

Lógica de programação



Iniciando na programação e criando
algoritmos com Linguagem C

Matheus Madureira

INTRODUÇÃO

O que é a Linguagem C?

A Linguagem C é uma linguagem de programação criada por Dennis Ritchie em 1972. Amplamente utilizada até hoje e influenciou muitas outras linguagens. A linguagem C é compilada, o que significa que o código-fonte é traduzido para código de máquina antes da execução pelo computador. Nesse ebook vamos abordar desde os tipos de dados básicos até estruturas de decisão e exemplos práticos.



Dennis Ritchie



Capítulo

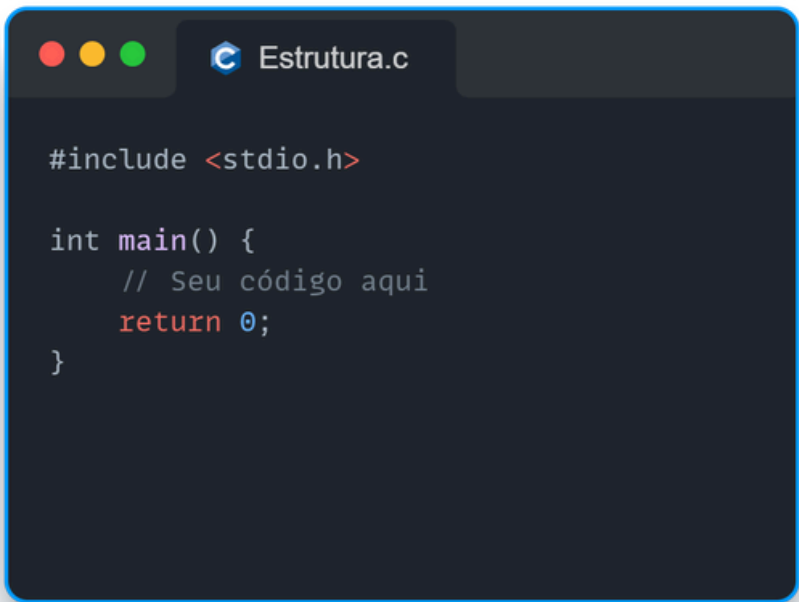
01

Estrutura Básica

Criando a estrutura para fazer um
código em linguagem C

Estrutura Básica de um Programa em C

Para começar, criamos um programa em C com a seguinte estrutura:



```
#include <stdio.h>

int main() {
    // Seu código aqui
    return 0;
}
```

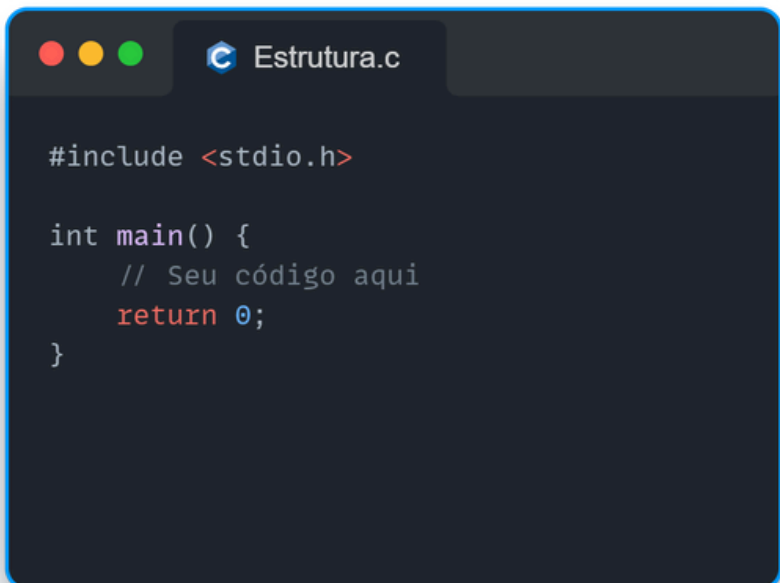


Estrutura Básica de um Programa em C

-A diretiva `#include <stdio.h>` inclui a biblioteca padrão de entrada/saída.

-A função `main()` é a função principal que será executada após a compilação.

-O `return 0;` indica que o programa foi encerrado.



```
#include <stdio.h>

int main() {
    // Seu código aqui
    return 0;
}
```



Capítulo

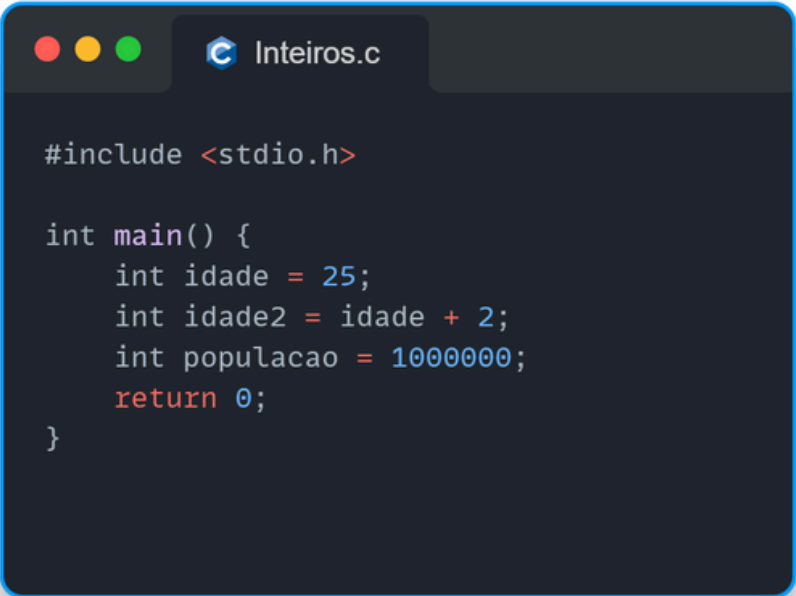
02

Tipos de Dados

As variáveis são essenciais para
representar informações e realizar operações

Inteiros (int)

Os inteiros(int) representam números inteiros, positivos ou negativos. Eles não possuem parte decimal. Perceba que é possível realizar operações matemáticas, como: adição, subtração, multiplicação e divisão.



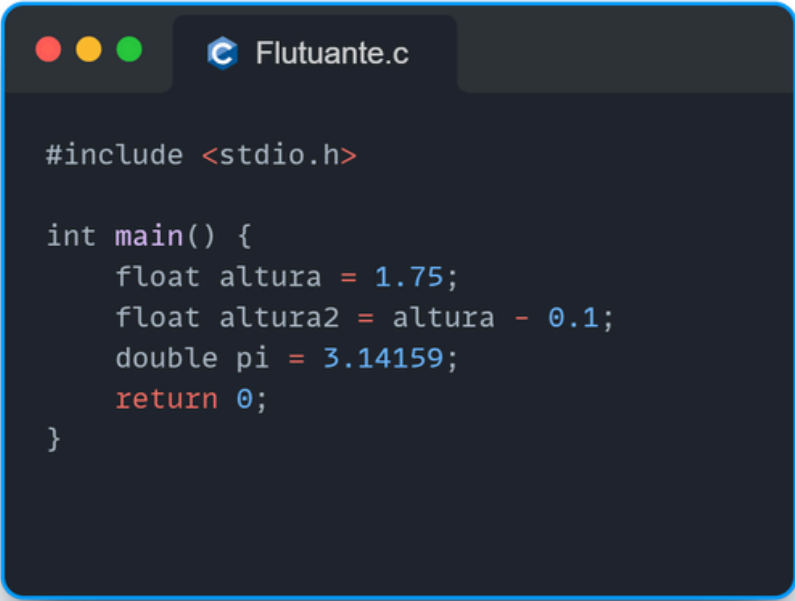
```
#include <stdio.h>

int main() {
    int idade = 25;
    int idade2 = idade + 2;
    int populacao = 1000000;
    return 0;
}
```



Ponto Flutuante (float e double)

Os tipos de ponto flutuante permitem representar números com parte decimal. O float é usado para números de precisão simples, enquanto o double oferece maior precisão. Exemplos:



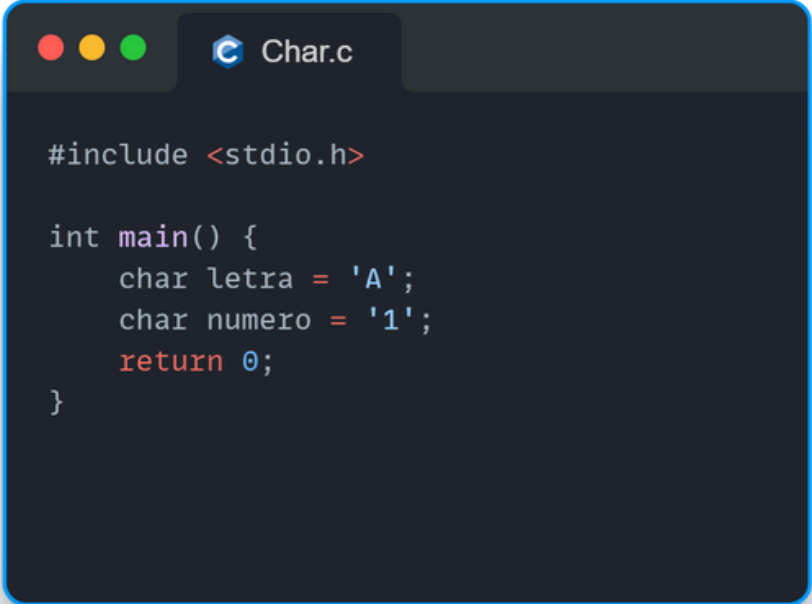
```
#include <stdio.h>

int main() {
    float altura = 1.75;
    float altura2 = altura - 0.1;
    double pi = 3.14159;
    return 0;
}
```



Caracteres (char)

O tipo char representa um único caractere (letra, número ou símbolo) e deve ser colocado entre aspas em sua declaração. Exemplo:



```
#include <stdio.h>

int main() {
    char letra = 'A';
    char numero = '1';
    return 0;
}
```



Capítulo

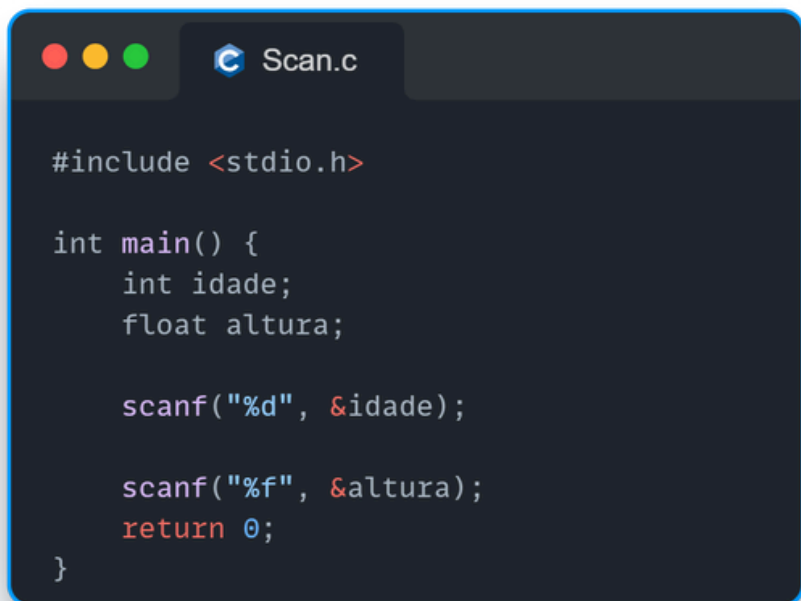
03

Entrada e Saída

Em C, usamos funções específicas para obter dados do usuário (entrada) e para mostrar resultados ou mensagens (saída)

Entrada de Dados

A função principal para entrada de dados em C é `scanf`. Ela permite que o programa leia valores fornecidos pelo usuário e os armazene em variáveis.



```
#include <stdio.h>

int main() {
    int idade;
    float altura;

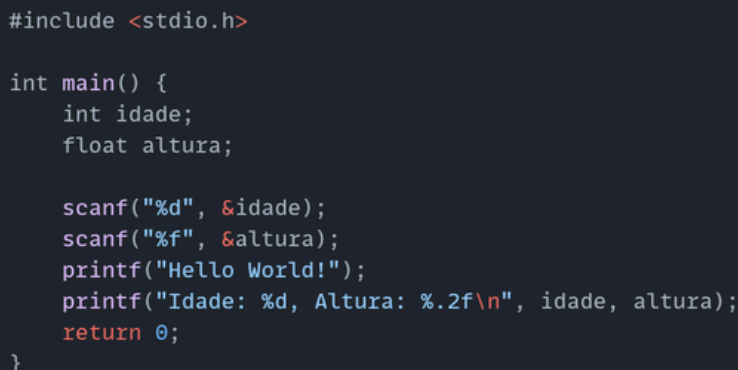
    scanf("%d", &idade);

    scanf("%f", &altura);
    return 0;
}
```



Saída de Dados

Para mostrar dados na tela, utilizamos a função `printf` acompanhada da variável que iremos apresentar(se houver).



```
#include <stdio.h>

int main() {
    int idade;
    float altura;

    scanf("%d", &idade);
    scanf("%f", &altura);
    printf("Hello World!");
    printf("Idade: %d, Altura: %.2f\n", idade, altura);
    return 0;
}
```



Capítulo

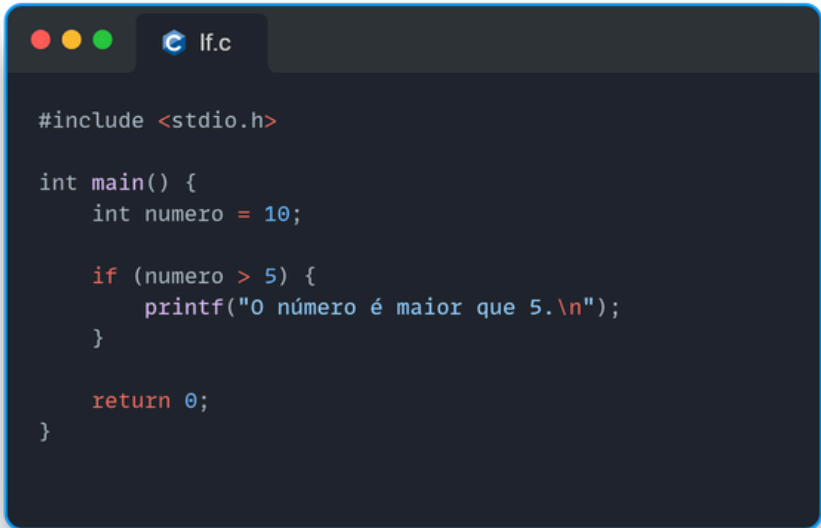
04

Estruturas de Decisão

Entenderemos como usar “if”, “else if” e “else” para tomar decisões em nossos programas

If (se)

A estrutura if permite que o programa execute um bloco de código apenas se uma condição especificada for verdadeira. Se a condição for falsa, o bloco de código é ignorado.



```
#include <stdio.h>

int main() {
    int numero = 10;

    if (numero > 5) {
        printf("O número é maior que 5.\n");
    }

    return 0;
}
```



Else (senão)

A estrutura if-else é uma extensão da estrutura if. Ela permite que o programa execute um bloco de código se a condição for verdadeira e um bloco diferente (else) se a condição for falsa.

```

#include <stdio.h>

int main() {
    int numero = 3;

    if (numero > 5) {
        printf("O número é maior que 5.\n");
    } else {
        printf("O número não é maior que 5.\n");
    }

    return 0;
}
```



Else-if (senão se)

A estrutura else-if permite testar múltiplas condições em sequência. É útil quando há mais de duas possibilidades a serem verificadas.

```
Elself.c

#include <stdio.h>

int main() {
    int numero = 7;

    if (numero > 10) { //Entra se for verdadeira
        printf("O número é maior que 10.\n");
    } else if (numero > 5) { //Entra se for verdadeira e anterior falsa
        printf("O número é maior que 5, mas menor ou igual a 10.\n");
    } else { // Se não for nenhuma das anteriores verdadeiras
        printf("O número é 5 ou menor.\n");
    }

    return 0;
}
```



Capítulo

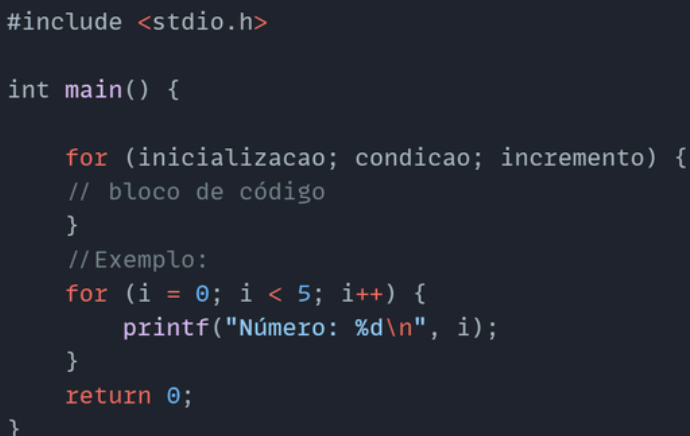
05

Laços de Repetição

Entenderemos como usar os laços “for” e
“while” no seu programa

For

A estrutura de controle for é usada para repetir um bloco de código um número específico de vezes.



```
#include <stdio.h>

int main() {

    for (inicializacao; condicao; incremento) {
        // bloco de código
    }
    //Exemplo:
    for (i = 0; i < 5; i++) {
        printf("Número: %d\n", i);
    }
    return 0;
}
```



While

A estrutura de controle while é usada para repetir um bloco de código enquanto uma condição específica for verdadeira. Ao contrário do for, o while é mais flexível e é usado quando não sabemos quantas vezes o loop será executado.



```
#include <stdio.h>

int main() {
    int i = 0;
    while (condicao) {
        // bloco de código
    }
    //Exemplo:
    while (i < 5) {
        printf("Número: %d\n", i);
        i++;
    }
    return 0;
}
```



Capítulo Final

**Finalização e
Agradecimentos**

Obrigado por ler até aqui!

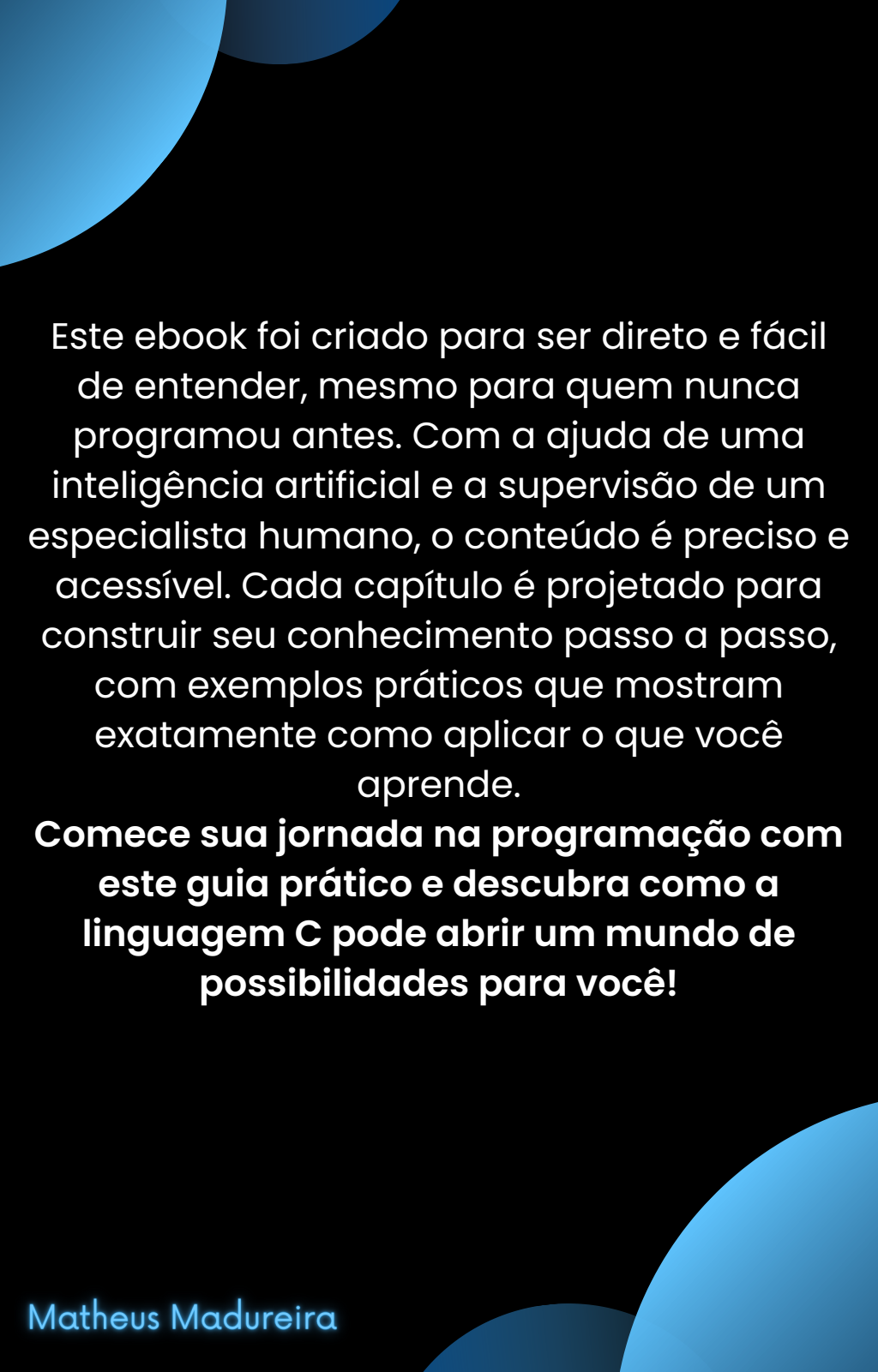
Espero que este ebook tenha sido útil em sua jornada de aprendizado da linguagem C. O conteúdo foi criado com o objetivo de tornar a programação acessível e compreensível, mesmo para aqueles que estão começando agora.

Este ebook foi desenvolvido com a assistência de uma inteligência artificial (IA), que forneceu conteúdos técnicos e exemplos práticos. No entanto, todo o material foi supervisionado e validado por mim, para garantir precisão, clareza e relevância.



<https://github.com/MatheusMadureiraa/prompts-recipe-to-create-a-ebook-SantanderBootcamp>





Este ebook foi criado para ser direto e fácil de entender, mesmo para quem nunca programou antes. Com a ajuda de uma inteligência artificial e a supervisão de um especialista humano, o conteúdo é preciso e acessível. Cada capítulo é projetado para construir seu conhecimento passo a passo, com exemplos práticos que mostram exatamente como aplicar o que você aprende.

Comece sua jornada na programação com este guia prático e descubra como a linguagem C pode abrir um mundo de possibilidades para você!