



Professor
Maromo

LINGUAGEM DE PROGRAMAÇÃO

Material 001



Github
maromo71

C



Histórico

- Criada em 1972 nos Bells Telephone laboratories por Dennis Ritchie.
- Finalidade: Permitir a escrita de um sistema operacional (Unix), usando uma linguagem de alto nível se comparado ao Assembly.
- A linguagem resulta da evolução de uma outra linguagem, chamada de B, desenvolvida no mesmo laboratório por Ken Thompson.



Vantagens

- **Rapidez:** Consegue obter performance semelhante ao Assembly, usando instruções em alto nível.
- **Simples:** número diminuto de palavras reservadas, de tipos de dados e de operadores.
- **Portável:** Padrão ANSI – código escrito em uma máquina pode ser compilado em outra máquina (com poucas ou sem alterações)
- **Popular:** É a mais conhecida e utilizada no mundo.
- **Modular:** Permite a programação modular, facilita a separação de projetos em módulos distintos e independentes, uso de funções.
- **Alto Nível:** Linguagem de terceira geração, permite acesso a maior parte das funcionalidades de Assembly.
- **Outros:** Bibliotecas adicionais, evolução (POO) C++. Java se baseia em C/C++.



Filosofia da programação em C

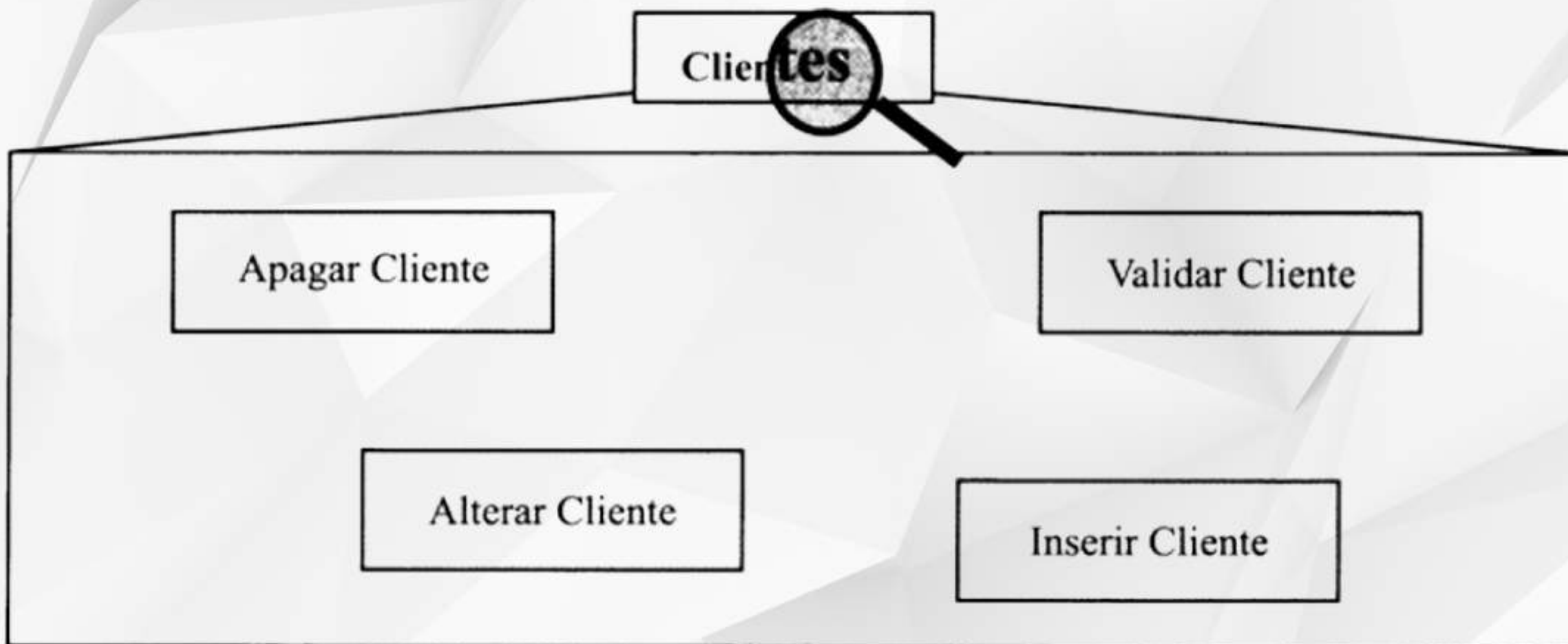


- A modularidade ajuda a tornar o código mais organizado, reutilizável e fácil de manter.
- Em linguagem C, a modularidade é frequentemente implementada usando funções e arquivos separados.

Cada módulo é implementado de maneira independente.

Cada módulo, por sua vez, é dividido nos diversos componentes que o compõe

Filosofia da programação em C



Possíveis componentes de um módulo Clientes

C versus C++



Linguagem C

- C é uma linguagem de programação procedural de nível médio, combinando elementos de linguagens de alto e baixo nível.
- É amplamente utilizado para desenvolvimento de sistemas e aplicações devido à sua eficiência e controle sobre a memória.
- A linguagem C oferece um conjunto limitado de palavras-chave e uma biblioteca padrão rica, permitindo extensibilidade.



Linguagem C++

- C++ é uma extensão da linguagem C, incorporando características de programação orientada a objetos.
- Além das bibliotecas padrão de C, C++ introduz a Standard Template Library (STL), facilitando operações com coleções de dados.
- C++ suporta sobrecarga de funções, polimorfismo, e herança, permitindo maior flexibilidade no design de software.

Ciclo do desenvolvimento



1. Edição do código fonte



2. Compilação do programa



3. Linkagem de Objeto



4. Execução do programa

4 fases



Anatomia de um programa em C



Um programa em linguagem C é formado por uma ou mais funções.



Cada função possui um nome exclusivo e corresponde à um bloco de código, delimitado por um par de chaves: { }



Contém um conjunto de declarações, expressões, comandos de controle e chamadas à outras funções.

Função: **main**

```
1: main()  
2: {  
3: }
```

A função denominada **main** é obrigatória em todos os programas, pois é o seu ponto de entrada, isto é, o programa começa a ser executado no início da função main e termina ao final desta função.

Normalmente a declaração desta função possui a seguinte forma: `main(void)`

Os parênteses sem mais nada após a função indicam que ela não recebe qualquer informação exterior.

Nota: C é Case Sensitive. Faz diferenciação entre maiúsculas e minúsculas. Todas as instruções de C são escritas em letras minúsculas. Usa-se letras maiúsculas quando se deseja utilizar variáveis, mensagens ou funções.

Exemplo Hello World



```
int main() {  
    printf("Olá Mundo!\n");  
    return 0;  
}
```




No exemplo anterior temos:

Linha 1

Função principal – entrada principal do programa.

Na Linha { e na última linha }:

Respectivamente início e fim do bloco de comandos.

Linha 2:

Comando de saída – exibe a mensagem “Ola mundo” na console.

Linha 3:

Finaliza o método

New Line "\n"

Em pascal usa-se os comandos Write e Writeln para a saída em tela, com a diferença que o segundo, além da saída, avança para a próxima linha.

Em C usa-se o mesmo comando printf. Porém para avançar para uma nova linha usa-se "**\n**" (**New Line**).

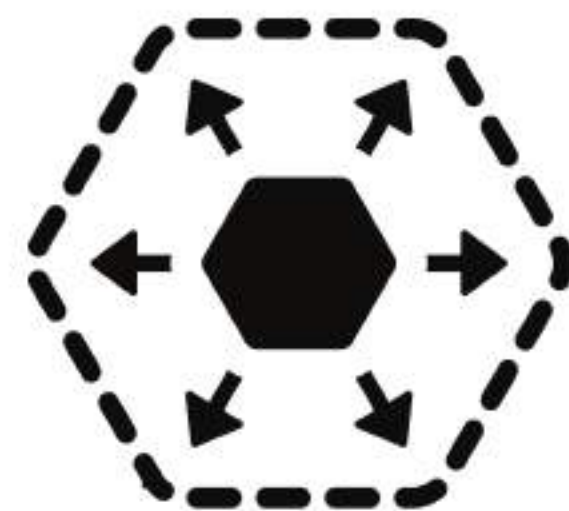
Problema com delimitador de string

```
1: #include <stdio.h>
2: main()
3: {
4:     printf("Hoje está um "LINDO" dia!!!\n");
5: }
```



Para que não ocorra erro de compilação usa-se um caractere “\” barra invertida antes da aspas duplas.

```
1: #include <stdio.h>
2: main()
3: {
4:     printf("Hoje está um \"LINDO\" dia!!!\n");
5: }
```



Caractere especial “\”

Usado para retirar o significado especial que um caractere apresenta.

No caso anterior, retirou o significado da delimitação de string dos caracteres (“”).

No caso de `\n` – representa um caractere que de outro modo seria quase impossível representar.

Veja a lista completa dos caracteres que podem ser representados pelo `\`.

Sequência de Escape	Caractere Representado	Descrição
\\	\	Barra invertida
\'	'	Apóstrofo ou aspas simples
\"	"	Aspas duplas
\n		Nova linha
\r		Retorno de carro
\t		Tabulação horizontal
\b		Retrocesso
\f		Avanço de página
\a		Sinal sonoro
\v		Tabulação vertical
\0		Caractere nulo (terminador nulo)
\ooo		Valor octal (onde 'ooo' é um número em base 8)
\xhh		Valor hexadecimal (onde 'hh' é um número em base 16)

Comentários

Comentários não são interpretados pelo compilador, ou seja, são ignorados pelo compilador, e o programa executável não terá qualquer sinal deles.

Comentário em C, é qualquer conjunto de caracteres compreendido entre os sinais de `/*` e `*/`

```
/******  
* PROG0199.C :    Comentários e Companhia      *  
* AUTOR:      Luís Damas                        *  
* DATA:      01/11/1995                       *  
*****/
```

```
/* PROG0199.C :    Comentários e Companhia  
* AUTOR:      Luís Damas  
* DATA:      01/11/1995  
*/
```





1. Olá, Mundo com Nova Linha

Escreva um programa que imprime a frase "Olá, Mundo!" seguida de uma nova linha usando um caractere de escape.

2. Aspas no Meio

Escreva um programa que imprime:

Ela disse: "A prática leva à perfeição."

Garanta que você está usando caracteres de escape para incluir as aspas duplas no meio da string.

3. Tabs e Espaços

Imprima a seguinte estrutura usando tabulações horizontais (`\t`):

```
Nome:      [Seu Nome]
Matrícula: [Número de Matrícula]
Curso:     [Nome do Curso]
```

Referências

DAMAS, L. M. D. Linguagem C. LTC, 2007.

HERBERT, S. C completo e total. 3a. ed. Pearson, 1997.

