



# ~~O PIOR CURSO DE~~ ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES ~~DO UNIVERSO~~

USANDO LINGUAGEM C E RAYLIB  
INTRODUÇÃO

PROF. DR. DAVID BUZATTO

# INTRODUÇÃO

- Objetivo

- Tentar ensinar, quem não sabe programar, o básico sobre algoritmos e programação de computadores;

- Motivação

- Necessidade de criar um curso de verdade, gratuito e que preste sobre o tema;

- Público alvo

- Qualquer pessoa que tenha e que saiba usar um computador pessoal, que goste de ciências exatas e que queira aprender o bô-á-bá sobre programação, sem conversa fiada!

# REQUISITOS

- Um computador pessoal, não muito antigo;
  - não vai dar certo fazer pelo celular nem em máquinas muito ultrapassadas;
- Windows 10 ou superior;
  - quem usa Linux pode fazer também, mas vai ter que se virar com ferramentas e configurações;
- Vontade de aprender;
- Disponibilidade para treinar;
  - teremos MUITOS exercícios;

# O QUE SERÁ USADO

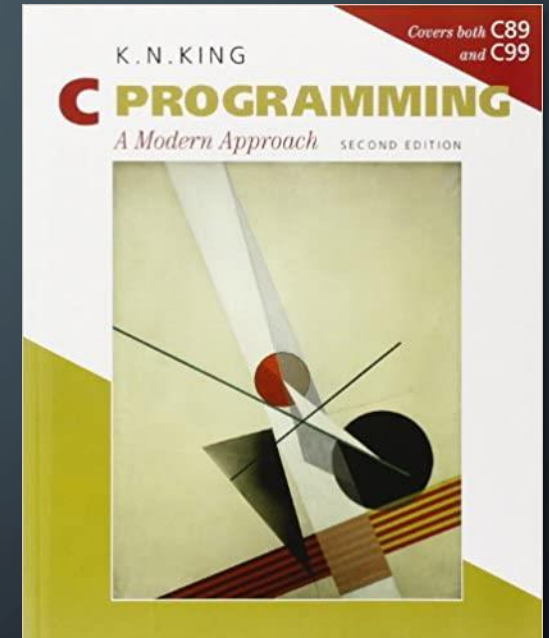
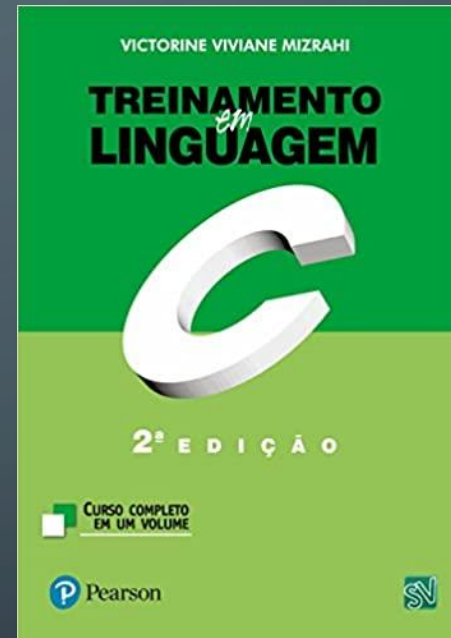
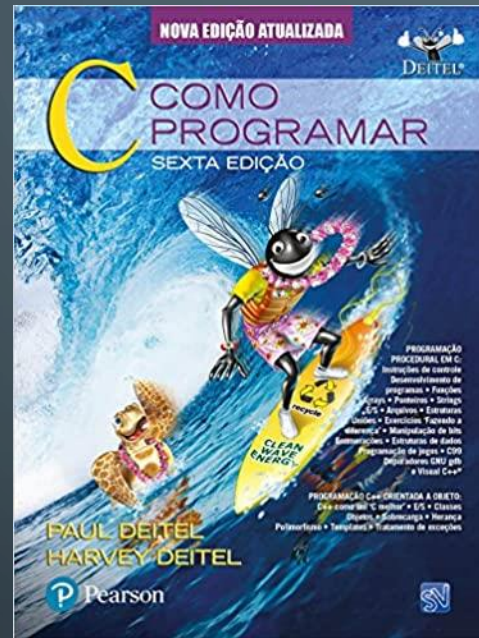
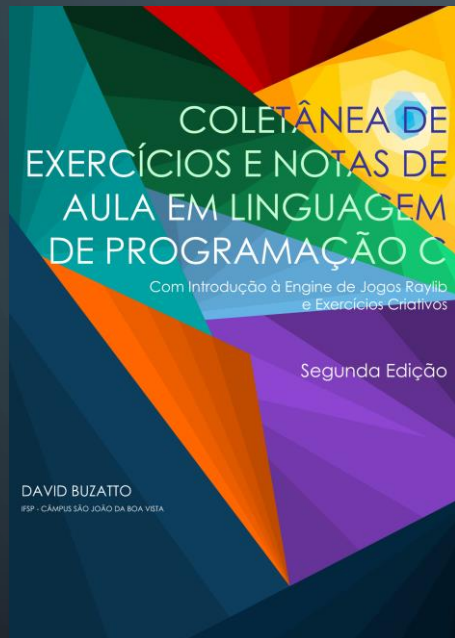
- Linguagem de programação C;
  - compilador GCC do MinGW, ambos instalados via MSYS2;
- Visual Studio Code;
  - usaremos algumas extensões para nos ajudar;
- Raylib;
  - engine (motor) para desenvolvimento de jogos;
- JJudge;
  - corretor de exercícios;
- Livro: “Coletânea de Exercícios e Notas de Aula em Linguagem de Programação C: com introdução à engine de jogos Raylib e exercícios criativos”, 2ª edição.

# CONTEÚDO

1. Preparação do ambiente;
2. Hello World!
3. Entrada e saída padrão;
4. Raylib;
5. Estruturas condicionais;
6. Estruturas de repetição;
7. Arrays ~~(não são vetores-pô!)~~;
8. Algoritmos de ordenação elementares;
9. Biblioteca matemática padrão;
10. Funções;
11. Ponteiros;
12. Caracteres e strings;
13. Estruturas;
14. Uniões e Enumerações;
15. Organização de código;
16. Arquivos;
17. Recursividade;
18. Algoritmos de ordenação não elementares;
19. Funções variádicas;
20. Uso avançado de ponteiros;
21. Classes de armazenamento, qualificadores e inicialização;
22. Resolução de exercícios por capítulo do livro;
23. Desenvolvimento de simulações e jogos usando Raylib.

# BIBLIOGRAFIA SUGERIDA

6/17





# VAMOS COMEÇAR?

# O QUE É ALGORITMO?

- **Informal:**

- Conjunto finito de instruções, executadas em ordem lógica, que tem como objetivo resolver um problema;

- **Formal:**

- Conjunto das regras e procedimentos lógicos perfeitamente definidos que levam à solução de um problema em um número de etapas (Dicionário Houaiss da Língua Portuguesa, 2001);
- *The term algorithm is used in computer science to describe a problem-solving method suitable for implementation as a computer program (Algorithms in C, Sedgewick, 1998);*
- *Methods for solving problems that are suited for computer implementation (Algorithms, Sedgewick, 2011).*



# PENSAR COMO UM CIENTISTA DA COMPUTAÇÃO

- O que é computação?
- Tipos de conhecimento:
  - **Declarativo:** definição de algo;
    - $\sqrt{x} = y \Rightarrow y^2 = x \text{ e } y \geq 0$
  - **Imperativo:** como fazer algo (receita);

```
início
  escolher um valor para y
  se  $y^2 \approx x$  então
    pare  $\rightarrow y$ 
  senão
     $y \leftarrow \frac{(y + \frac{x}{y})}{2}$ 
  repita
```

# PENSAR COMO UM CIENTISTA DA COMPUTAÇÃO

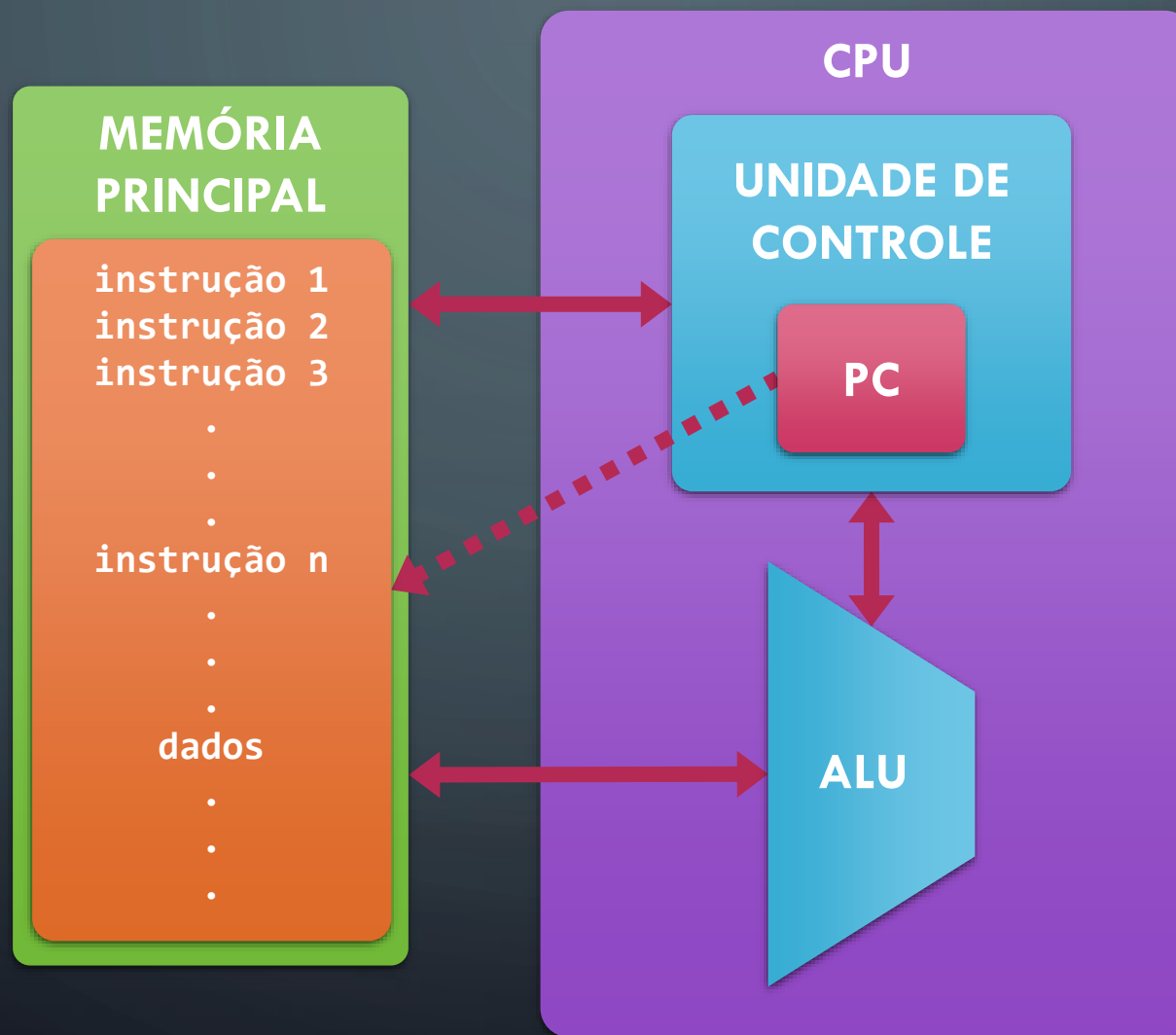
- Como automatizar “mecanicamente” a receita apresentada?
  - **Computador de programa fixo:**
    - Conjunto de circuitos projetado para resolver apenas um tipo problema;
    - **Exemplo:** calculadora.

# PENSAR COMO UM CIENTISTA DA COMPUTAÇÃO

- Ao invés de haver uma máquina com um circuito fixo, que tal haver uma máquina que recebe como entrada um circuito, ou uma receita, se reorganiza internamente, e é capaz de realizar aquela receita, ou seja, que tenha a capacidade de agir como a receita?
- **Computador de programa armazenado!**

# ARQUITETURA DE VON NEUMANN

12/17



# TIPOS DE LINGUAGENS DE PROGRAMAÇÃO

13/17

Alto Nível



Linguagem Natural



Linguagem de Máquina

Baixo Nível

# LINGUAGENS DE PROGRAMAÇÃO

- Propósito geral ou específica de domínio:
  - **Geral:** linguagem de programação projetada para resolver problemas de diversas áreas do conhecimento ou para realizar atividades diversas;
    - C, C++, Java, Python, Rust, Pascal, Ruby, Go etc.
  - **Específica de domínio:** linguagem de programação projetada para resolver problemas e/ou realizar atividades de áreas específicas.
    - HTML (*Hypertext Markup Language*);
    - SQL (*Structured Query Language*);
    - HDL (*Hardware Definition Language*);
    - MATLAB (MATrix LABoratory) etc.



# LINGUAGENS DE PROGRAMAÇÃO

- Compilada ou Interpreta:

**C, C++, Rust, COBOL, FORTRAN, Pascal etc.**



**Java, C#, Closure, Groovy, Kotlin, Scala, Python etc.**



**JavaScript, Ruby, Lua etc.**



# LINGUAGEM DE PROGRAMAÇÃO C

- Desenvolvida em 1972 no Bell Labs por Dennis Ritchie e Ken Thompson com o objetivo de ser usada na criação de sistemas operacionais, no caso o UNIX, e que fosse de fácil leitura e manutenção;
- Padronizada pela ISO/IEC 9899;
- Atualizada constantemente;
- Última versão, C23/C2X (ISO/IEC 9899:2024);
- Ancestrais: BCPL  $\rightarrow$  B  $\rightarrow$  C.

# MÃOS À OBRA!