

# **Paradigma Orientado a Objetos**

**Cauã Ribas Devitte**

Universidade do Vale do Itajaí  
Escola Politécnica  
Ciência da Computação  
cauaribas@edu.univali.br

## **Paradigmas de Programação**

Pesquisa M1 - Paradigma Orientado a Objetos

03/11/23

**Resumo:** *Este paradigma estrutura o código em torno de objetos, que são instâncias de classes, promovendo a encapsulação, herança e polimorfismo. Originado na década de 1960, o POO evoluiu para se tornar um dos paradigmas mais amplamente adotados, sendo usado em uma ampla gama de aplicações, incluindo sistemas complexos, aplicativos de desktop, web, jogos e móveis. Algumas linguagens específicas para o paradigma orientado a objetos incluem Java, C++ e C#. No entanto, muitas linguagens multiparadigma, como Python e JavaScript, também suportam o POO.*

## **1. O nome do paradigma:**

Paradigma orientado a objetos (POO).

## **2. Sua Origem:**

A programação orientada a objetos surgiu como uma alternativa a programação estruturada. O intuito da sua criação foi de aproximar o manuseio das estruturas de um programa ao manuseio das coisas do mundo real, daí o nome "objeto" como algo genérico, que pode representar qualquer coisa tangível. O conceito fundamental da POO começou a surgir com linguagens como Simula 67 e Smalltalk. Essas linguagens introduziram a ideia de encapsulação, classes e objetos como componentes centrais da programação.

## **3. Evolução e Uso:**

Ao longo do tempo, a POO passou por diversas mudanças e aprimoramentos, impulsionados pela necessidade de lidar com sistemas de software cada vez mais complexos. A noção de classes como modelos para criar objetos foi uma evolução crucial. Isso permitiu a reutilização de código e uma maior abstração na programação. A introdução de conceitos como herança e polimorfismo aprimorou a flexibilidade e a organização de código orientado a objetos. A popularização de linguagens de programação orientadas a objetos desempenhou um papel significativo na evolução do paradigma. Exemplos incluem C++, que combinou programação orientada a objetos com programação procedural, e o desenvolvimento do Java, que se tornou amplamente adotado para desenvolvimento empresarial e web. O paradigma é amplamente utilizado na programação de sistemas complexos, como aplicativos de desktop, sistemas web, jogos, aplicativos móveis e muitos outros. Linguagens populares que suportam o POO incluem Java, C++, C#, Python, Ruby e PHP.

## **4. Linguagens específicas Orientado a Objetos:**

Algumas linguagens de programação são específicas para o paradigma orientado a objetos. Exemplos incluem:

### **4.1 Java:**

Java é a mais popular linguagem de programação orientada a objetos. Ela foi desenvolvida pela Sun Microsystems e hoje é mantida pela Oracle. O Java possui uma vasta gama de ferramentas e recursos para o desenvolvimento de sistemas orientados a objetos. Desde a incorporação da tecnologia Java pela Oracle, ela tem ganhado cada vez mais bibliotecas e tem sofrido atualizações mais frequentes, incorporando tecnologias recentes e inovadoras.

### **4.2 C#:**

Foi desenvolvido pela Microsoft como parte da plataforma .NET. Ela foi criada com forte influência de outras linguagens de programação como Object Pascal e, especialmente, o Java.

### **4.3 C++:**

O C++ foi desenvolvido a partir da linguagem C, uma linguagem estruturada e procedural. O C++ foi criado justamente para adaptar o C, que era bastante popular à época para esse paradigma de programação que ganhava força. O C++ se destaca pelo desempenho, uma vez que possibilita o uso de recursos de mais baixo nível, como ponteiros. Uma diferença interessante entre C++ e Java no assunto específico da orientação a objetos é que C++ permite a implementação direta do conceito de herança múltipla, isto é, um objeto herdar de dois ou mais objetos. Em Java, isto também é possível, porém de forma menos direta, por meio de interfaces.

## **5. Linguagens multi-paradigma Orientado a Objetos:**

Muitas linguagens de programação são multiparadigma e permitem a programação orientada a objetos. Alguns exemplos incluem:

### **5.1 Python:**

Python já nasceu sendo uma linguagem de programação multi-paradigma, isto é, é possível programar em Python de maneira Imperativa, Funcional e também no paradigma Orientado a Objetos, utilizando conceitos da Programação Orientada a Objetos.

Embora seja principalmente uma linguagem de programação imperativa, Python suporta programação orientada a objetos. A linguagem de programação Python vem

crescendo muito nos últimos anos devido à sua simplicidade e, principalmente, à sua grande compatibilidade, pois funciona bem na maioria dos sistemas operacionais. Projetos desenvolvidos utilizando a orientação a objetos, além de mais estáveis, são de fácil manutenção e sua reutilização é mais simples.

## **5.2 JavaScript:**

JavaScript é uma linguagem baseada em protótipos. Protótipos são o mecanismo pelo qual os objetos JavaScript herdam recursos uns dos outros e funcionam de maneira diferente dos mecanismos de herança em linguagens de programação clássicas orientadas a objetos.

Em JavaScript, a maioria das coisas são objetos, desde os principais recursos do JavaScript, como arrays, até o navegador APIs construído sobre o JavaScript. Você pode até mesmo criar seus próprios objetos para encapsular funções e variáveis relacionadas em pacotes eficientes e atuar como contêineres de dados úteis.

## **5.3 Ruby:**

Ruby é uma linguagem de programação dinâmica e orientada a objetos. Ela foi criada pelo programador japonês Yukihiro Matsumoto, conhecido como "Matz", e lançada oficialmente em 1995. Ruby foi projetada com o objetivo de combinar a simplicidade e a facilidade de leitura com uma sintaxe elegante.

Ruby é uma linguagem interpretada, o que significa que o código-fonte é executado diretamente por um interpretador, sem a necessidade de compilação prévia. Ela é conhecida por sua produtividade e expressividade, permitindo aos programadores escreverem código de forma concisa e clara.

Uma das principais características do Ruby é sua orientação a objetos. Tudo em Ruby é um objeto, incluindo números, strings e até mesmo classes e módulos. Isso permite que os desenvolvedores usem conceitos de programação orientada a objetos de forma consistente em seus projetos.

Em resumo, Ruby é uma linguagem de programação orientada a objetos que se destaca por sua elegância, expressividade e produtividade. É amplamente utilizado no desenvolvimento web e em várias outras áreas, oferecendo uma ótima experiência de desenvolvimento para os programadores.

## 6. Um trecho de código que exemplifica o uso desse paradigma:

Exemplo de Código em Java:

```
1 public class Carro {
2     Double velocidade;
3     String modelo;
4
5     public Carro(String modelo) {
6         this.modelo = modelo;
7         this.velocidade = 0.0;
8     }
9
10    public void acelerar() {
11        /* código do carro para acelerar */
12    }
13
14    public void frear() {
15        /* código do carro para frear */
16    }
17
18    public void acenderFarol() {
19        /* código do carro para acender o farol */
20    }
21 }
```

## 7. Comparativo entre este paradigma e o orientado a objetos:

O paradigma orientado a objetos e o paradigma estruturado diferem em sua abordagem de programação.

### Paradigma Orientado a Objetos:

- Estrutura o código em torno de objetos, que têm estado (atributos) e comportamento (métodos).
- Promove a encapsulação, onde os detalhes internos de um objeto são ocultos.
- Suporta herança, permitindo a criação de novas classes com base em classes existentes.
- Possibilita o polimorfismo, onde objetos de diferentes classes podem responder de maneira semelhante a mensagens.
- É adequado para modelar entidades do mundo real.

### Paradigma Estruturado:

- Estrutura o código em torno de funções e procedimentos.
- Estrutura o código em torno de funções e procedimentos.
- Não enfatiza a encapsulação, e os dados podem ser globais ou compartilhados entre funções.
- Não suporta herança ou polimorfismo de maneira direta.
- É adequado para tarefas mais simples e algoritmos lineares.

## 8. Conclusão

Em síntese, o paradigma orientado a objetos é uma abordagem de programação que evoluiu ao longo das décadas, tornando-se uma base sólida para o desenvolvimento de software em uma ampla variedade de domínios.

A evolução do paradigma orientado a objetos testemunhou o desenvolvimento de conceitos fundamentais, como classes, objetos, herança, polimorfismo e encapsulação. A popularização de linguagens de programação orientadas a objetos, como Java, C++, C#, Python e Ruby, tornou o POO amplamente adotado em diferentes setores, desde aplicações empresariais até jogos e desenvolvimento web.

À medida que o desenvolvimento de software continua a avançar, o paradigma orientado a objetos continua a se adaptar, incorporando conceitos de outras abordagens, como a programação orientada a objetos funcional. Em resumo, o POO continua sendo uma pedra angular do desenvolvimento de software e uma ferramenta poderosa para a criação de sistemas complexos e eficazes.

## Referências

- Alura, “POO: o que é programação orientada a objetos?”,  
<https://www.alura.com.br/artigos/poo-programacao-orientada-a-objetos>
- Escola Virtual, “Desenvolvimento Orientado a Objetos Utilizando a Linguagem Python ”,  
<https://www.ev.org.br/cursos/desenvolvimento-orientado-a-objetos-utilizando-a-linguagem-python>
- Python Academy, “Programação Orientada a Objetos no Python: Introdução”,  
<https://pythonacademy.com.br/blog/introducao-a-programacao-orientada-a-objetos-no-python>
- FreeCodeCamp, “Programação orientada a objetos em JavaScript”,  
<https://www.freecodecamp.org/portuguese/news/programacao-orientada-a-objetos-em-javascript-explicada-com-exemplos/>
- DeveloperMozilla, “Introdução a objetos em JavaScript”,  
<https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Objects>
- Dio, “Ruby: Uma linguagem eficiente e produtiva”,  
<https://www.dio.me/articles/ruby-conheca-essa-produtiva-e-eficiente-linguagem-de-programacao>