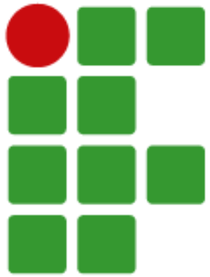


INSTITUTO FEDERAL
Sergipe

Programação de Computadores

Prof. Fausto Bernard Melo Soares

Conteúdo

INSTITUTO FEDERAL
Sergipe

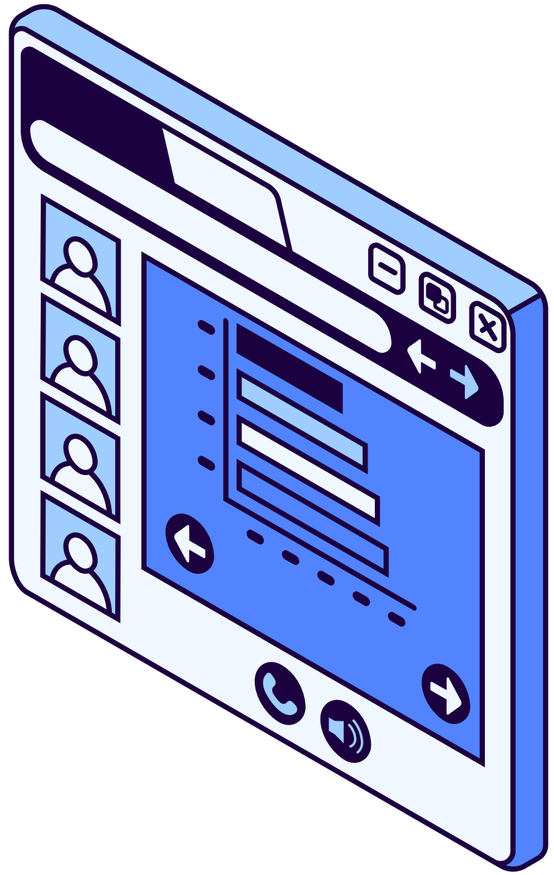
- Hardware x Software
- A tarefa de programar
- Linguagem de programação
- Conhecendo Java
- Paradigma orientado a objetos
- Aplicações da programação: desenvolvimento web, mobile, IoT

INSTITUTO FEDERAL
Sergipe

Como meu computador funciona?

Hardware x Software

INSTITUTO FEDERAL
Sergipe

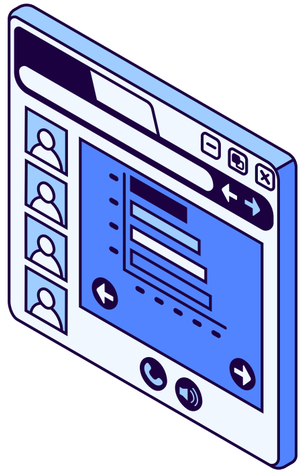


Hardware x Software

INSTITUTO FEDERAL
Sergipe

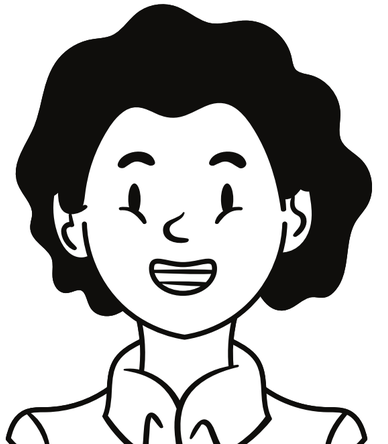
Componentes físicos

Componentes lógicos



INSTITUTO FEDERAL
Sergipe

Como eu crio um software?

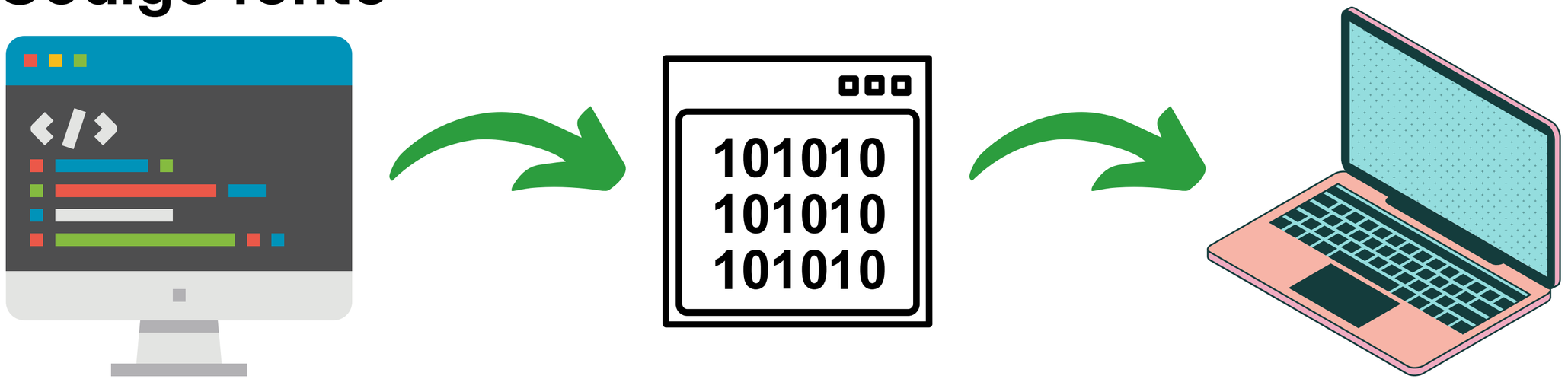


Programação

INSTITUTO FEDERAL
Sergipe

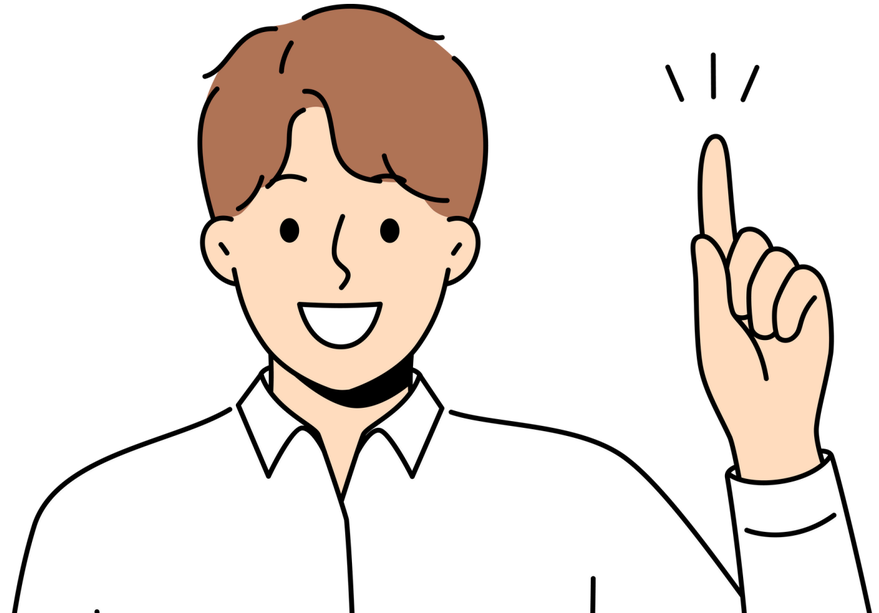
- É a tarefa de criar programas;
- Atividade de propor soluções computacionais para problemas por meio de **instruções** à máquina.

Código fonte



INSTITUTO FEDERAL
Sergipe

Como eu escrevo as instruções?



Linguagem de programação

INSTITUTO FEDERAL
Sergipe

É um conjunto de **regras** sintáticas e semânticas que define uma maneira formal de descrever uma sequência de **instruções para um computador**.

LUTZ, Mark (2019)

Linguagens de Programação

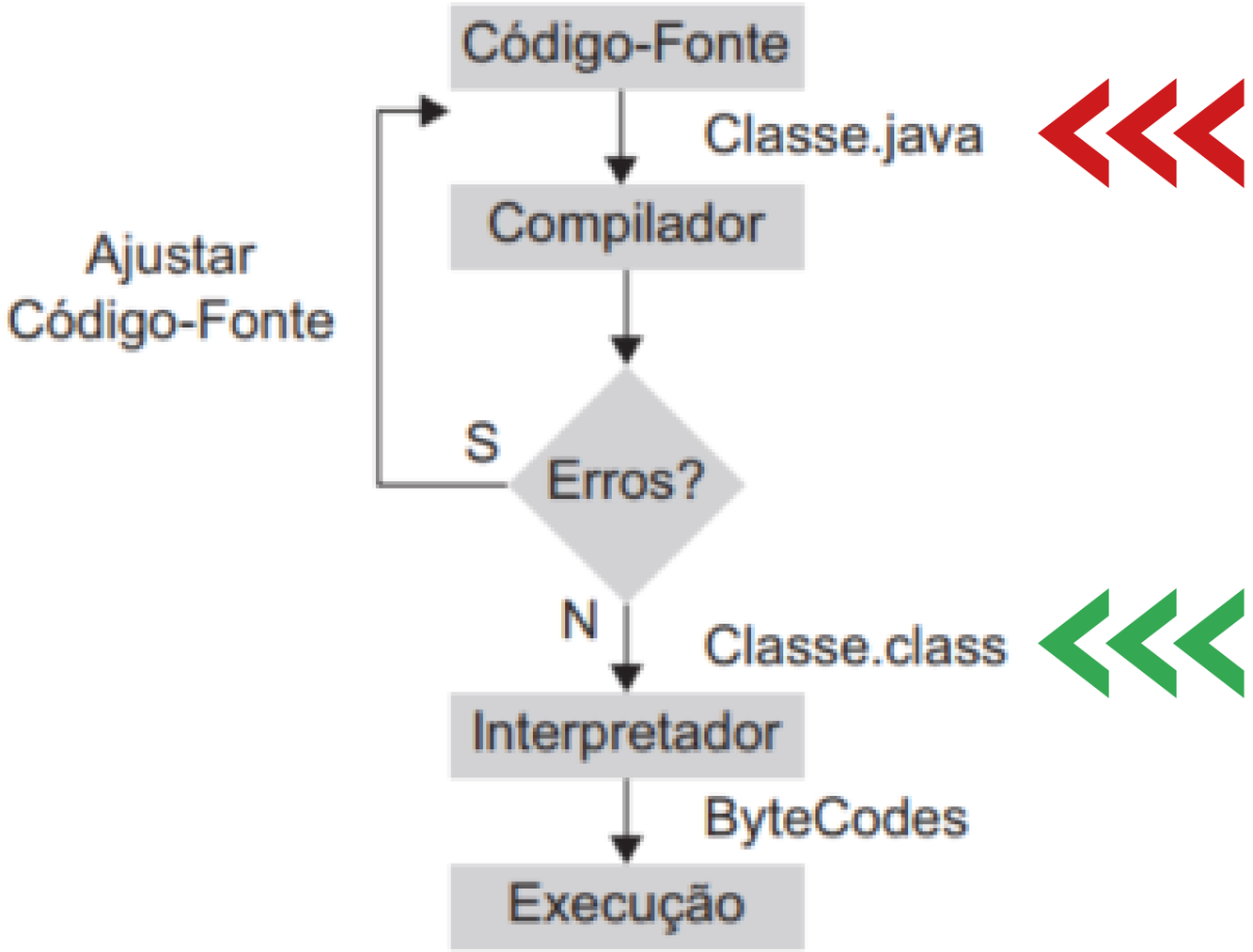
INSTITUTO FEDERAL
Sergipe

JavaScript

Conhecendo Java

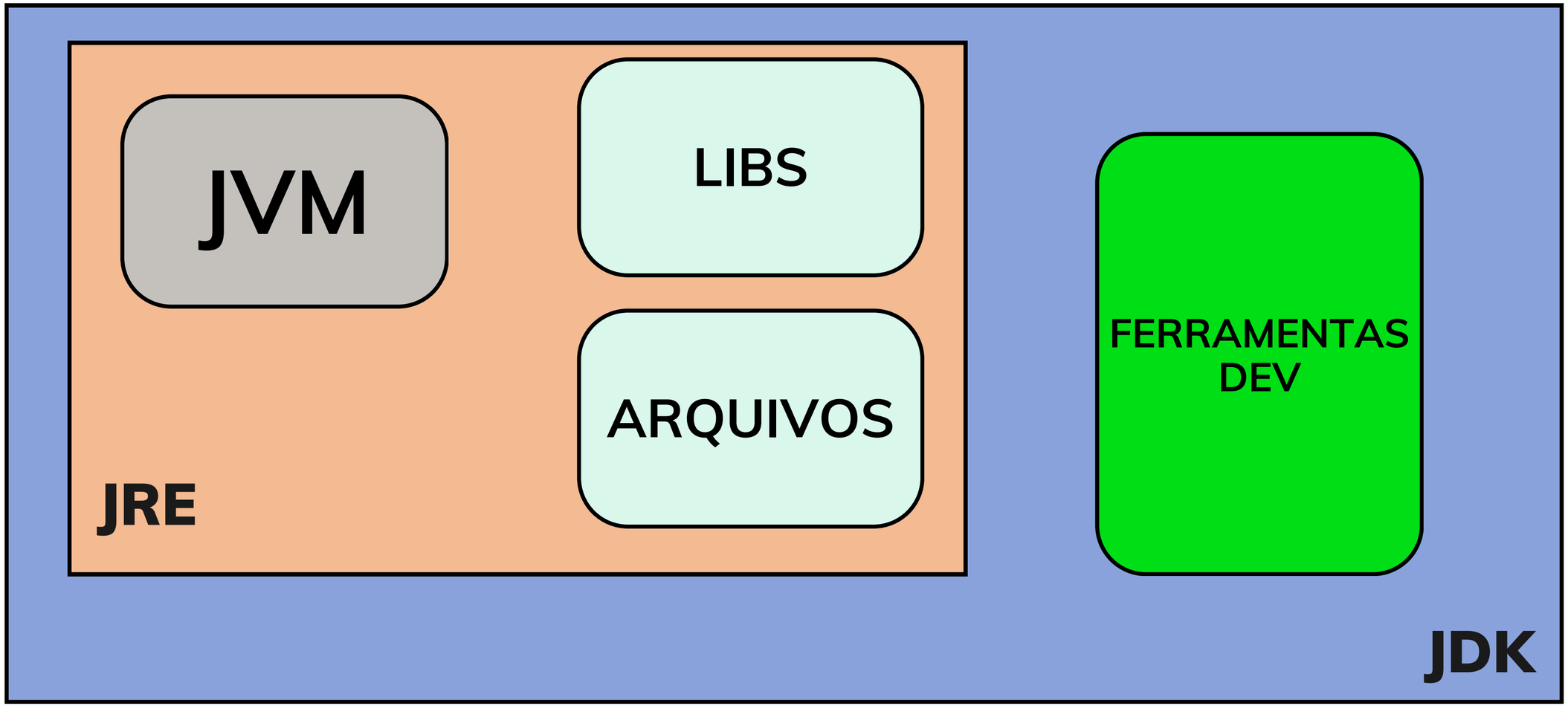
É uma linguagem de programação de **propósito geral**, orientada a objetos e baseada em classes, que foi desenvolvida pela Sun Microsystems (agora parte da **Oracle** Corporation) e lançada pela primeira vez em 1995.

Conhecendo Java



Conhecendo Java

INSTITUTO FEDERAL
Sergipe

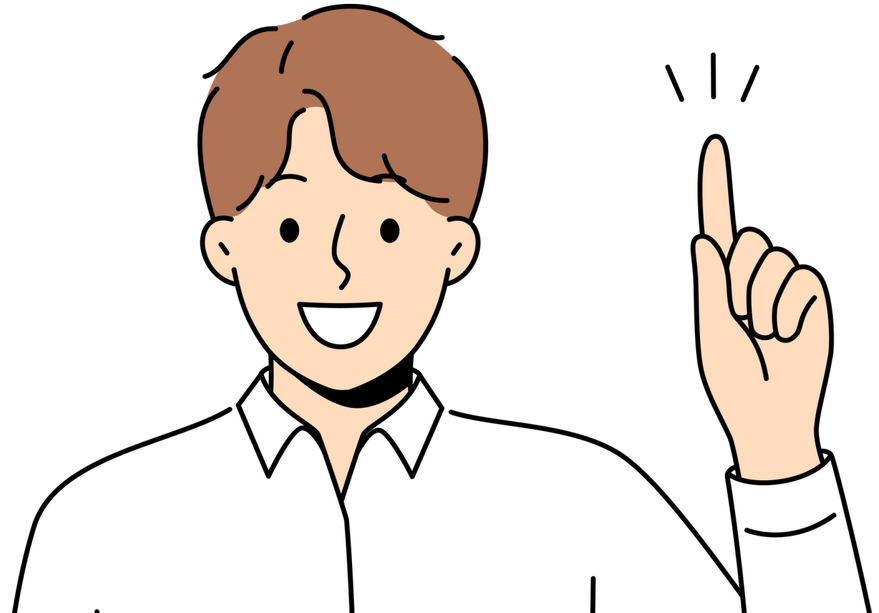


Java

Paradigmas de programação

INSTITUTO FEDERAL
Sergipe

- **Paradigma**: maneira, forma, padrão de abordar um problema.



Paradigmas de programação

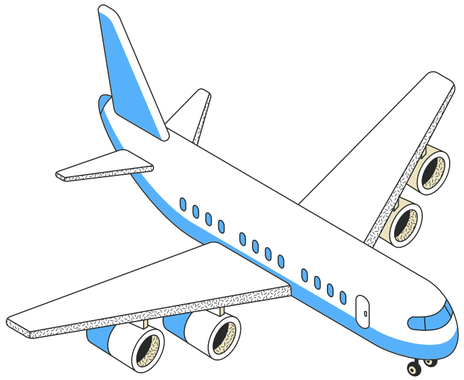
INSTITUTO FEDERAL
Sergipe

- **Orientado a Objetos**
- Imperativo
- Declarativo
- Lógico
- Funcional
- Orientado a Eventos

Orientação a Objetos

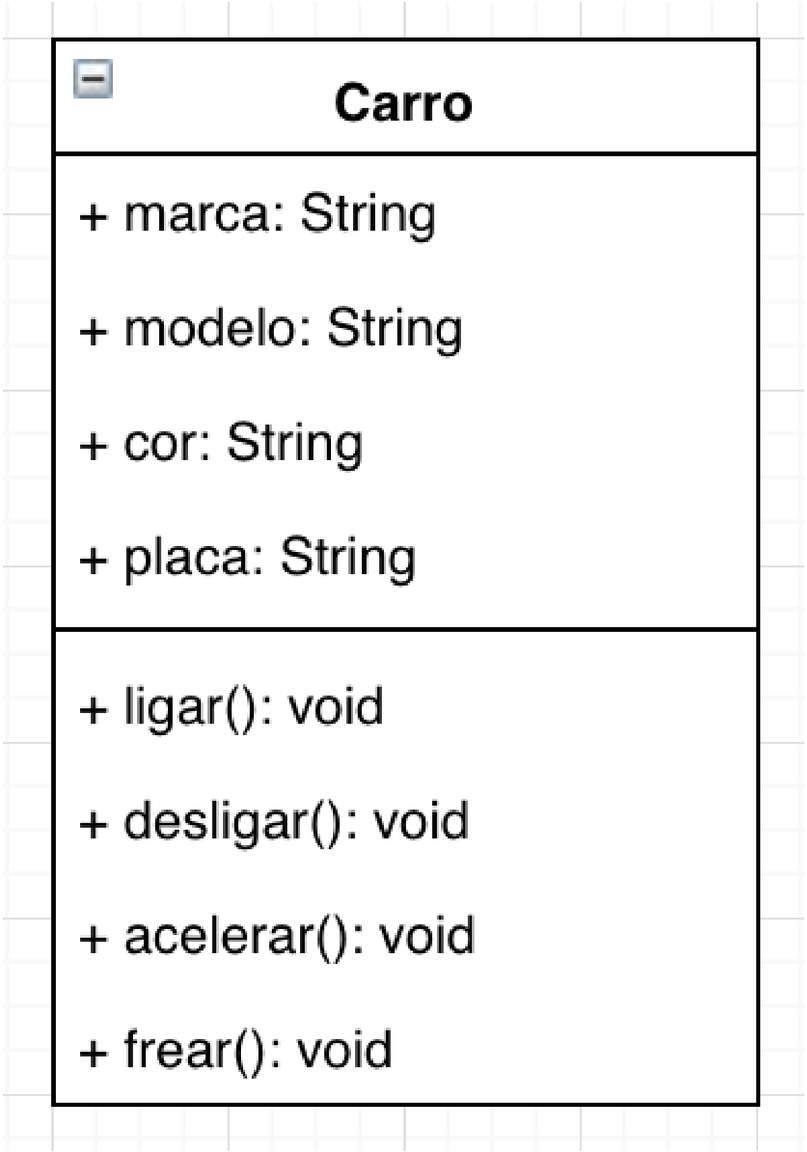
INSTITUTO FEDERAL
Sergipe

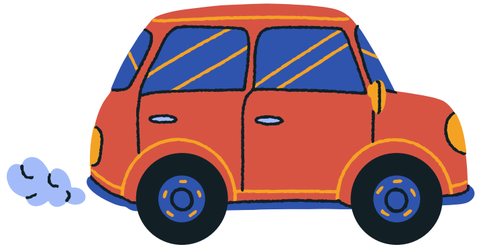
- Enxerga um sistema como um conjunto de agentes autônomos, os **objetos**, que interagem entre si.



O que são Classes e Objetos?

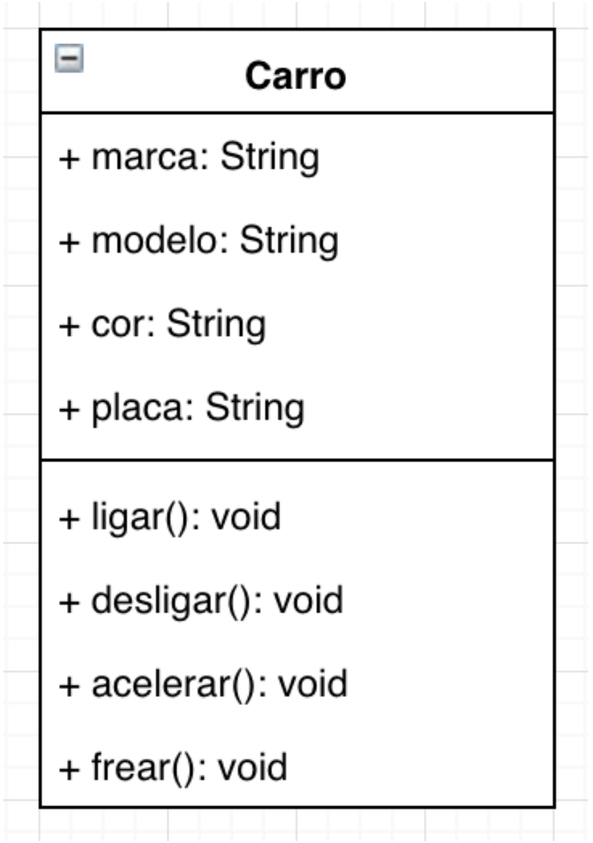
INSTITUTO FEDERAL
Sergipe





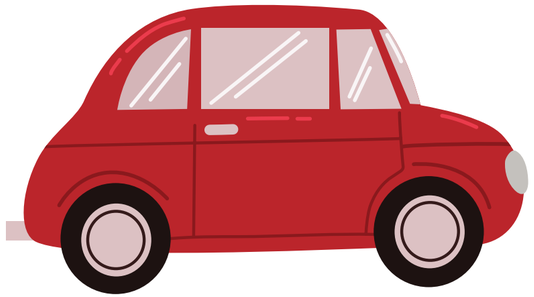
O que são Classes e Objetos?

INSTITUTO FEDERAL
Sergipe



Especificação

Implementação



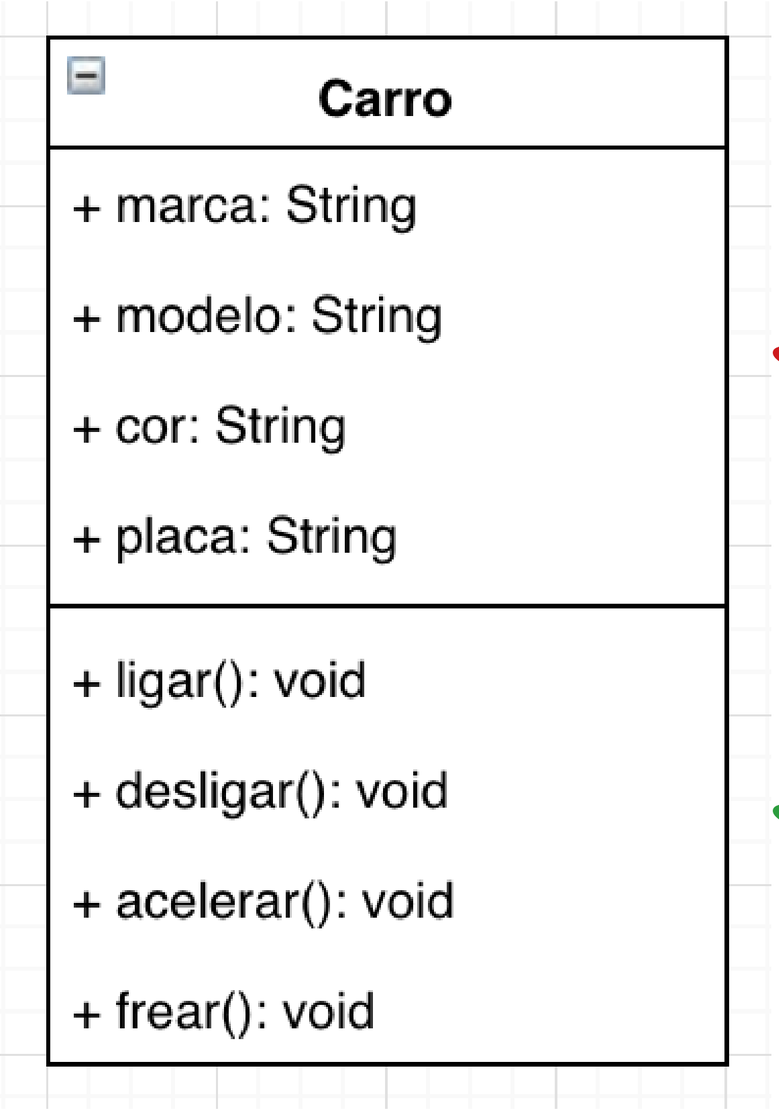
Classe

INSTITUTO FEDERAL
Sergipe

Abstração das características de um grupo de coisas do mundo real. É uma estrutura que encapsula dados (chamados de **atributos**) e um conjunto de operações possíveis de serem usados, chamados **métodos**.

Classe

INSTITUTO FEDERAL
Sergipe



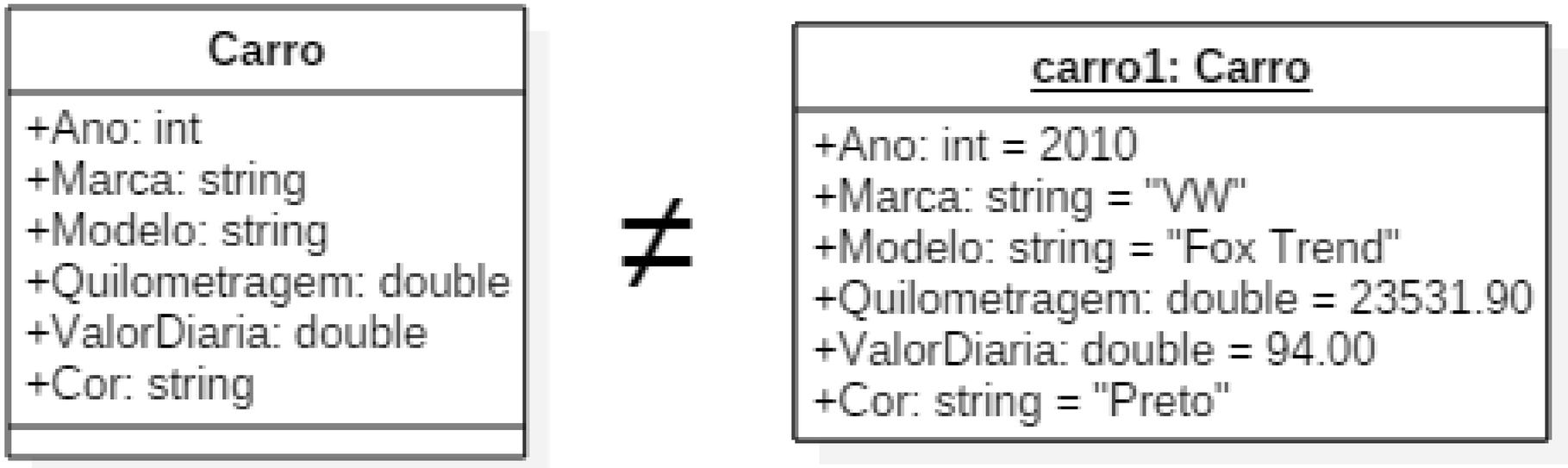
Atributos

Métodos

Objeto

INSTITUTO FEDERAL
Sergipe

Objeto: um elemento específico de uma classe ou instância de uma classe.



Linguagem Orientada a Objetos

INSTITUTO FEDERAL
Sergipe

Classe em Java

INSTITUTO FEDERAL
Sergipe

```
3   public class Pessoa {
4
5       String nome;
6       int idade;
7       String endereco;
8
9   }
```

Pessoa.java

Objeto em Java

```java
public class TestePessoa {
    public static void main(String[] args) {
        Pessoa pessoa = new Pessoa();
        pessoa.endereco = "Rua A, Cond. Jardins das Flores, 330";
        pessoa.nome = "José Santos";
        pessoa.idade = 34;

        System.out.println("Ben-vindo: " + pessoa.nome);
    }
}
```

TestePessoa.java

Pilares da Orientação a Objetos

INSTITUTO FEDERAL
Sergipe

Abstração

INSTITUTO FEDERAL
Sergipe

Envolve a simplificação e a generalização para lidar com complexidade, permitindo que se foque nos **aspectos essenciais** de um problema ou sistema enquanto se ignora os detalhes menos relevantes.

Abstração

INSTITUTO FEDERAL
Sergipe

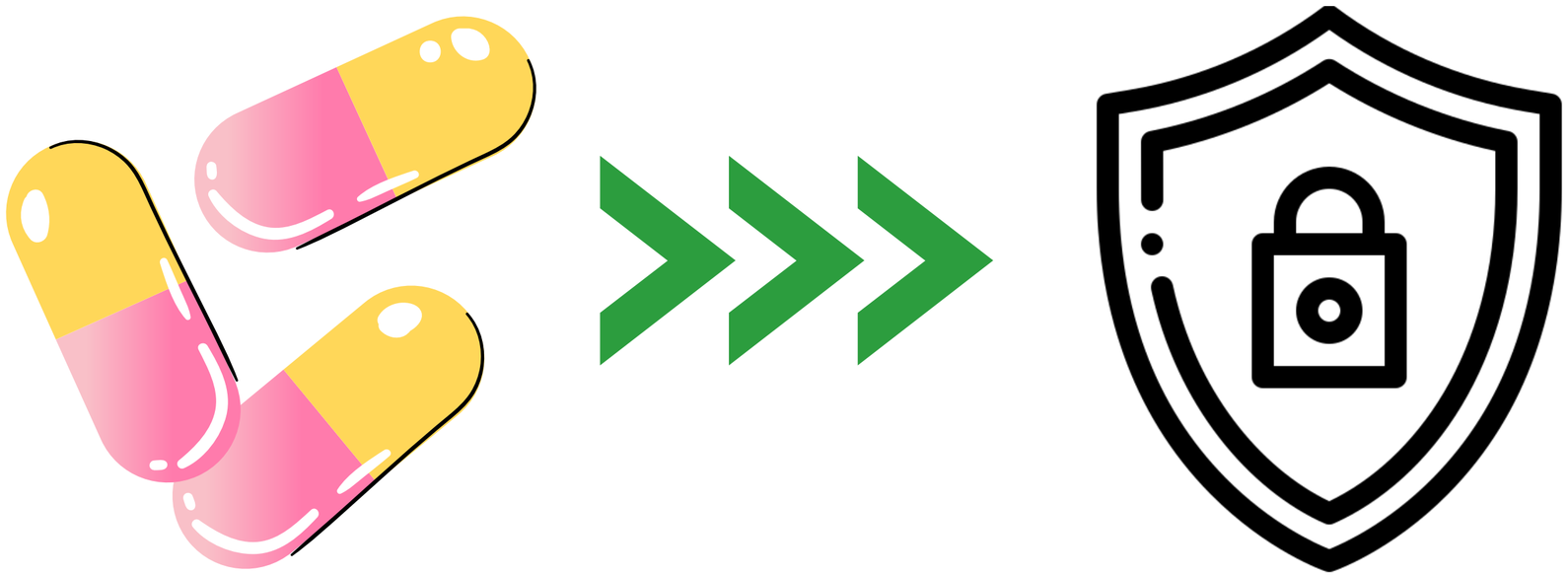
Abstração

```java
package pilares;

public class Pessoa {
    String nome;
    int idade;
    String endereco;

    public String escreverNaTela() {
        return "Pessoa{" +
                "nome='" + nome + '\'' +
                ", idade=" + idade +
                ", endereco='" + endereco + '\'' +
                '}';
    }
}
```

Encapsulamento

INSTITUTO FEDERAL
Sergipe

Esconder os detalhes internos de uma classe e expor apenas o que é necessário para o exterior.



Encapsulamento

INSTITUTO FEDERAL

```
public class Pessoa {
    private String nome;
    private int idade;
    private String endereco;

    public String getNome() { return nome; }
    public void setNome(String nome) { this.nome = nome; }
    public int getIdade() { return idade; }
    public void setIdade(int idade) { this.idade = idade; }
    public String getEndereco() { return endereco; }
    public void setEndereco(String endereco) { this.endereco = endereco; }
    public void escreverNaTela() {...}
}
```

Sem encapsulamento

```java
public class TestePessoa {
    public static void main(String[] args) {
        Pessoa pessoa = new Pessoa();
        pessoa.endereco = "Rua A, Cond. Jardins das Flores, 330";
        pessoa.nome = "José Santos";
        pessoa.idade = 34;

        System.out.println("Ben-vindo: " + pessoa.nome);
    }
}
```

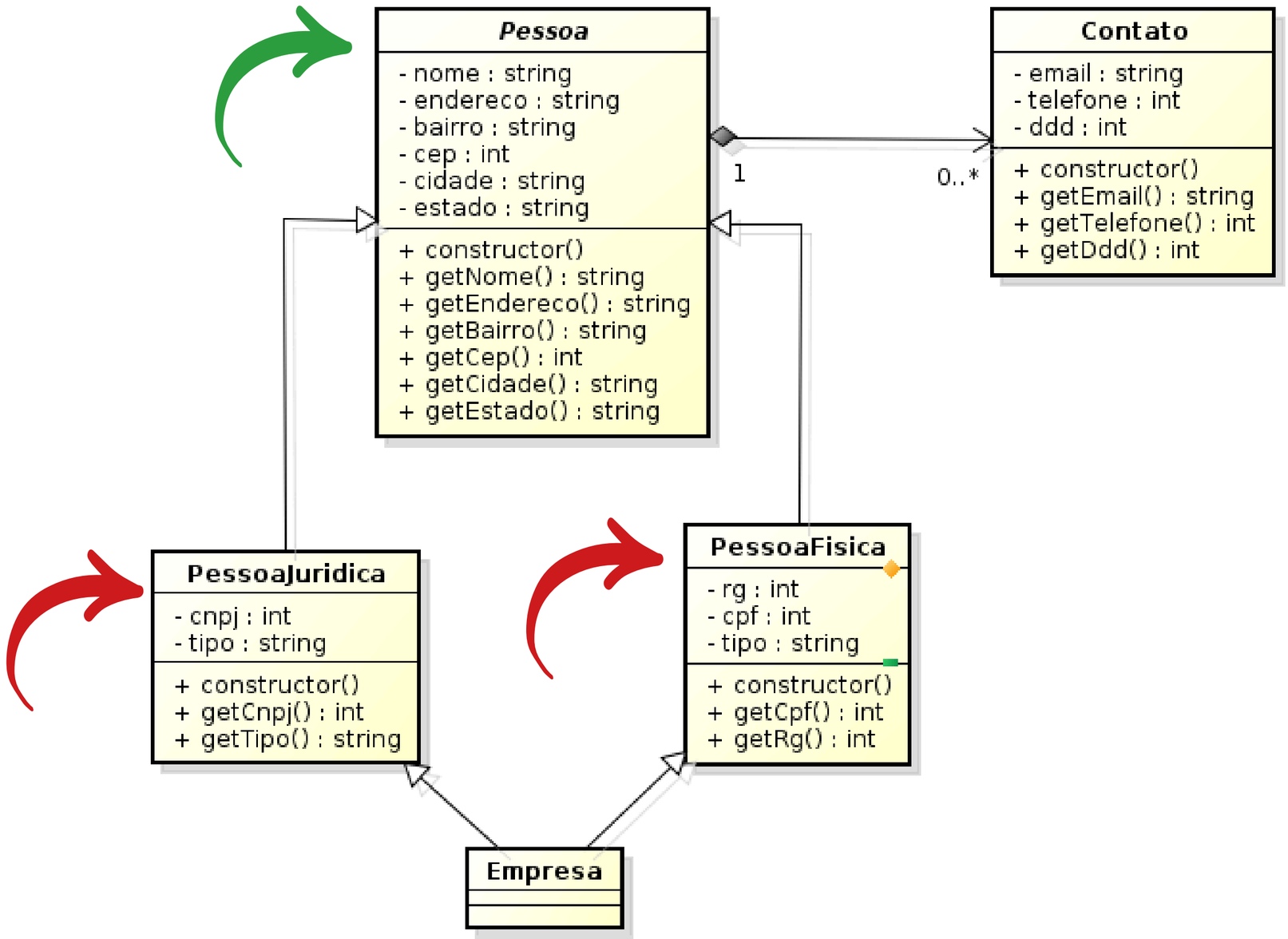
TestePessoa.java

Herança

INSTITUTO FEDERAL
Sergipe

A herança permite que uma nova classe (subclasse ou classe derivada) **herde** atributos e métodos de uma classe existente (superclasse ou classe base).

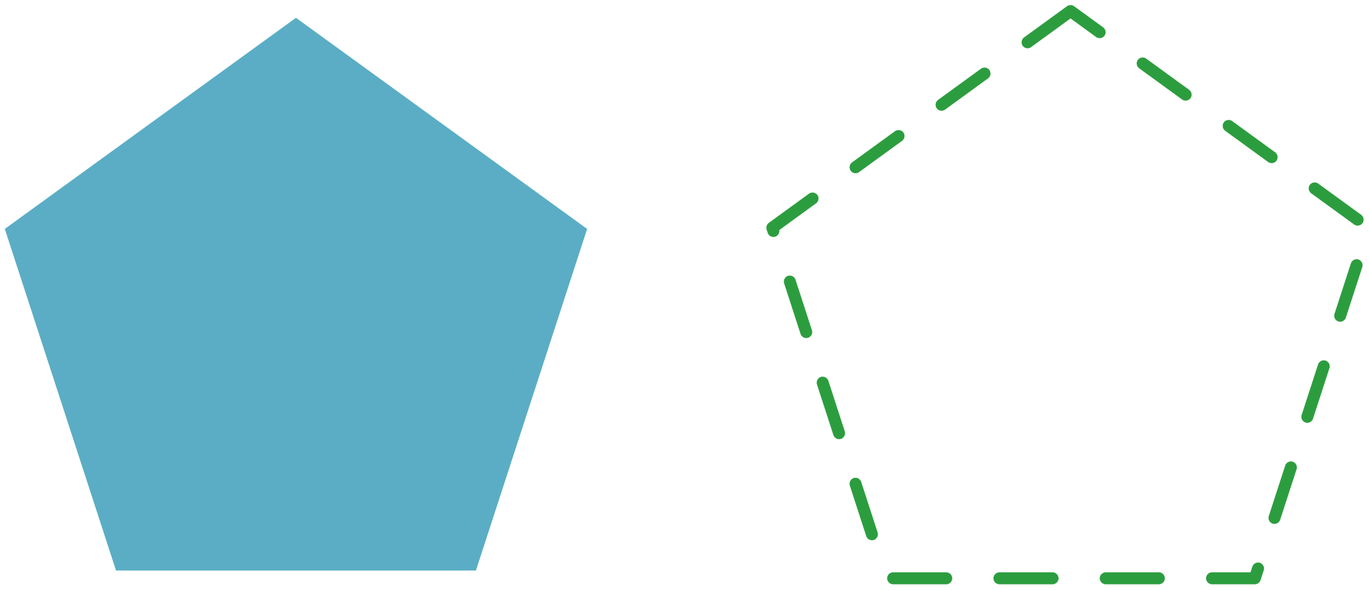
Herança


Pessoa
- nome : string
- endereco : string
- bairro : string
- cep : int
- cidade : string
- estado : string
+ constructor()
+ getNome() : string
+ getEndereco() : string
+ getBairro() : string
+ getCep() : int
+ getCidade() : string
+ getEstado() : string
Contato
- email : string
- telefone : int
- ddd : int
+ constructor()
+ getEmail() : string
+ getTelefone() : int
+ getDdd() : int
1
0..*
PessoaJuridica
- cnpj : int
- tipo : string
+ constructor()
+ getCnpj() : int
+ getTipo() : string
PessoaFisica
- rg : int
- cpf : int
- tipo : string
+ constructor()
+ getCpf() : int
+ getRg() : int
Empresa

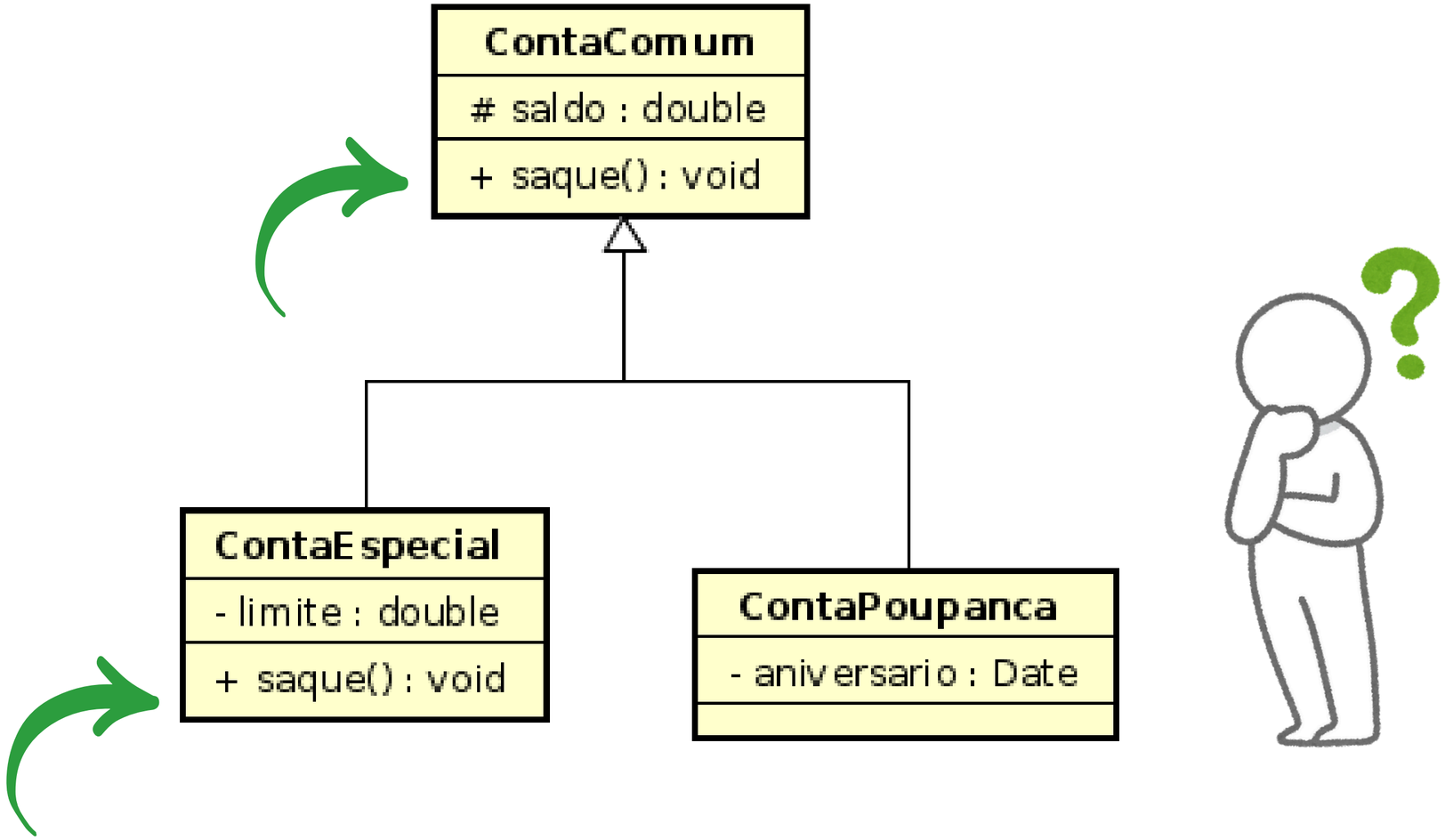
Polimorfismo

INSTITUTO FEDERAL
Sergipe

Permite que métodos com o mesmo nome se comportem de maneira diferente dependendo do tipo do objeto que os chama.



Polimorfismo



Herança e Polimorfismo

INSTITUTO FEDERAL
Sergipe

```java
public class Pessoa {
    protected String nome;
    protected int idade;
    protected String endereco;
    protected float rendaAnual;

    1 override    new *
    public double calcularImposto() {
        return 0.0;
    }

    Fausto Soares
    public String getNome() { return nome; }

    Fausto Soares
    public void setNome(String nome) { this.nome = nome; }
```

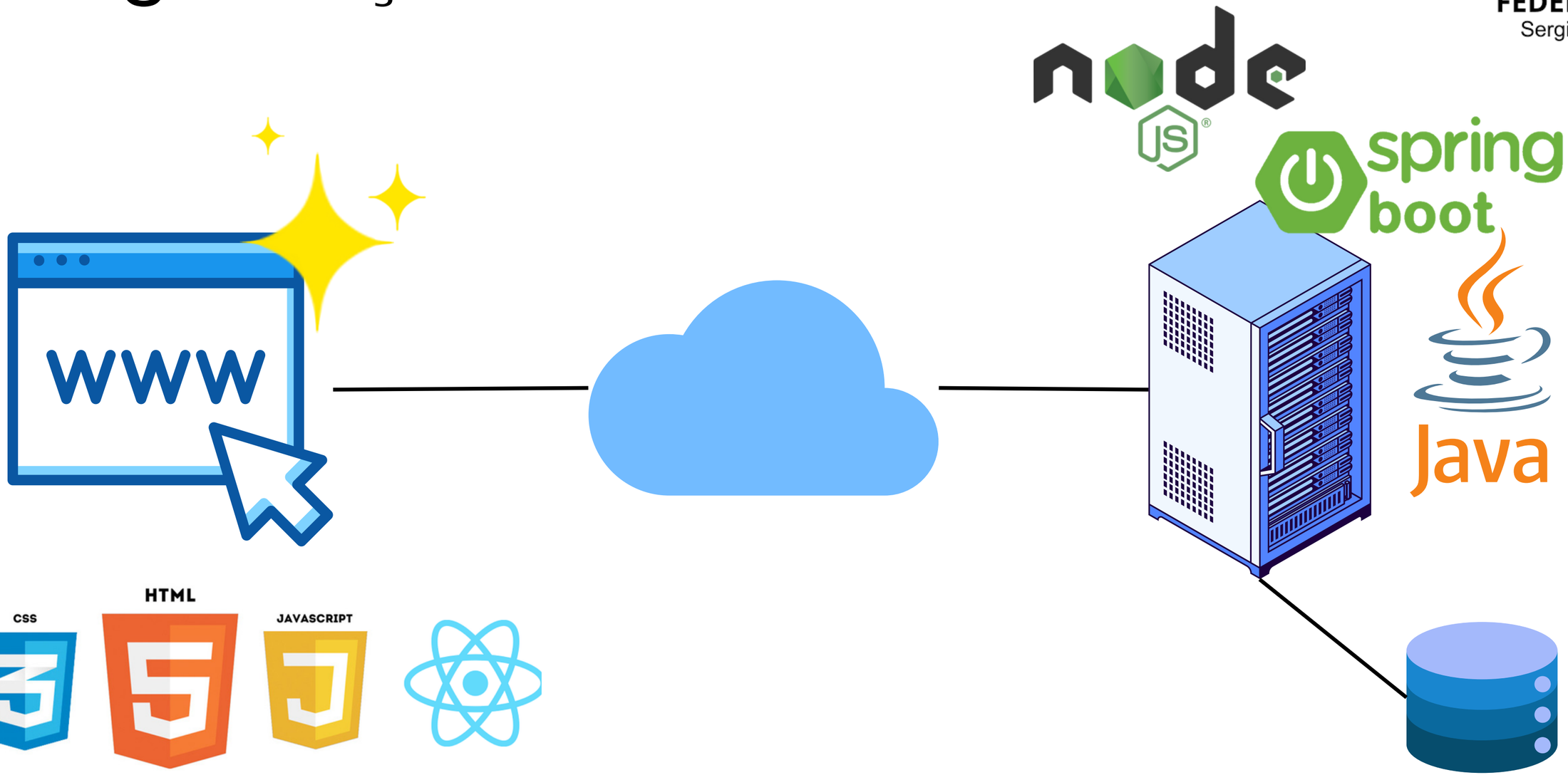
```java
public class PessoaFisica extends Pessoa{
    private String cfp;

    new *
    public String getCfp() { return cfp; }

    new *
    public void setCfp(String cfp) { this.cfp = cfp; }

    new *
    @Override
    public double calcularImposto() {
        return rendaAnual * 0.15;
    }
}
```

INSTITUTO FEDERAL
Sergipe

E agora? Qual próximo passo?

Programação Web

INSTITUTO FEDERAL
Sergipe



Programação Mobile

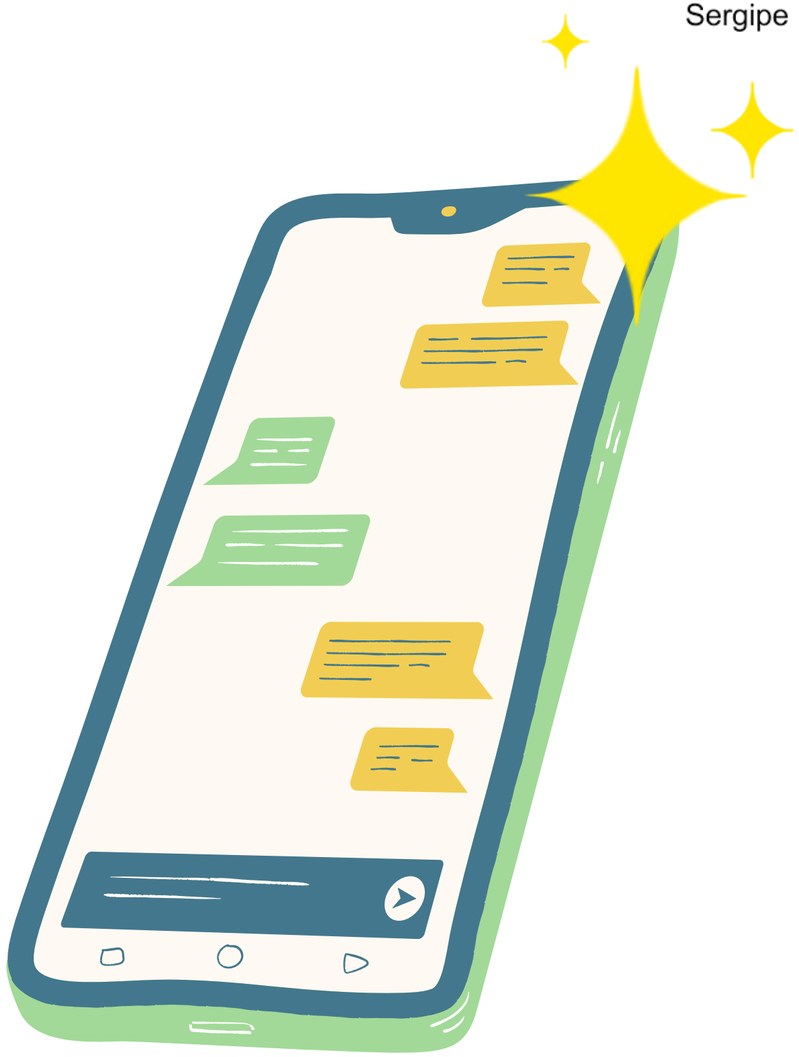
INSTITUTO FEDERAL
Sergipe

Swift

ANDROID

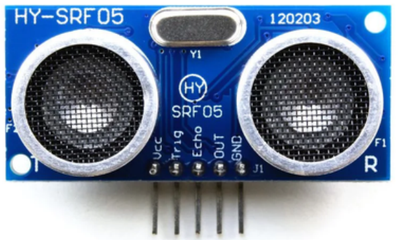
act Native

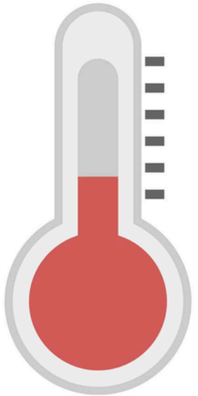
Kotlin

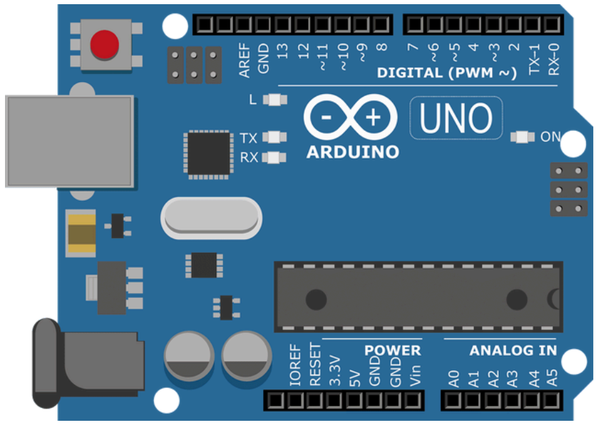


Programação IoT

INSTITUTO FEDERAL
Sergipe







python™

Exercício

INSTITUTO FEDERAL
Sergipe

Utilizando Java, desenvolva uma classe **Conta** que simule uma conta bancária, com a capacidade de realizar transferências entre contas.

Requisitos:

1. Criar Classe Conta, com os seguintes atributos:
    - numero: Um número inteiro que representa o número da conta.
    - titular: Uma string que representa o nome do titular da conta.
    - saldo: Um valor decimal que representa o saldo atual da conta.
    - agencia: Uma string que representa o número da agência onde a conta está registrada.
2. Implementar método **saque** e **transferência** (entre duas contas)
3. Criar uma classe Principal para criar duas contas e realizar um saque na conta 1 e uma transferência de 100 da conta 1 para a conta 2.

Material (Slide + Código)

INSTITUTO FEDERAL
Sergipe

https://github.com/faustosoares/aula_programacao_IFS.git

GitHub

Referências

DEITEL, P.J. **Java Como Programar**. Rio de Janeiro: Pearson, 2005.

FURGERI, Sérgio. **Java 8 Ensino Didático: Desenvolvimento e Implementação deAplicações**. 1ª Ed. São Paulo: Érica, 2015.

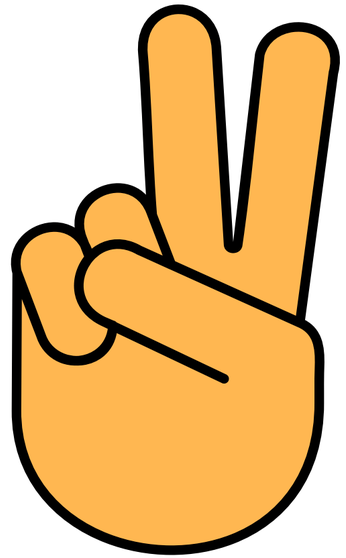
GAMMA, E., HELM, R., JOHNSON, R., VLISSIDES, J. **Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos.** Porto Alegre: Bookman, 2005.

LUTZ, Mark. **Programming Python**. 4ª edição. O'Reilly Media, 2019.

SEBESTA, Robert W. **Concepts of Programming Languages**. 11ª edição. Boston: Pearson, 2016.

INSTITUTO FEDERAL
Sergipe

Fim



INSTITUTO FEDERAL
Sergipe

Dúvidas?