

# Introdução

## Teoria da Computação – Ciência da Computação

Prof. Daniel Saad Nogueira Nunes



**INSTITUTO  
FEDERAL**

Brasília

---

Campus  
Taguatinga

# Teoria da Computação

- ▶ Por que estudar Teoria da Computação?

# Teoria da Computação

Por que estudar TC?

- ▶ A prática tem relação intrínseca com a Teoria.

# Teoria da Computação

## Por que estudar TC?

- ▶ Está projetando uma nova linguagem de programação: Gramáticas Livres de Contexto.
- ▶ Acredita que o problema que você quer resolver é difícil: que tal olhar na teoria da NP-Completeness?
- ▶ Será que o problema que você quer resolver é possível de ser resolvido? Computabilidade pode ajudar a te responder.
- ▶ Casamento de padrões ou expressões regulares: Linguagens Formais e Autômatos.

# Teoria da Computação

## Por que estudar TC?

- ▶ Precisa comparar as suas soluções com outras: que tal analisar o seu algoritmo?
- ▶ Seu algoritmo está lento? Tentou utilizar outro paradigma de projeto?

# Teoria da Computação

## Por que estudar TC?

- ▶ Além dos motivos óbvios, ao estudar Teoria, você consegue enxergar um lado mais simples e elegantes dos modelos computacionais.
- ▶ Um design elegante e simples pode influenciar em uma aplicação elegante, eficiente e livre de erros.
- ▶ Um curso de Teoria reforça o lado estético, o que possibilita você criar sistemas mais belos.

# Teoria da Computação

## Por que estudar TC?

- ▶ Estudar Teoria também ajuda a expandir a mente.
- ▶ Tecnologia fica ultrapassada em anos, a teoria não.
- ▶ As habilidades de se expressar bem, resolver problemas, e saber quando você não pode resolver um problema de um determinado jeito são cruciais.
- ▶ Teoria trabalha com isso.

# Subáreas

- ▶ Três das principais subáreas da Teoria da Computação são:
  1. Teoria dos Autômatos.
  2. Computabilidade.
  3. Complexidade.
- ▶ Elas estão relacionadas por uma questão: “Quais são as capacidades e limitações dos computadores?”
- ▶ É claro que cada área vai interpretar e atacar esta indagação da sua própria forma.



# Complexidade Computacional

## Complexidade Computacional

- ▶ Problemas computacionais vem em diferentes formas.
- ▶ Alguns são fáceis, outros médios e outros difíceis.
- ▶ Por exemplo: o problema da ordenação é dito **fácil**. Mesmo um computador fraco com um algoritmo eficiente pode ordenar milhões de números em pouco tempo.
- ▶ O problema do escalonamento, que consiste alocar recursos de modo a satisfazer restrições já é mais complicado. Se você tem milhares de recursos, a computação pode levar centenas de anos.

# Complexidade Computacional

- ▶ O que faz alguns problemas mais difíceis do que os outros?
- ▶ Esta é a questão principal da área de Complexidade Computacional.
- ▶ Não é uma questão fácil. Problemas similares podem ter dificuldades bem distintas.

# Complexidade Computacional

- ▶ Uma das principais contribuições desta área é a classificação de problemas em classes de complexidade.
- ▶ Através destas classes, podemos demonstrar que um determinado problema é difícil ao “compará-los” com outros problemas difíceis e verificar serem semelhantes.

# Complexidade Computacional

- ▶ Uma vez identificado que um problema é difícil, o que pode ser feito?
- ▶ Desistir ?

# Complexidade Computacional

- ▶ Se o problema é difícil não quer dizer que não existam instâncias que podem ser resolvidas eficientemente.
- ▶ Se um problema é difícil, você pode tentar outras abordagens, como algoritmos aproximados e heurísticos.
- ▶ Nem sempre precisamos da melhor resposta possível.

# Complexidade Computacional

- ▶ Problemas difíceis também são úteis na prática.
- ▶ A área de Criptografia depende de problemas difíceis para garantir a segurança.

# Computabilidade

## Computabilidade

- ▶ Na primeira metade do século XX, matemáticos como Kurt Gödel, Alonzo Church e Alan Turing descobriram que existem problemas que não podem ser resolvidos por computadores.
- ▶ Não importa quanto tempo você dê para eles, eles não irão conseguir resolver estes problemas.

# Computabilidade

- ▶ Tome o problema de determinar se um enunciado matemático é verdadeiro ou falso.
- ▶ Se conseguíssemos resolver isso através de um computador, as coisas seriam bem mais simples.
- ▶ Parece até uma coisa natural, pois a computação está relacionada com a Matemática de certa forma.
- ▶ No entanto, não existe nenhum algoritmo que consegue resolver este problema.



# Computabilidade

- ▶ Complexidade Computacional e Computabilidade estão relacionadas, mas são diferentes.
- ▶ Complexidade Computacional: classifica os problemas e, graus de dificuldade.
- ▶ Computabilidade: classifica os problemas em resolvíveis ou não.

# Teoria de Autômatos

## Teoria de Autômatos

- ▶ A Teoria de Autômatos foca nas definições e propriedades dos modelos de computação.
- ▶ Estes modelos desempenham um papel muito importante em diversas áreas da computação.
  - ▶ Design de Hardware.
  - ▶ Processamento de palavras.
  - ▶ Tradutores.
  - ▶ Verificação Formal.
  - ▶ ...

# Teoria da Computação

- ▶ Neste curso, focaremos em computabilidade com algumas pinceladas de Complexidade Computacional.
- ▶ Teoria de Autômatos: Linguagens Formais e Autômatos (6°).