

# Paradigmas de Programação - Resumo

## Paradigma Imperativo

Baseado em instruções sequenciais que alteram o estado do programa. Foca em *\*como\** as coisas devem ser feitas, usando estruturas como loops, variáveis e condicionais. Linguagens típicas: C, Python, Java (em parte).

## Paradigma Declarativo

Descreve *\*o que\** deve ser feito, e não como fazer. O controle do fluxo fica sob responsabilidade do sistema. Inclui programação funcional e lógica. Linguagens típicas: SQL, HTML, Prolog.

## Paradigma Funcional

Um subtipo do paradigma declarativo. Computação baseada em funções matemáticas puras, sem efeitos colaterais. Evita estado mutável e dados compartilhados. Linguagens típicas: Haskell, Lisp, Elixir, Scala.

## Paradigma Orientado a Objetos

Organiza o software em objetos que combinam dados e comportamentos. Utiliza conceitos como herança, polimorfismo, encapsulamento e abstração. Linguagens típicas: Java, Python, C++, Ruby.

## Paradigma Lógico

Baseado em lógica formal e regras. O programa consiste em fatos e regras, e o sistema tenta provar ou refutar proposições. Usado em inteligência artificial e sistemas especialistas. Linguagens típicas: Prolog.

## Paradigma Concorrente

Foca na execução simultânea de processos ou threads. Muito usado em sistemas distribuídos ou de alto desempenho. É complementar e pode coexistir com outros paradigmas.