

Paradigma Lógico

Eduardo Moretto, Mateus Barbosa, Matheus de Oliveira Rocha

Ciências da Computação – Universidade do Vale do Itajaí (Univali)
Itajaí, SC – Brasil

{eduardo.moretto, mateus.babosa, matheus.rocha}@edu.univali.br

Abstract. *The Logical programming paradigm is an approach that aims to model problems and systems based on logical relationships and inferences. It was developed from research in artificial intelligence and has been widely used in areas such as natural language processing, expert systems, and inference systems. In this report, we explore the main concepts of the Logical programming paradigm, its origin, evolution, and use, as well as some specific languages and code examples that illustrate its application. Furthermore, we compare the Logical paradigm with the object-oriented paradigm, highlighting the main differences between them.*

Resumo. *O paradigma de programação Lógico é uma abordagem que tem como objetivo modelar problemas e sistemas baseados em relações e inferências lógicas. Ele foi desenvolvido a partir da pesquisa em inteligência artificial e tem sido amplamente utilizado em áreas como processamento de linguagem natural, sistemas especialistas e sistemas de inferência. Neste relatório, vamos explorar os principais conceitos do paradigma de programação Lógico, sua origem, evolução e uso, bem como algumas linguagens específicas e exemplos de código que ilustram sua aplicação. Além disso, vamos comparar o paradigma Lógico com o paradigma orientado a objetos, destacando as principais diferenças entre eles.*

1. O que são os Paradigmas de Programação?

Primeiramente, o termo "paradigma" é usado em diversas áreas científicas, como por exemplo na filosofia usa-se para descrever um conjunto de crenças, valores e métodos compartilhados por uma comunidade de pensadores ou cientistas. É uma espécie de "modelo mental" que orienta a maneira como essas pessoas pensam sobre o mundo, formulam perguntas, coletam dados e interpretam resultados. E assim que a computação criou modelos para que programadores consigam ler códigos e facilitar a manutenção sem ter que aprender o código novamente, ou se forçar a comentar cada linha.

“Qualquer um pode escrever um código que o computador entenda. Bons programadores escrevem códigos que os humanos entendam.” – Martin Fowler

Os paradigmas de programação são um meio de padronizar como se é estruturado um projeto, a fim de facilitar a manutenção, escalabilidade e leitura do código. Ou seja, informam a melhor maneira de solucionar problemas usando uma linguagem em específico. Com isso, a linguagem se torna mais uma ferramenta, que possui utilidade em um certo cenário, enquanto o paradigma é uma forma eficaz de usar aquela ferramenta.

Por mais que seja um conceito de “programação”, os paradigmas datam desde os primórdios dos computadores, na década de 1950 à 1960, mas ainda não possuía esse

conceito, eram apenas “meios” e formas de abordar a programação, algo que era muito complexo naquela época, já que as máquinas possuíam pouca capacidade de processamento e tinha uma complexidade mais alta, tendo em mente que os computadores levavam dias para retornar algo e não havia as diversas abstrações das linguagens de hoje.

Assim foi implementado os paradigmas: Imperativo e Declarativo. Eles se tornaram mais complexos e definiram muitos outros métodos de programação, como o OOP (Object Oriented Programming), Funcional, entre outros.

2. Paradigma Lógico

O paradigma lógico emergiu na década de 1970, sendo fundamentalmente distinto dos outros paradigmas. A abordagem do paradigma lógico é expressar programas na forma de lógica simbólica e usar um processo de inferência lógica para produzir resultados, sendo algumas de suas principais aplicações nas áreas de inteligência artificial, processamento de linguagem natural, criação de sistemas especialistas e prova de teoremas.

Dentre as linguagens de programação que utilizam esse paradigma, podemos citar QLISP, Mercury e Prolog. São utilizadas na solução de problemas que envolvem inteligência artificial, criação de programas especialistas e comprovação de teoremas.

Segue abaixo um exemplo de código em Prolog que exemplifica o uso do paradigma de programação Lógico:

```
pai(joao, maria).
pai(joao, pedro).
mae(maria, ana).
mae(maria, jose).

irmao(X, Y) :- pai(Z, X), pai(Z, Y), X \= Y.

?- irmao(pedro, ana).
```

Figura 1. Exemplo de código feito em Prolog com Paradigma Lógico.

Na figura 1, são definidos quatro fatos: que João é pai de Maria e de Pedro, e que Maria é mãe de Ana e de José. Em seguida, é definido um predicado chamado "irmao", que tem dois argumentos, X e Y, e que é verdadeiro se X e Y são irmãos. Para isso, é necessário que exista um pai Z que seja pai tanto de X quanto de Y, e que X e Y sejam diferentes.

Por fim, o predicado "irmao" é utilizado para verificar se Pedro e Ana são irmãos. O resultado dessa consulta é falso, pois Pedro e Ana não têm o mesmo pai.

3. Comparativo entre este paradigma e o orientado a objetos.

Agora que compreendemos melhor sobre o Paradigma lógico, podemos compara-lo com o Paradigma mais comum utilizado em projeto reais, o Paradigma Orientado a Objetos.

3.1 Exemplo em Paradigma Lógico

Suponha que queremos escrever um programa em Prolog que determine se uma pessoa é um adulto ou não, com base em sua idade. Podemos definir fatos e regras para resolver esse problema da seguinte forma: prolog

```
1 % Definir fatos: nome, idade
2 pessoa(joao, 20).
3 pessoa(maria, 30).
4 pessoa(pedro, 15).
5 pessoa(ana, 25).
6
7 % Definir regra: pessoa é adulta se tem 18 anos ou mais
8 adulto(Nome) :-
9     pessoa(Nome, Idade),
10    Idade >= 18.
```

Figura 2. Exemplo de código feito em Prolog com Paradigma Lógico.

3.1 Exemplo em Paradigma Orientado a Objetos

Suponha que queremos escrever um programa em Java que modele uma conta bancária simples. Podemos definir uma classe Conta com os atributos numero, saldo e titular, e os métodos depositar, sacar e consultarSaldo, da seguinte forma:

```
public class Conta {
    private int numero;
    private double saldo;
    private String titular;

    public Conta(int numero, double saldoInicial, String titular) {
        this.numero = numero;
        this.saldo = saldoInicial;
        this.titular = titular;
    }

    public void depositar(double valor) {
        this.saldo += valor;
    }

    public void sacar(double valor) {
        if (valor > this.saldo) {
            throw new IllegalArgumentException("Saldo insuficiente");
        }
        this.saldo -= valor;
    }

    public double consultarSaldo() {
        return this.saldo;
    }
}
```

Figura 3: Exemplo de código feito em Java com Paradigma Orientado a Objetos.

Podemos criar objetos Conta e chamar seus métodos para depositar, sacar e consultar o saldo.

```
Conta minhaConta = new Conta(123, 1000.0, "João");
minhaConta.depositar(500.0);
minhaConta.sacar(200.0);
double saldoAtual = minhaConta.consultarSaldo();
System.out.println("Saldo atual: " + saldoAtual);
```

Figura 5. Uso da Classe Conta criada na Figura 4.

Esse programa criará uma conta com número 123, saldo inicial de 1000.0 e titular "João". Em seguida, depositará 500.0, sacará 200.0 e imprimirá o saldo atual (que deve ser 1300.0).

4. Conclusão

Concluem-se que o paradigma lógico ainda é muito utilizado em aplicações referente a inteligência artificial, pois tem grande afinidade na área lógica-matemáticas. Porém fora desse escopo, perde sua força para outros paradigmas, principalmente o orientado a objetos.

As linguagens de programação Lógico mais conhecidas, como Prolog, oferecem uma sintaxe simples e elegante que permite expressar conceitos lógicos de forma clara e concisa. O paradigma Lógico também é conhecido por sua capacidade de lidar com problemas complexos.

Embora existam diferenças significativas entre a programação Lógico e a programação orientada a objetos, ambas têm suas aplicações e são importantes em diferentes contextos. O desenvolvimento de aplicações complexas muitas vezes requer o uso de múltiplos paradigmas de programação para atender a diversos requisitos.

Em resumo, o paradigma de programação Lógico é uma abordagem poderosa e versátil para a solução de problemas baseados em relações e inferências lógicas. Seu uso tem se expandido em diversas áreas, e o aprendizado de linguagens de programação Lógico pode ser uma habilidade valiosa para programadores e cientistas da computação.

References

- Yuri, P. (2023) “História da Programação”, <https://www.infoescola.com/informatica/historia-da-programacao/>, Março.
- Isaac, F. de S. (2021) “Paradigmas de Programação”, <https://guia.dev/pt/pillars/languages-and-tools/programming-paradigms.html>, Março.
- Eduardo, S. (2022) “Quais são os paradigmas de programação mais importantes?”, https://blog.geekhunter.com.br/quais-sao-os-paradigmas-de-programacao/#Paradigmas_de_programacao_Declarativos, Março.
- Administrador (2021) “As vantagens de estabelecer processos bem definidos no desenvolvimento de softwares”, <https://blog.cronapp.io/as-vantagens-de-estabelecer-processos-bem-definidos-no-desenvolvimento-de-softwares/>, Março.
- Cairo, N. (2020) “Paradigmas de programação: o que são e quais os principais?”, <https://blog.betrybe.com/tecnologia/paradigmas-de-programacao/>, Março.

- Aryclenio, B. (2020) “Programação imperativa e declarativa”, <https://dev.to/aryclenio/programacao-imperativa-e-declarativa-1ooi>, Março.
- Ivan, L. M. R. (2003) “Estruturas de dados”, <https://www.dca.fee.unicamp.br/cursos/EA876/apostila/HTML/node10.html>, Março.
- Yuri, P. (2023) “História do Java”, <https://www.infoescola.com/informatica/historia-do-java/>, Março.
- Tedesco, Kennedy. Linguagens e paradigmas de programação. Treinaweb. 2016. Disponível em <<https://www.treinaweb.com.br/blog/linguagens-e-paradigmas-de-programacao>>. Acesso em 13 de maio de 2023