# Métodos para ler uma string em linguagem C

Em C, existem várias maneiras de ler uma string. A forma mais comum é usar funções da biblioteca padrão, como
scanf, gets, fgets, e outras mais avançadas, como getline. A seguir, explico cada uma delas e discuto as variações
de formatação de entrada com scanf.

1. Usando scanf

A função scanf é usada frequentemente para ler dados de entrada, incluindo strings. No entanto, ela possui limitações,
como não lidar bem com espaços em branco dentro da string.

Exemplo básico:

```
char nome[50];
scanf("%s", nome);
```

Isso lê uma string até o primeiro espaço em branco. Portanto, se você digitar "Maria Silva", apenas "Maria" será
armazenado na variável nome.

Variantes de formatação com scanf:

- Limitar o número de caracteres lidos:

```
scanf("%49s", nome);
```

- Ler até um caractere específico:

```
scanf("%[^n]", nome); // Lê até encontrar o caractere 'n'
```

- Ler até encontrar um espaço em branco ou um caractere específico:

scanf("%[^ ]", nome); // Lê até encontrar espaço em branco

- Ignorar espaços em branco:

scanf(" %s", nome);

2. Usando gets

A função gets lê toda a linha de entrada, incluindo espaços em branco, até encontrar uma nova linha (

). No entanto,

ela foi descontinuada a partir do C11 por questões de segurança, como a falta de controle sobre o tamanho da entrada.

Exemplo:

char nome[50];

gets(nome); // Não recomendado (descontinuada no C11)

3. Usando fgets

A função fgets é uma alternativa mais segura a gets, pois permite limitar o número de caracteres a serem lidos.

Exemplo:

char nome[50];

fgets(nome, sizeof(nome), stdin);

4. Usando getline

A função getline (não padrão do C, mas disponível em muitas implementações como no GNU C) é usada para ler uma linha

completa, sem limitação de tamanho pré-definido. Ela aloca dinamicamente o buffer necessário.

Exemplo:

```
char *nome = NULL;
size_t tamanho = 0;
getline(&nome, &tamanho, stdin);
```

Diferenças entre fgets e scanf:

- fgets:

  Lê até o final da linha ou até atingir o número máximo de caracteres permitidos.

  Pode capturar espaços e o caractere de nova linha.

  Não para de ler ao encontrar um espaço em branco.

- scanf:

  Pode ser formatado para ler uma palavra, várias palavras ou até com filtros específicos.

  Para de ler ao encontrar um espaço em branco.

Considerações finais:

- scanf é útil para casos simples, como ler uma única palavra.
- fgets é mais seguro e eficiente para capturar strings que contenham espaços.
- getline é ideal quando você não sabe o tamanho da entrada.