

# Fundamentos

## Introdução

**Prof. Edson Alves**

Faculdade UnB Gama

2020

---

# Sumário

1. **Motivação**
2. **Contexto Histórico**

# Paradigmas de Programação

O que é paradigma?

# Paradigmas de Programação

O que é programação?

# Termos associados aos paradigmas de programação

Programação Imperativa

CÁLCULO  $\lambda$

*Programação Funcional*

**Orientação à objetos**

*Multiagentes*

Programação Concorrente

PROGRAMAÇÃO DECLARATIVA

Orientação à Configuração

Programação Estruturada

**Máquinas de Turing**

# Benefícios do estudo dos paradigmas de programação

- ▶ Aumento da capacidade de expressar ideias
- ▶ Escolhas bem fundamentadas das linguagens de programação a serem utilizadas em um projeto
- ▶ Melhora na capacidade de aprendizado de novas linguagens
- ▶ Uso mais eficaz das linguagens já conhecidas
- ▶ Maior entendimento das diferentes implementações de um mesmo conceito
- ▶ Visão mais ampla da computação como um todo

# Os ideais de Leibniz

O matemático alemão Gottfried Wilhelm Leibniz (1646-1716) tinha dois ideais:

1. Criar uma “*linguagem universal*” na qual todos os problemas pudessem ser descritos
2. Encontrar um método de decisão para todos os problemas descritos nesta linguagem pudessem ser resolvidos

# Teoria dos conjuntos e lógica de primeira ordem

No que diz respeito aos problemas matemáticos, o primeiro ideal de Leibniz pode ser alcançado por meio de uma Teoria de Conjuntos formulada em termos de uma Lógica de Primeira Ordem.

O matemático/lógico/filósofo inglês Bertrand Russel (1872-1970) e o matemático/lógico alemão Ernst Zermelo (1871-1953) trouxeram grandes contribuições para esta questão.



# A grande questão

O segundo ideal trazia consigo uma grande questão filosófica, que ficou conhecida como *Entscheidungsproblem*:

*É possível resolver todos os problemas descritos na linguagem universal?*

# Turing e Church

A resposta negativa para o *Entscheidungsproblem* foi dada em 1936, independentemente, por dois grandes matemáticos. Para tal, eles precisaram formalizar a noção de decidibilidade, ou computabilidade:

- ▶ Alonzo Church (1936) inventou um sistema formal, denominado Cálculo  $\lambda$ , e definiu a noção de função computável por meio deste sistema
- ▶ Alan Turing (1936/1937) criou as Máquinas de Turing, e definiu computabilidade em termos destas máquinas

Características de ambos modelos estiveram presentes nas diversas linguagens de programação ao longo da história.

## Referências

1. **BARENDREGT**, Henk; **BARENDSSEN**, Erik. *Introduction to Lambda Calculus*, March 2000.
2. **HALE**, M. *Essentials of Mathematics: Introduction to Theory, Proof, and the Professional Culture*, Mathematical Association of America, 2003. (**eBrary**)
3. Wikipéida. [Alan Turing](#), acesso em 09/01/2020.
4. Wikipédia. [Alonzo Church](#), acesso em 09/01/2020.
5. Wikipédia. [Bertrand Russel](#), acesso em 09/01/2020.
6. Wikipédia. [Ernst Zermelo](#), acesso em 09/01/2020.
7. Wikipédia. [Gottfried Wilhelm Leibniz](#), acesso em 09/01/2020.