



POO

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

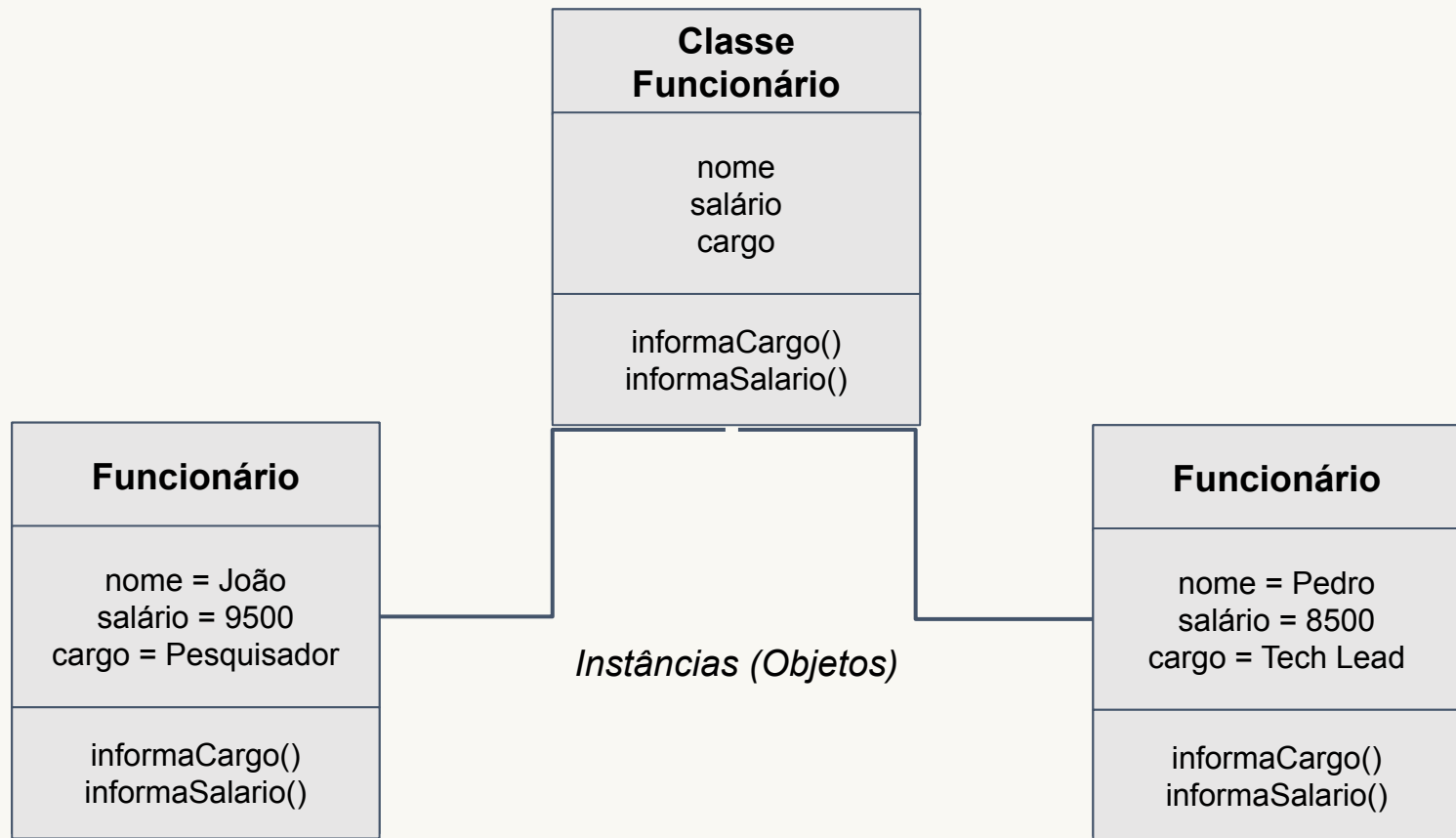
Definição POO

Programação Orientada a Objetos (POO ou OOP, em inglês) é um paradigma da programação baseado no conceito de objetos que podem conter dados na forma de campos, conhecidos como **atributos**, e códigos, na forma de procedimentos também conhecidos como **métodos**. Na POO, todo objeto possui as seguintes características:

- Estado: conjunto de propriedades de um objeto (valores dos atributos)
- Comportamento: conjunto de ações possíveis sobre o objeto (métodos da classe)
- Unicidade: todo objeto é único (possui um endereço na memória)

É um componente de programa que descreve a “estrutura” e o “comportamento” de um grupo de objetos semelhantes, ou seja, as informações que caracterizam o estado desses objetos e as ações (ou operações) que eles podem realizar.

Exemplo de Classe



```
1  # Definindo uma classe carro
2
3  class Carro:
4      def __init__(self, marca = "", modelo = "", ano = ""):
5          self.marca = marca
6          self.modelo = modelo
7          self.ano = ano
8
```

Classes (Conceitos)

classe : Define uma estrutura de dados para o modelo, com métodos, propriedades, etc.

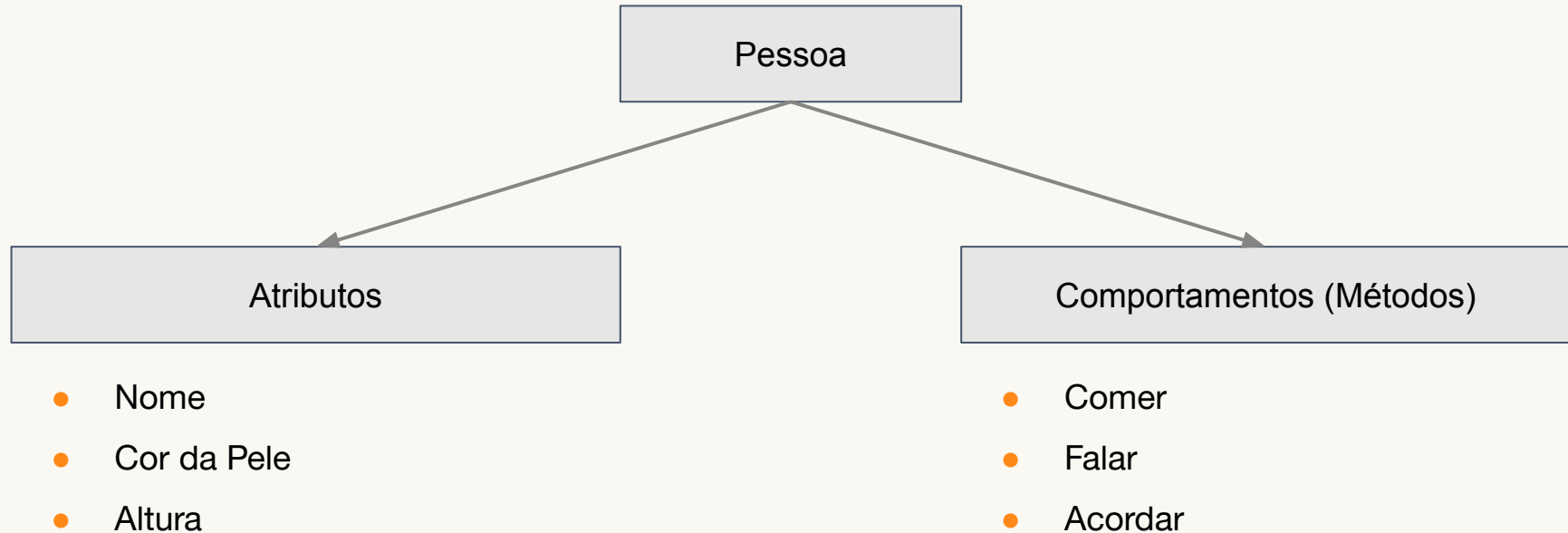
instância : uma representação da classe que foi inicializada.

`__init__` : “Construtor” da classe, é o método chamado da inicialização de uma nova instância

`self` : Pode ser compreendido como a referência da própria instância que está sendo inicializada

Um objeto é uma instância de uma classe. Quando criamos um um objeto (POO), criamos fisicamente uma representação concreta de sua classe. Exemplo: “Carro” é uma classe, um modelo de dados, e “VW Nivus” é uma instância de “Carro”, o objeto vai possuir todas as características do modelo “Carro” mas completamente diferente e independente das outras instâncias que forem criadas.

Exemplo de Objeto



Objeto

```
1  # Definindo uma classe carro
2
3  class Carro:
4      def __init__(self, marca = "", modelo = "", ano = ""):
5          self.marca = marca
6          self.modelo = modelo
7          self.ano = ano
8
9  # Criando uma instância de Carro (Objeto)
10 vw_nivus = Carro('VW', 'Nivus', 2020)
11
```

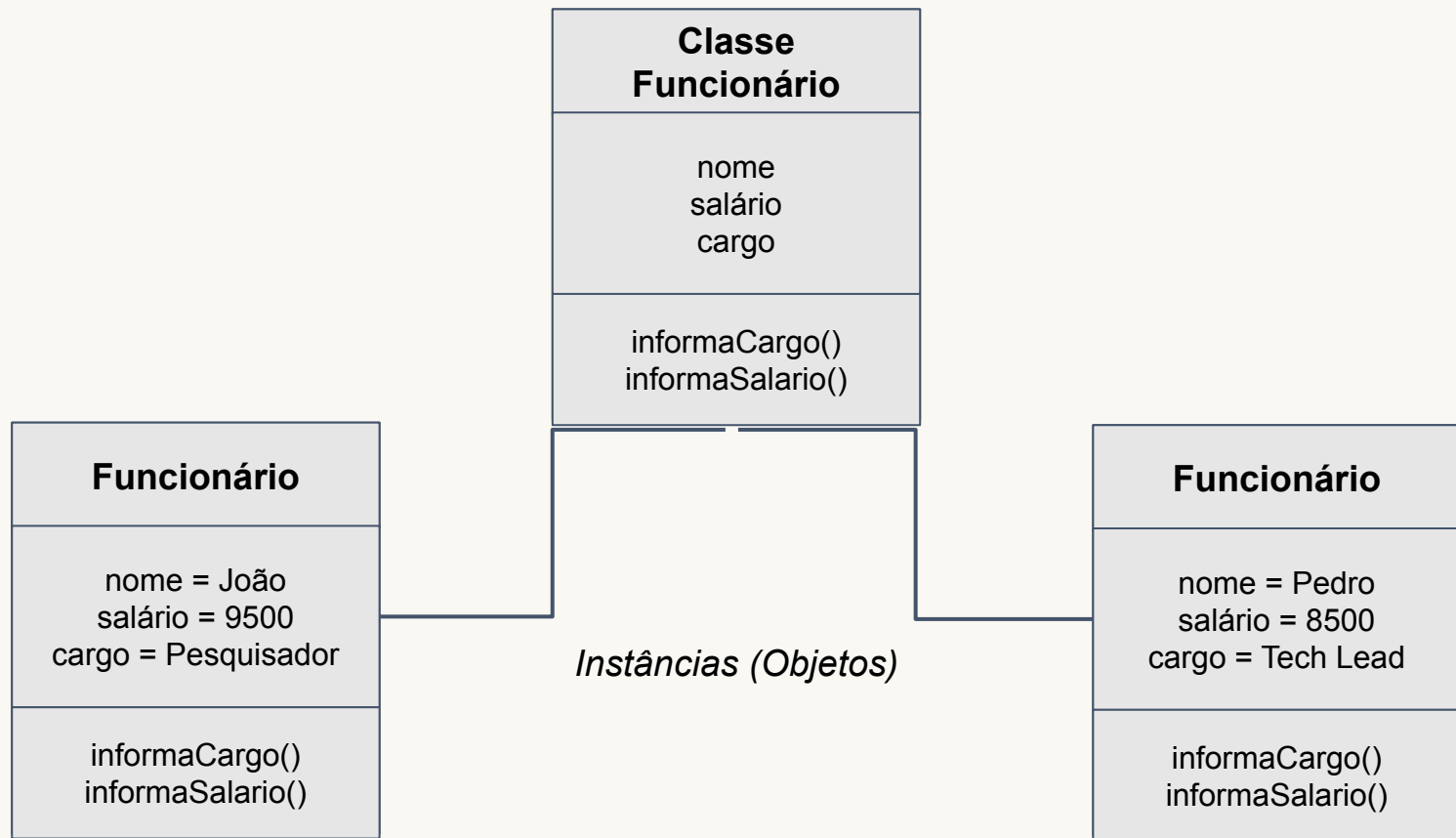
Exercício 1 (30min)

Construir uma aplicação para informatizar o sistema de uma Pizzaria. Esse sistema precisa possuir os seguintes objetos com seus atributos:

- Garçom (nome)
- Pizza (tamanho, preço, sabor)
- Cliente (nome, mesa, valor gasto)
- Pedido (código, mesa, valor, nome cliente, nome garçom)

Realizar a construção das classes e testar criando objetos pré-definidos.

Exemplo de Classe



Herança

DEVinHouse

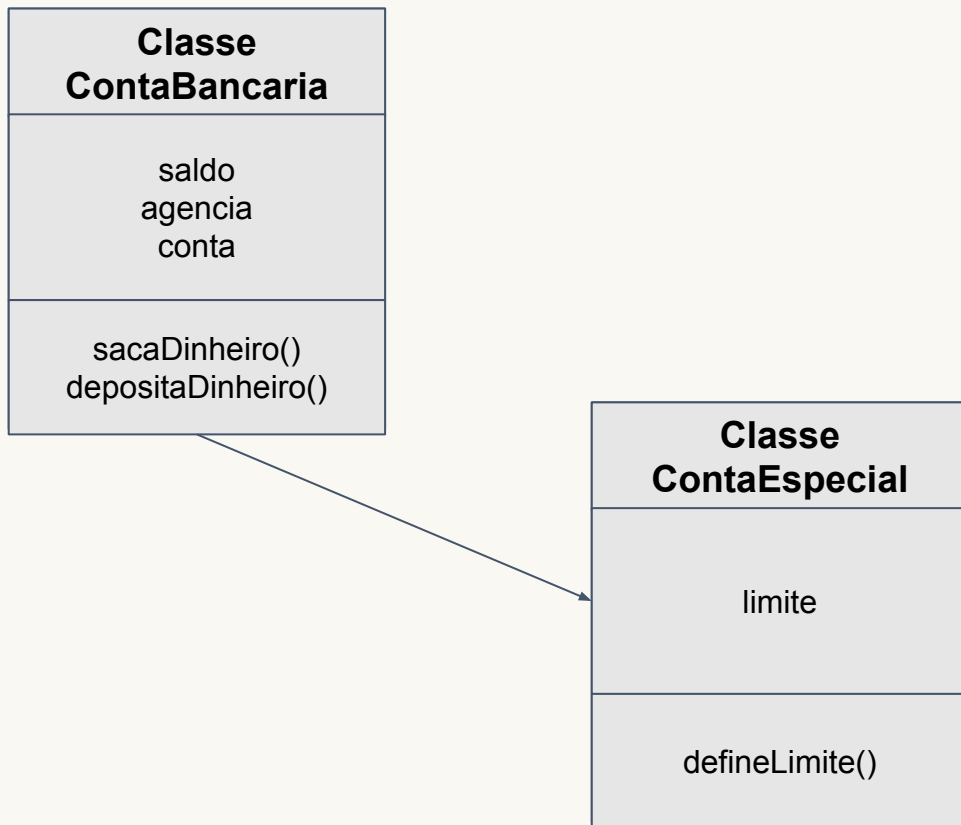
Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

É um conceito da programação orientada a objetos que determina que uma classe (filha) pode herdar atributos e métodos de uma outra classe (pai), e assim, evitar que haja repetição de código.

Exemplo de Herança



Os objetos inicializados com `ContaEspecial` vão conter as mesmas implementações das contas bancárias “simples” mas com a implementação de limite previsto na especial.

É um conceito da programação orientada a objetos que determina que uma classe (filha) pode herdar atributos e métodos de uma outra classe (pai), e assim, evitar que haja repetição de código. Este conceito pode ser utilizado por exemplo para uma classe “pai” com o nome funcionário e existirem outras “sub classes” com os nomes dos cargos, a classe pai seria construída para métodos e atributos “globais”, enquanto as sub classes seriam utilizadas para métodos e atributos referentes aos cargos.

Herança

```
1  class Animal():
2      def __init__(self, nome, cor):
3          self.nome = nome
4          self.cor = cor
5
6      def comer(self):
7          print(f"O {self.nome} está comendo")
8
9  class Cachorro(Animal):
10     def __init__(self, nome, cor):
11         super().__init__(nome, cor)
12
13  cachorro = Cachorro('Buzzy', 'Preto')
```


Herança

```
1 class Animal():
2     def __init__(self, nome, cor):
3         self.nome = nome
4         self.cor = cor
5
6     def comer(self):
7         print(f"O {self.nome} está comendo")
8
9 class Cachorro(Animal):
10     def __init__(self, nome, cor):
11         super().__init__(nome, cor)
12
13 cachorro = Cachorro('Buzzy', 'Preto')
```

São declaradas duas classes, com a “Animal” sendo a classe pai e “Cachorro” o filho. Enviamos para a classe Cachorro, a classe base para receber os métodos declarados.

Herança

```
1  class Animal():
2      def __init__(self, nome, cor):
3          self.nome = nome
4          self.cor = cor
5
6      def comer(self):
7          print(f"O {self.nome} está comendo")
8
9  class Cachorro(Animal):
10     def init (self, nome, cor):
11         super().__init__(nome, cor)
12
13  cachorro = Cachorro('Buzzy', 'Preto')
```

O `super()` permite que a classe filha acesse explicitamente algum método presente na classe pai.

Para este exemplo é como fizéssemos:

`this.Animal().__init__(nome, cor)`

Exercício 2 (40min)

Construir uma aplicação que contenha uma classe pai chamada funcionário e 3 classes filhas, desenvolvedor, designer, gerente de marketing e product manager. Implemente na classe pai (Funcionários), os seguintes métodos: `setarSalario`, `aumentarSalario` e `demitir`. E para a classe do **gerente de marketing**, adicione o método “`bonificacao`” que aumenta o salário do funcionário de acordo com a porcentagem passada por argumento.

Inicialize o sistema com 1 funcionário para cada grupo presente e defina como atributos seus nomes e salários.

Material Complementar

- Introdução a POO no Python -

<https://pythonacademy.com.br/blog/introducao-a-programacao-orientada-a-objetos-no-python>

- Utilizando Herança no Python - <https://www.treinaweb.com.br/blog/utilizando-heranca-no-python>

- Herança em Python - <https://algoritmoempython.com.br/cursos/programacao-python/heranca/>

- Uso do “super” em classe no Python -

<https://pt.stackoverflow.com/questions/22452/como-se-usa-e-para-que-serve-o-super-em-classe-s-python>



DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>

AGENDA

- Informações do template
- Exemplo de texto (Light mode)
- Exemplo de texto (Dark mode)
- Exemplo de tópicos (Light mode)
- Exemplo de tópicos (Dark mode)
- Exemplo com imagens (Light mode)
- Exemplo com imagens (Dark mode)
- Exemplo com tabelas (Light mode)
- Exemplo com tabelas (Dark mode)

INFORMAÇÕES DO TEMPLATE

- Título da Apresentação:
 - Fonte: Ubuntu Bold
 - Formato: Maiúsculo
 - Tamanho: 34
 - Cor: Branco
- Título do Slide:
 - Fonte: Ubuntu Bold
 - Formato: Maiúsculo
 - Tamanho: 22
 - Cor: Branco
- Parágrafos:
 - Fonte: Open Sans Normal
 - Tamanho: 14 a 18
 - Cores: Branco (Dark Mode) ou Preto (Light Mode)
- Marcadores de tópicos:
 - Formatos: Símbolos ou Alfanuméricos
 - Cor: Laranja
- Padrão de Cores:
 - Cinza - #868584
 - Preto - #1C1C19
 - Branco - #FAFAFA
 - Laranja - #F08305
 - Rosa - #c71d81
 - Azul - #0e1d8e

EXEMPLO DE TEXTO (LIGHT MODE)

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

EXEMPLO DE TEXTO (DARK MODE)

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

EXEMPLO DE TÓPICOS (LIGHT MODE)

- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;
- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;
- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;

EXEMPLO DE TÓPICOS (DARK MODE)

- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;
- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;
- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;

EXEMPLO COM IMAGENS (LIGHT MODE)

Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.



Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

EXEMPLO COM IMAGENS (DARK MODE)

Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.



Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

EXEMPLO COM TABELAS (LIGHT MODE)

Lorem	Ipsum	Ipsum	Ipsum	Ipsum	Ipsum
	1	2	3	4	5
xxx	xxx	xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx	xxx	xxx

EXEMPLO COM TABELAS (DARK MODE)

Lorem	Ipsum	Ipsum	Ipsum	Ipsum	Ipsum
	1	2	3	4	5
xxx	xxx	xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx	xxx	xxx



DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>