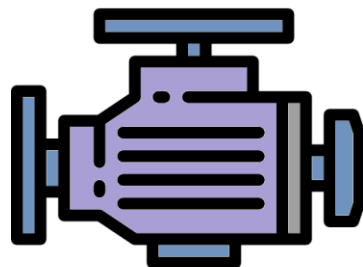
01 – ORIENTAÇÃO A OBJETOS

| O que é Programação Orientada à Objetos ?

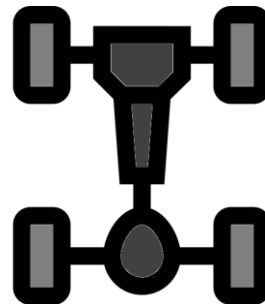
- É um paradigma (modelo, padrão) de **análise**, **projeto** e **programação** de sistemas de informação, baseado na composição e interação entre diversas unidades de software chamadas de **objetos**;



Pedais



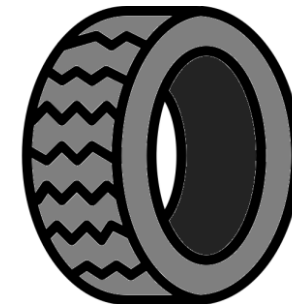
Motor



Chassi



Portas



Pneus

| Programação Orientada à Objetos

- **Programação**

É um método padronizado para **expressar instruções** para um computador através de uma **linguagem de programação**.

- **Linguagem de Programação**

É um conjunto de regras sintáticas (gramática) e semânticas (significado) usadas para definir um programa de computador.

- **Objeto**

Na programação orientada à objetos, um objeto representa uma entidade que pode ser física, conceitual ou de software.

| Exemplos do cotidiano

Na imagem abaixo temos como exemplo uma cadeira gamer e uma bola de futebol, mas poderíamos utilizar como exemplo outros tipos destes mesmos objetos com características ou propriedades específicas.

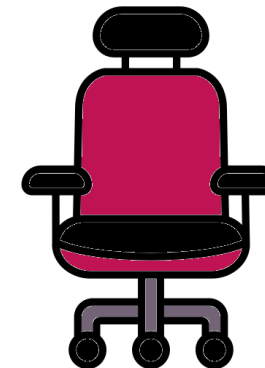
▪ Bola

- Futebol** (couro, redondo, 40cm diâmetro, branca, etc...);
- Tênis** (tecido, redonda, 5cm de diâmetro, amarela, etc...);
- Ping-Pong** (PVC, redonda, 1.5cm de diâmetro, branca, etc...);
- Futebol Americano** (couro, oval, 50cm de largura, marron, etc...);



▪ Cadeira

- Gamer** (plástico, com rodas, couro sintético, magenta, com braço, etc...);
- Praia** (alumínio, pano, de deitar, etc...);
- Escritório** (ferro, estofado macio, preta, com braço, etc...);
- Rodas** (ferro, branca, com rodas, automática, manual, etc...);



| Exemplos do cotidiano

- Bola (Material, formato ,cor ,etc...)



Características ou Propriedades

material: couro;
formato: redondo;
tamanho: 40cm;
cor: branca;

- Cadeira (Material, formato ,cor ,etc...)



Características ou Propriedades

estofado: sintético;
estrutura: redondo;
Contém_rodas: sim;
Contém_braço: sim;
Cor: magenta;

Vamos à prática

Busque ao menos **3 exemplos** do cotidiano e descreva as **propriedades** dos exemplos encontrados.

| Analogia com a Informática

▪ Sistema de Caixa Eletrônico

- **Objeto: Cliente**
 - Nome
 - Endereço
 - CPF
 - RG
- **Objeto: Conta Corrente**
 - Agência
 - Número
 - Saldo
 - **Cliente**



| Analogia com a Informática

▪ Sistema de E-Commerce

- Objeto: **Produto**
 - Nome
 - Descrição
 - Valor
- Objeto: **Estoque**
 - **Produto**
 - Quantidade
 - Prazo de Validade
- Objeto: **Cliente**
 - Cadastro
 - Senha do Cadastro
 - Nome
 - Endereço
 - CPF
 - RG



Classes

| O que é uma Classe?

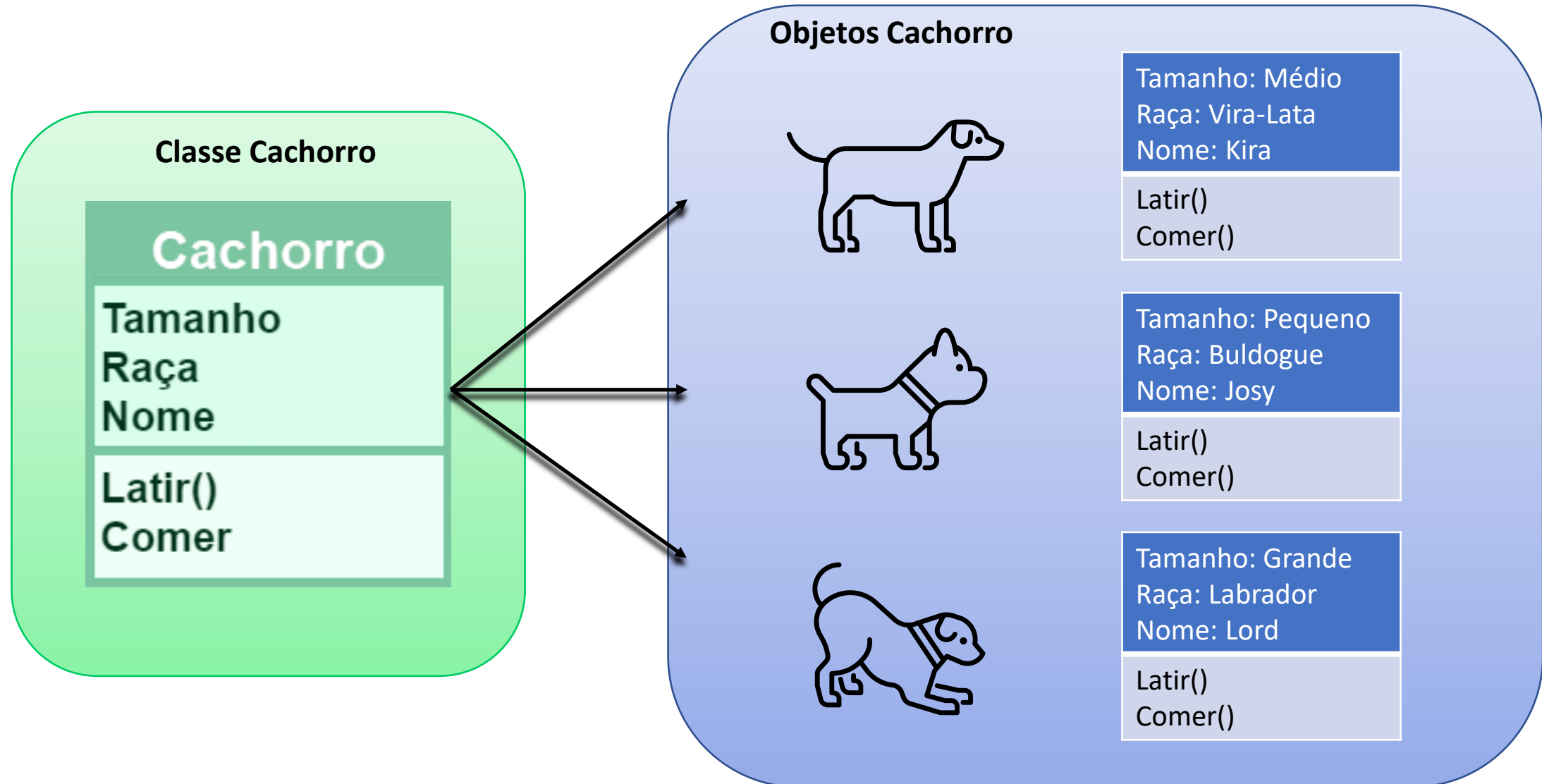
- As **abstrações** são representadas pelas **classes**;
- Uma classe deve **conter apenas os elementos necessários** para resolver um aspecto bem definido do sistema;
- A **classe é uma descrição nomeada** para um grupo de entidades (chamadas de **objetos** ou instancias de classe) **que tem as mesmas características**;



| O que é uma Classe?

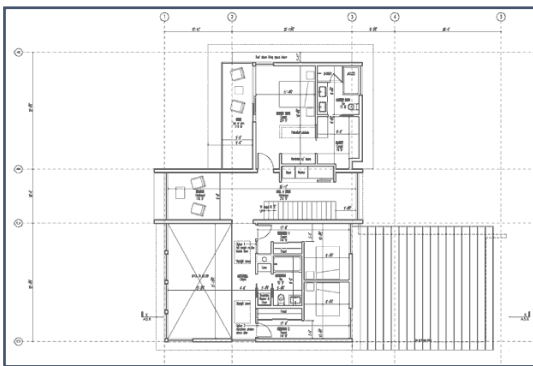
- As características são os **atributos** (propriedades, campos de dados) e as **operações** (comportamentos, métodos, funções) que podem ser executadas nesses objetos;
- Em outros termos, uma classe descreve os **serviços providos por seus objetos** e **quais informações** eles podem armazenar;
- Na programação orientada a objetos a classe é a **unidade básica de programação**;
- Todos os **programas são escritos como um conjunto de classes**, e todos códigos que você escrever devem fazer parte de uma classe;

| O que é uma Classe?



| O que é uma Classe?

- Uma **classe** é a **descrição** de um **conjunto de objetos** que compartilham os mesmos **atributos, operações, relações e semânticas**;
 - Um **objeto** é uma **instancia** de uma classe;
- Uma **classe** é uma **abstração**, uma vez que:
 - Enfatiza características relevantes;
 - Suprime outras características;



Classe



Instância



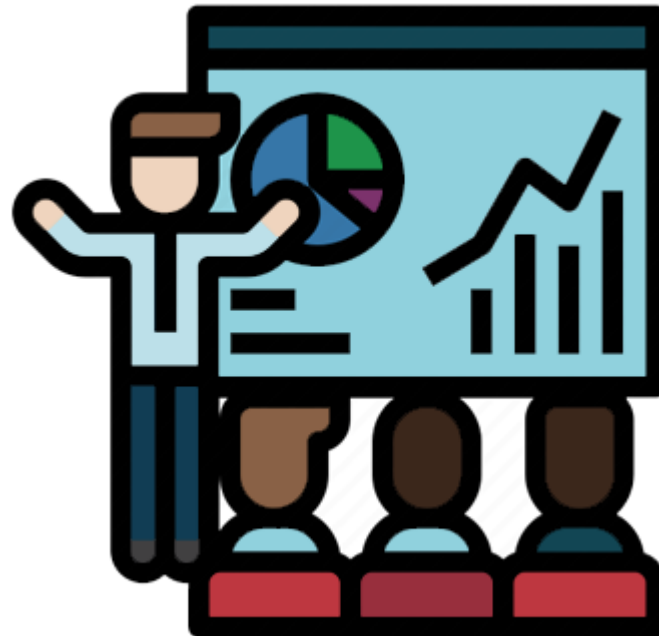
Objeto

| Exemplo de Classe

Classe Curso

Propriedades

Nome
Local
Dias oferecidos
Carga horária
Hora de início
Hora de término

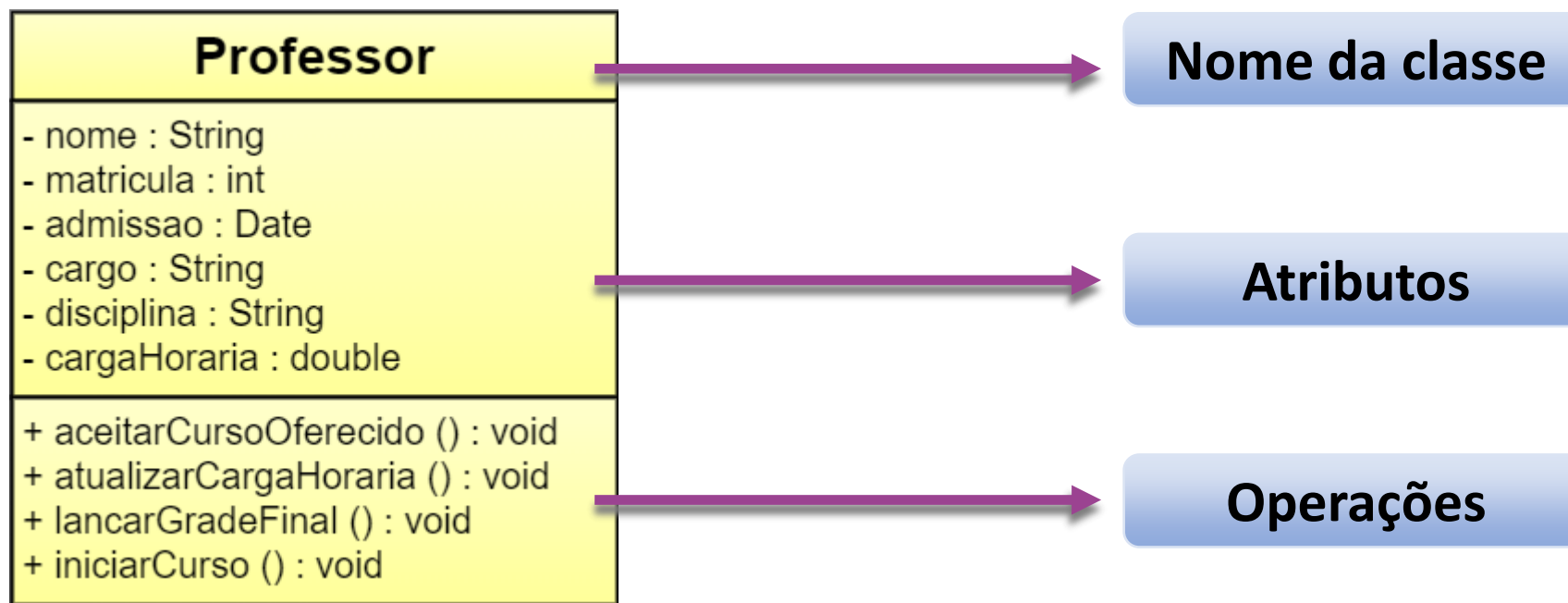


Comportamentos

Adicionar um aluno
Excluir um aluno
Obter lista de alunos
Verificar se está cheio

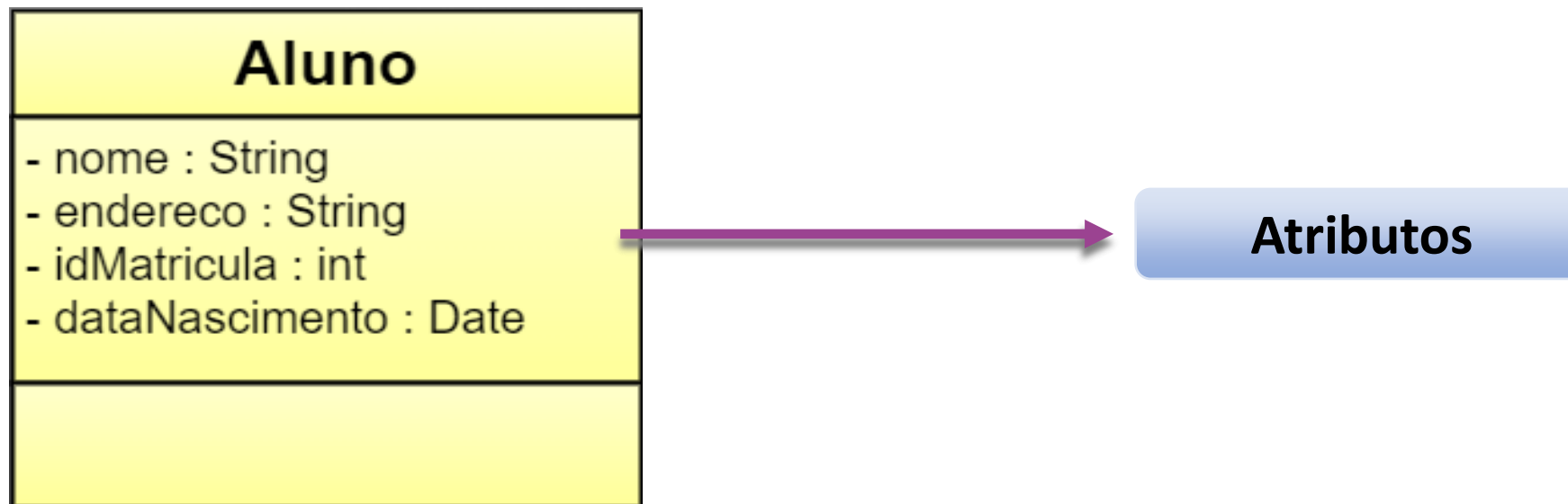
| Representação gráfica de uma Classe

- É possível representar **graficamente uma classe** através de um **diagrama de classes** (UML), este diagrama é um exemplo de uma representação da **estrutura e relações das classes** que servem de modelo para objetos;
- Uma classe é representada através de **um retângulo com três compartimentos**;



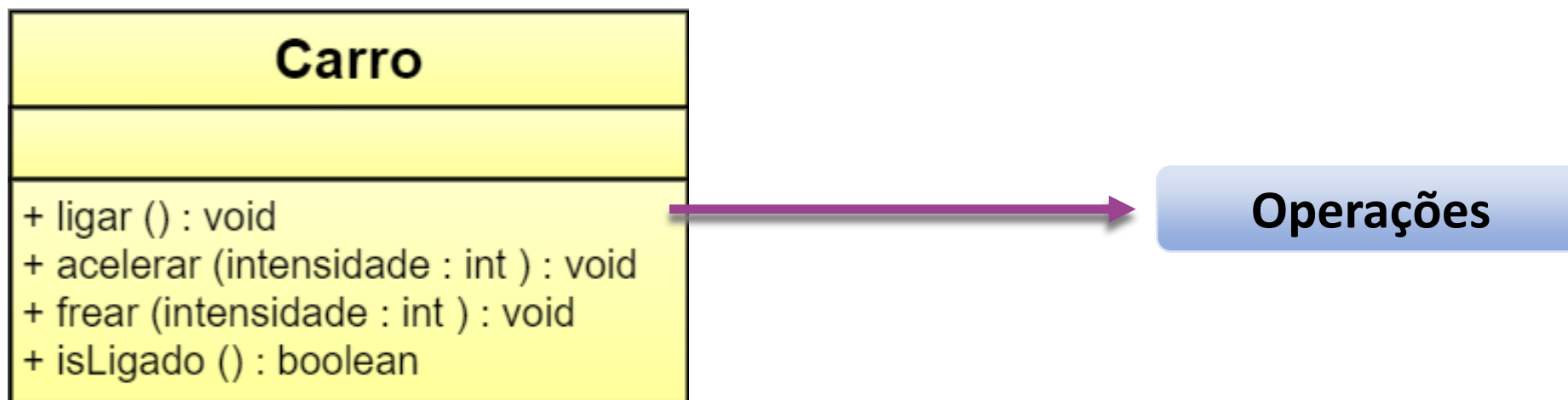
| Atributo

- Um **atributo** é o **nome** que se dá à **propriedade** de uma classe;
- O **atributo descreve** o **tipo** de **valores** que a propriedade possui;
 - Uma classe pode ter **qualquer quantidade de atributos** ou nenhum atributo;



| Operação / Métodos

- **Operação é um serviço** que pode ser solicitado a partir de um objeto para **executar um comportamento**. Uma operação tem uma assinatura, que pode restringir os parâmetros reais que são possíveis ;
- Uma **classe pode ter qualquer quantidade de operações** ou nenhuma operação;



| Relação entre Classes e Objetos

- Nesse exemplo, a classe **Pessoa** possui os seguintes atributos e operações:


Pessoa
- nome : String - genero : String - idade : int - casa : Casa - carro : Carro
+ exibirDadosPessoais () : void + exibirPatrimonio () : void



| Relação entre Classes e Objetos

- A classe **Pessoa** pode gerar vários objetos:

Objeto #1

nome = "Caroline"
idade = 25
genero = "feminino"
casa =
carro = 

exibirDadosPessoais()
exibirPatrimonio()





Objeto #2

nome = "Rafael"
idade = 38
genero = "masculino"
casa =
carro =

exibirDadosPessoais()
exibirPatrimonio()



Objeto #3

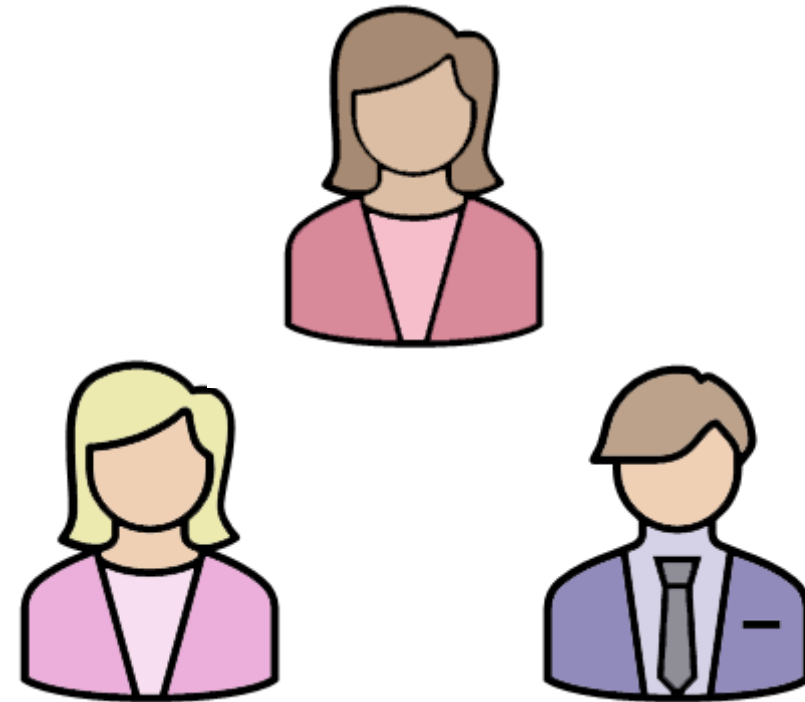
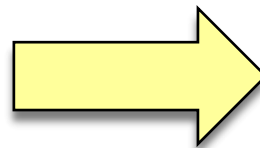
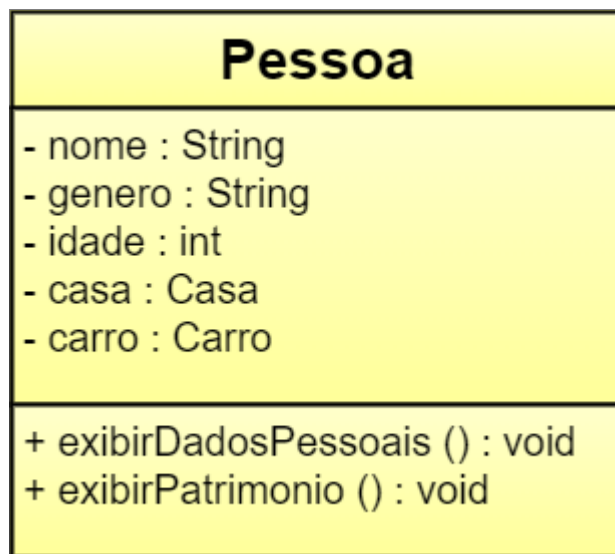
nome = "Fatima"
idade = 62
genero = "feminino"
casa = 
carro = 

exibirDadosPessoais()
exibirPatrimonio()



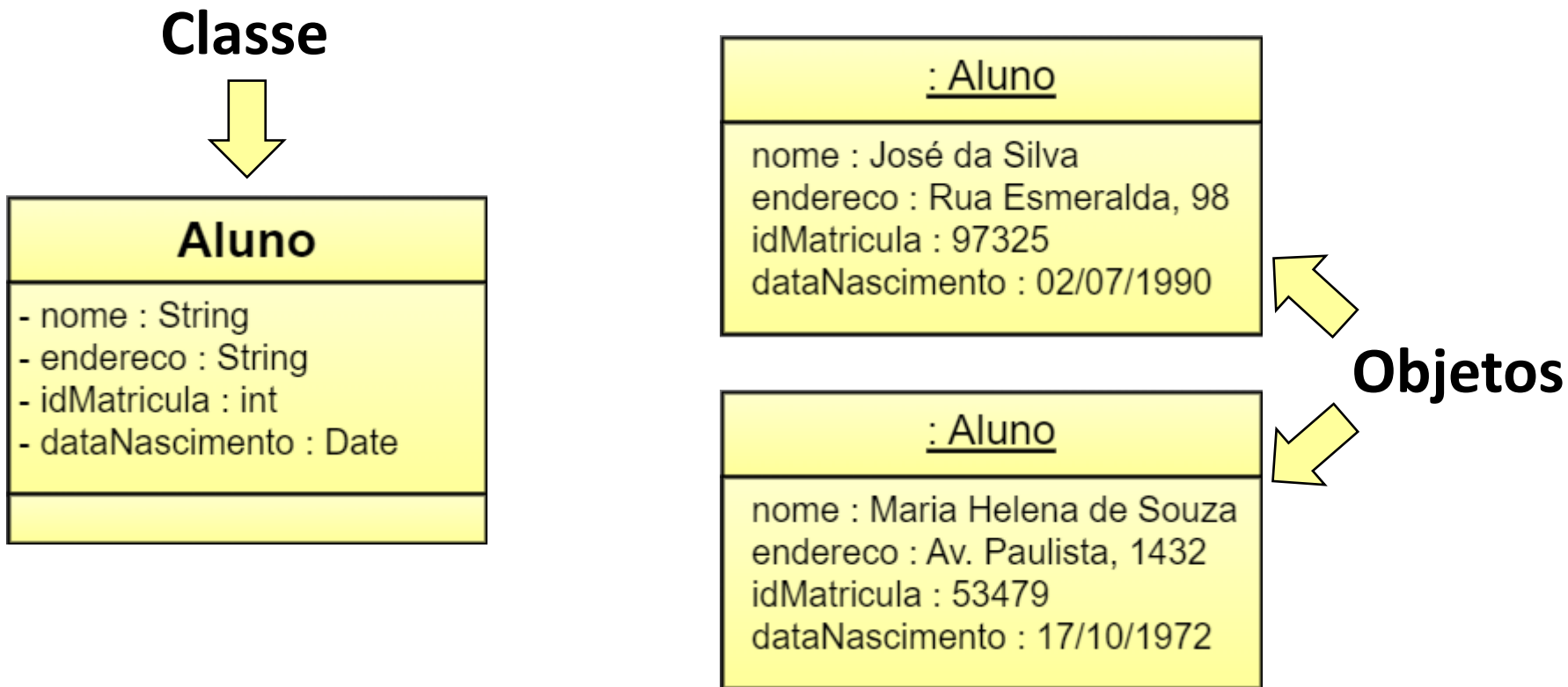
| Relação entre Classes e Objetos

- Uma **classe** é uma **definição abstrata de um objeto**;
 - Ela define a **estrutura e comportamento** de cada objeto da classe;
 - Ela serve como um **modelo** para a **criação de objetos**;
- Classes **não** são coleções de objetos;



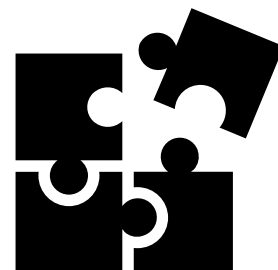
| Relação entre Classes e Objetos

- Atributos em classes e objetos:



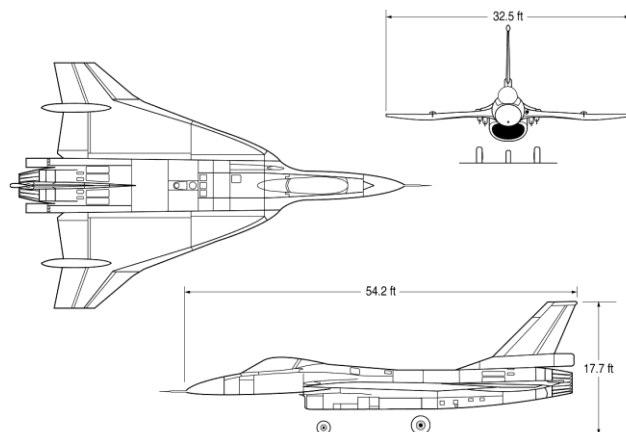
| Definição – Orientação à Objetos

- Um **conjunto de Princípios** (abstração, encapsulamento, herança e polimorfismo) **guiando a construção do software**, em conjunto com linguagens, banco de dados e outras ferramentas que suportam esses princípios. (*Object Technology - A Manager's Guide, Taylor, 1997.*)
- **Vantagens** da orientação à objetos:
 - Facilidades arquiteturais e reuso de código;
 - Reflete em modelos do mundo real;
 - Incentiva a estabilidade;
 - É adaptável à mudanças;



| O que é um Modelo Visual?

- Um **modelo** é a **simplificação da realidade**;



| Pergunta

- É **possível construir** um **prédio sem a maquete**, as plantas, a estruturação total de elétrica, gás e hidráulica?

Sim é possível (hehe), mas **NÃO** faça isto!



| Senão...

FIAP



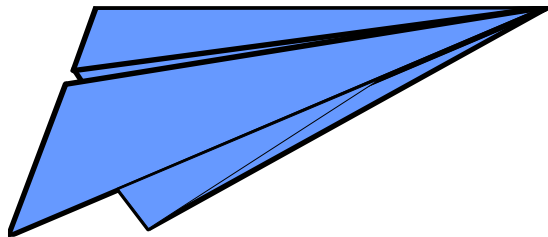
| Porque precisamos de um Modelo Visual?

- **A modelagem atinge quatro objetivos:**
 - Ajuda a visualizar um sistema como deseja que ele seja;
 - Permite especificar a estrutura ou o comportamento de um sistema;
 - Disponibiliza um modelo que orienta na construção de um sistema;
 - Documenta as decisões realizadas;
- Os **modelos de sistemas** são construídos porque não é possível **compreender o sistema em sua totalidade**;
- Os **modelos são construídos para melhor entendimento** do sistema que esta sendo desenvolvido;

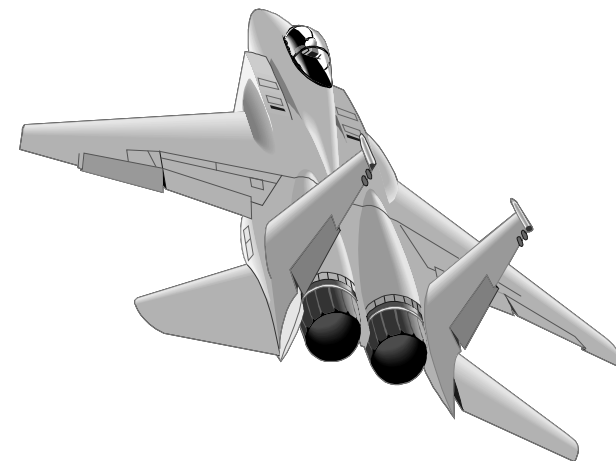
| A importância da modelagem

Menos importante

Mais importante



Avião de papel



Avião à jato

Vamos à prática

Faça a modelagem em diagrama de classe:

- Que tenha que ser realizado um cadastro de cliente em um sistema de compras e-commerce;
- Use abstração para identificar as classes, atributos e operações.

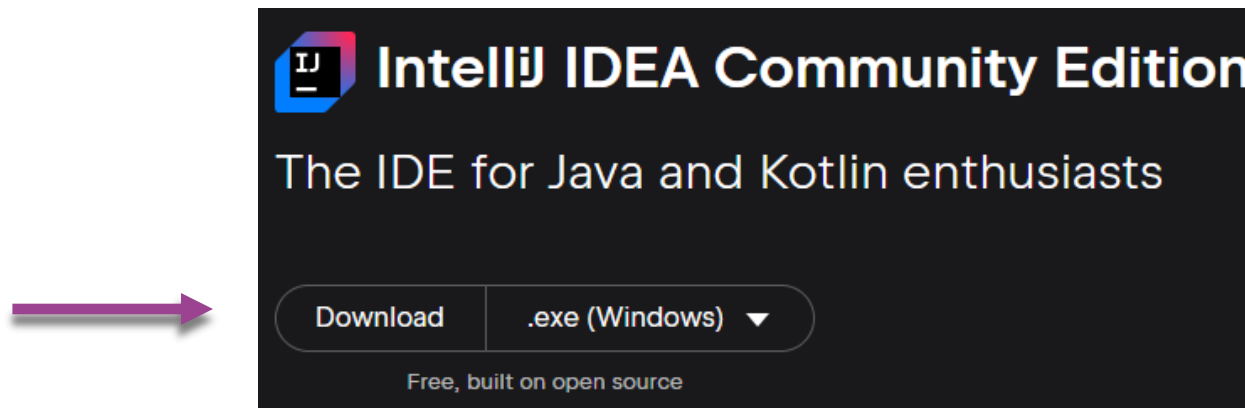
Vamos à prática

Montar as classes com os atributos definindo seus tipos:

- String - para texto
- int - para número inteiro
- double - para número real
- boolean - para verdadeiro/falso
- Date - para data

| Instalação do Java – IntelliJ IDEA Community Edition (IDE)

- Faça o download do instalador **IntelliJ IDEA Community Edition**:
 - <https://www.jetbrains.com/idea/download/?section=windows>



- Instalar o **IntelliJ** :



| Instalação do Java - JDK

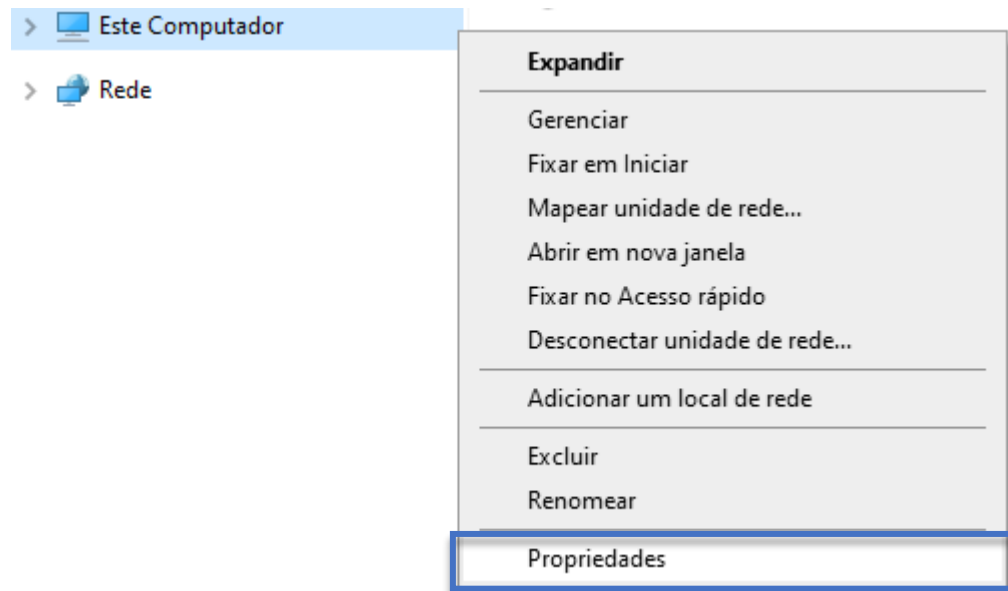
- Faça o download do **JDK**
 - <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- Instalar **JDK**
 - Incluir na variável de ambiente Path: **<diretorio_do_java>\bin**
 - Onde **<diretorio_do_java>** é o diretório de instalação do JDK

Para facilitar a configuração da Variável de Ambiente após a instalação do JDK, vamos seguir um passo a passo de exemplo...

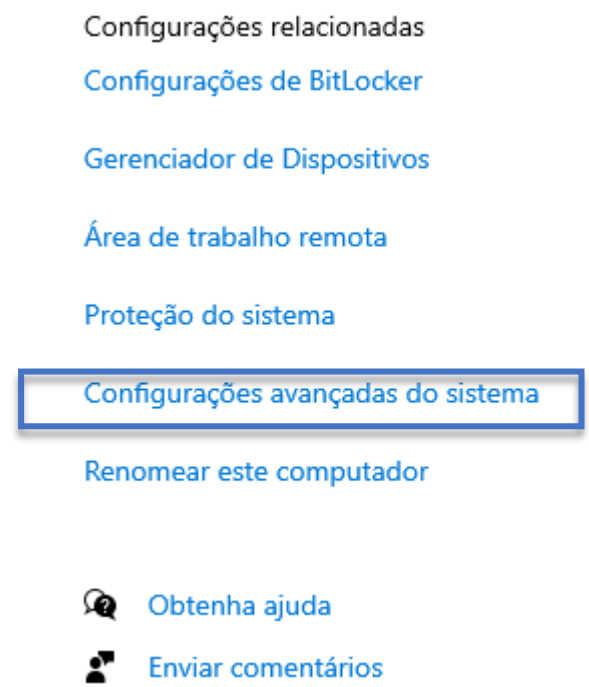


| Instalação do Java - JDK (passo a passo)

1. Clicar “Este computador” com o botão direito do mouse e ir em “Propriedades”.

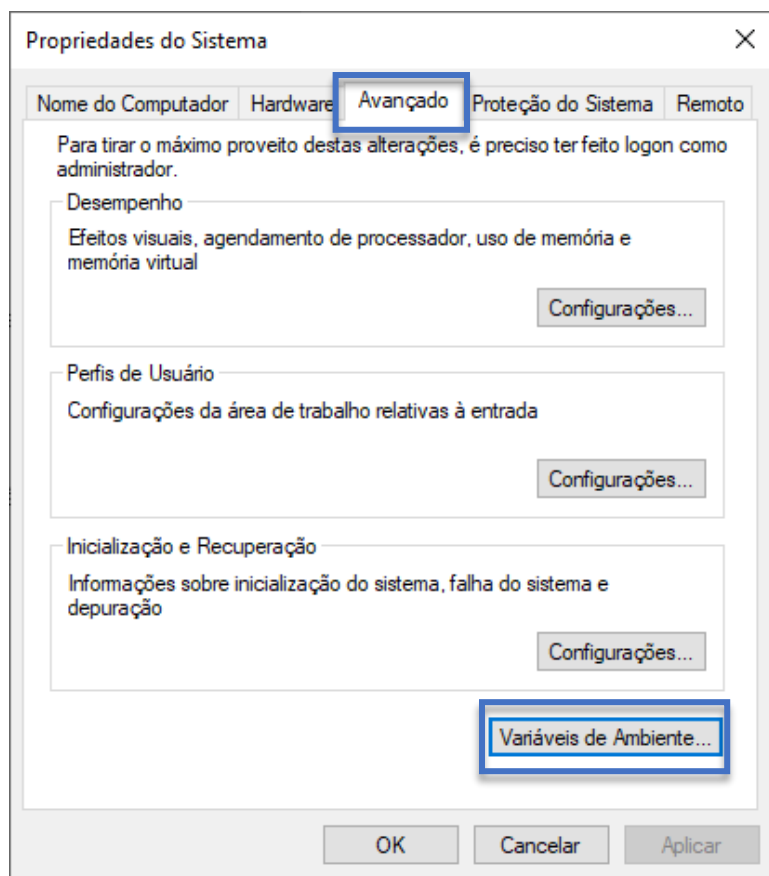


2. Abrirá a janela de configurações no canto direito, clique na opção “Configurações avançadas do sistema”.

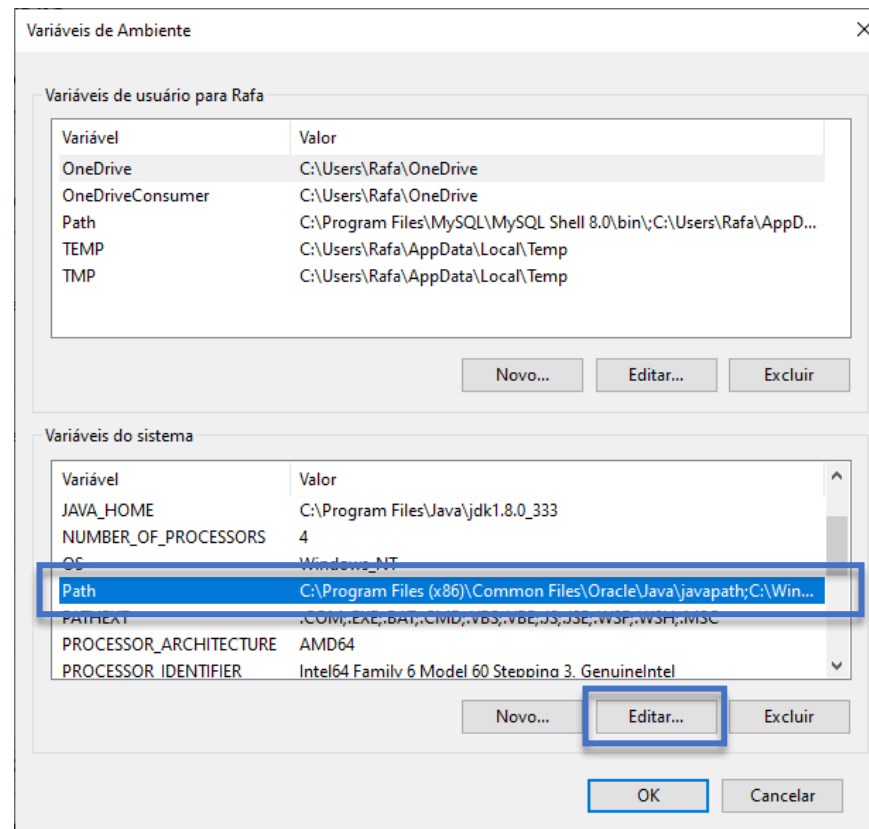


| Instalação do Java - JDK (passo a passo)

3. Na janela Propriedades do Sistema, vá na aba “avançado” e depois em “Variáveis de Ambiente”.



4. Agora em Variáveis do Sistema, selecione o “Path” e em seguida clique no botão “Editar...”



| Instalação do Java - JDK (passo a passo)

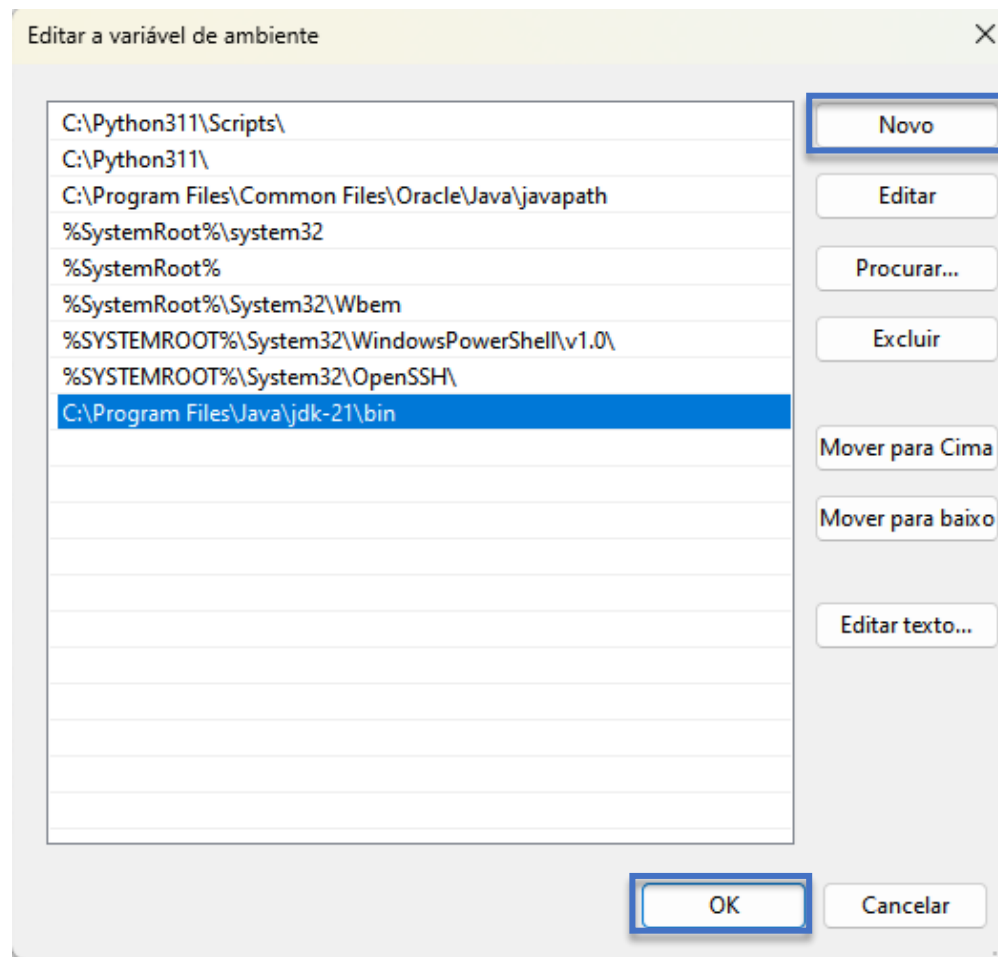
5. Vá ao local onde foi instalado o JDK em seu computador, acesse a pasta “bin” e copie o caminho do Explorer conforme apresentado na ilustração abaixo.

6. Em “Editar a variável de ambiente”, clique no botão “Novo”, no primeiro espaço vazio abaixo, cole o caminho copiado no passo anterior, depois clique no botão “OK”.



Copiar local

Colar



Copyright © 2025
Prof. Rafael Desiderio

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito do Professor (autor).