



Linguagem C

Entendendo a Programação

By : Yngrid Monike



MINICURSO





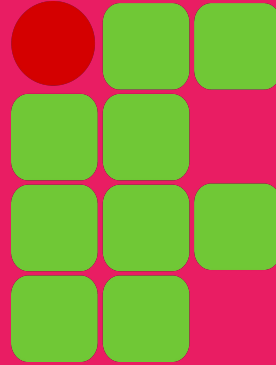
Yngrid Monike

Formação : Técnica em Informática

Profissão : Analista de Teste Pleno

@yngridmonike





Curso : ADS - ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

AGENDA

- Contextualizando
- História
- Linguagem de Programação
- Por que programar ?
- Exemplos de Códigos
- Tipos de Linguagem
- Estrutura Básica
- Ambiente de Desenvolvimento
- Plano de Estudos

História

- Ada Lovelace
- Alan Turing

-- História

- Condessa de Lovelace, conhecida como Ada Lovelace
- Foi a matemática que criou o primeiro algoritmo para ser processado por uma máquina, sendo a primeira programadora da história.
-
- Sim, Lovelace foi a primeira pessoa programadora de todos os tempos, e não apenas a primeira mulher a escrever um código.
- Seu fascínio foi ainda maior quando tomou conhecimento da Máquina Analítica, projeto de computador mecânico feito pelo professor Charles Babbage

-- História

- Alan Turing sucedeu o trabalho de Ada ele fez referência a seu trabalho.
- Turing, ficou conhecido como “o pai da computação moderna”.
- Foi um matemático e cientista da computação britânico responsável por formalizar o conceito de algoritmo criando a Máquina de Turing, que abriu as portas para a invenção dos computadores que utilizamos hoje em dia.

Linguagem de Programação

O que é ?

- - Definição

- A linguagem de programação é um método padronizado para comunicar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador.
- Ou seja “Idioma” linguagem de máquina a grosso modo.

Linguagem de Programação C

Referências

<https://www.inf.pucrs.br/~pinho/Laprol/IntroC/IntroC.htm>

-- Por que programar ?

- Quando o homem necessita do auxílio do computador para executar algumas tarefas.
- O Que é um programa ?
 - Conjunto de instruções de uma determinada linguagem através das quais, um computador executa algumas tarefas.

-- O Que é uma linguagem ? ?

- Consiste em um conjunto de palavras reservadas e regras de sintaxe que possibilita criar programas de computadores. Este conjunto de palavras possui regras de estruturação lógica e sintática própria.
- No nosso caso iremos utilizar a linguagem de programação C.

-- Exemplos de Códigos: C

Pseudo-código

```
leia(num)
para n de 1
    até 10 passo 1
        faça
            tab ← num * n
            imprima(tab)
fim-para;
```

C

```
scanf(&num);
for(n=1;n<=10;n++){
    tab=num*n;
    printf("\n %d",
    tab);
};
```

--Tipos de Linguagem : Baixo Nível

- São linguagens com foco na máquina, ou seja, utiliza instruções detalhadas que controla os circuitos internos do computador. Usualmente são genericamente chamadas de linguagens de máquina, Assembly ou de linguagem de montagem.
- Vantagens: Maior velocidade de processamento e ocupam menor espaço na memória.
- Desvantagens: Pouca portabilidade, ou seja, o código é gerado para um tipo de processador não serve para outro.

--Tipos de Linguagem: Alto Nível

- São linguagens voltadas para que haja uma maior interação entre o homem e a máquina.
- Necessitam de compiladores ou interpretadores para gerar as instruções do microprocessador.
- Vantagens: São compiladas ou interpretadas, têm maior portabilidade podendo ser executados em várias plataformas com pouquíssimas modificações.
- Desvantagens: São mais lentas e ocupam mais memória.

--Compiladores x Interpretadores

- A única linguagem que o computador entende é a linguagem de máquina.
- Programas escritos em um linguagem de alto nível, devem ser traduzidos para a linguagem de máquina.
-
- Os Programas que fazem esta tradução, classificam em:
 - INTERPRETADORES
 - COMPILADORES

--Compiladores x Interpretadores

- Os INTERPRETADORES, traduzem o código fonte em linguagem de máquina através da interpretação de cada instrução feita a medida que o software é executado. Necessitam de um componente interpretador presente na máquina.
- Os COMPILADORES, por sua vez, traduzem o código fonte em linguagem de máquina através da geração de um programa .
 - OBJ, que após ser linkeditado, torna-se um arquivo executável.
 - Em C os programas são compilados.

-- Sobre a Linguagem C x Histórico

- C foi originalmente desenvolvida por Dennis Ritchie e K. Thompson nos Laboratórios Bell nos anos 70. Derivadas de duas linguagens anteriores chamadas BCPL e B.
- Inicialmente para máquinas com o sistema operacional UNIX.
- Tornou-se uma das mais importantes e populares nos últimos dias.
- Foi projetada para o desenvolvimento de programas estruturados e modulares.

-- Sobre a Linguagem C x Histórico

- Nos anos seguintes, a popularidade da linguagem C aumentou consideravelmente, aumentando também o surgimento de ferramentas de desenvolvimento.
- Inicialmente, muitas implementações do C não traduziam fielmente a definição original, o que gerou alguns problemas de incompatibilidade
- A portabilidade proposta pela definição original ficou bastante comprometida.

-- Estrutura básica de um programa C

- Um programa em C consiste de uma ou várias funções, onde uma delas precisa ser denominada “main” e deve existir em algum lugar de seu programa. Esta função marca o início da execução do programa.
- Outras funções podem ser definidas pelo programador ou preencher a função main, porém em um programa executável em C, a função main deve sempre existir.

-- Estrutura básica de um programa C



```
main(Argumentos)
```

```
{    /* início do corpo da função */
```

```
}    /* término do corpo da função */
```

Uma função deve conter :

- Um *header* que consiste do nome da função
- Uma lista de argumentos entre parênteses.
- Um bloco de instruções delimitado por chaves.

-- Estrutura básica de um programa

- O nome da função, os parênteses e as chaves, são os únicos elementos obrigatórios de uma função.
- Os comentários podem aparecer em qualquer lugar de um programa, devendo ser colocados entre os delimitadores `/*` e `*/`
- Letras minúsculas e maiúsculas não são equivalentes em C.
- Note que cada expressão dentro do bloco deve terminar com um ponto-e-vírgula.

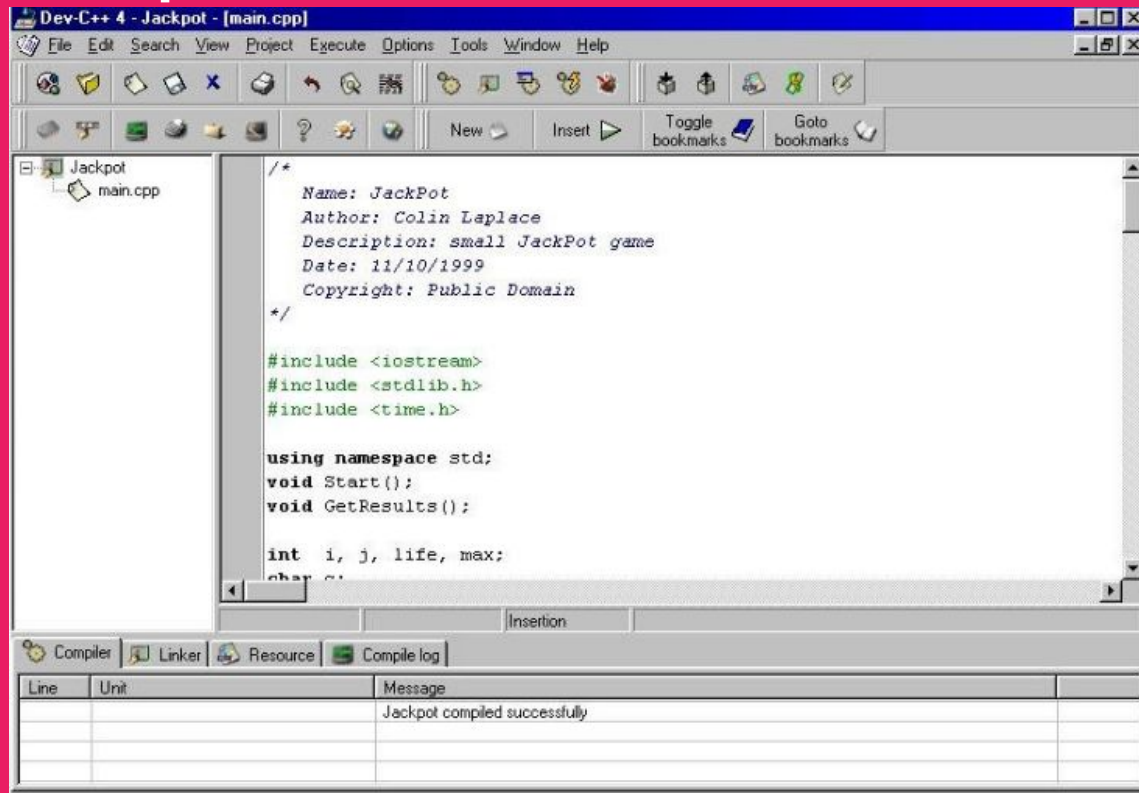
-Geração de Executável Editor (módulo fonte em C)

- Pré- processador (novo fonte expandido)
- Compilador (arquivo objeto)
- Lincador (executável)

-Dev C Ambiente de Desenvolvimento Linguagem C



- Tela Principal



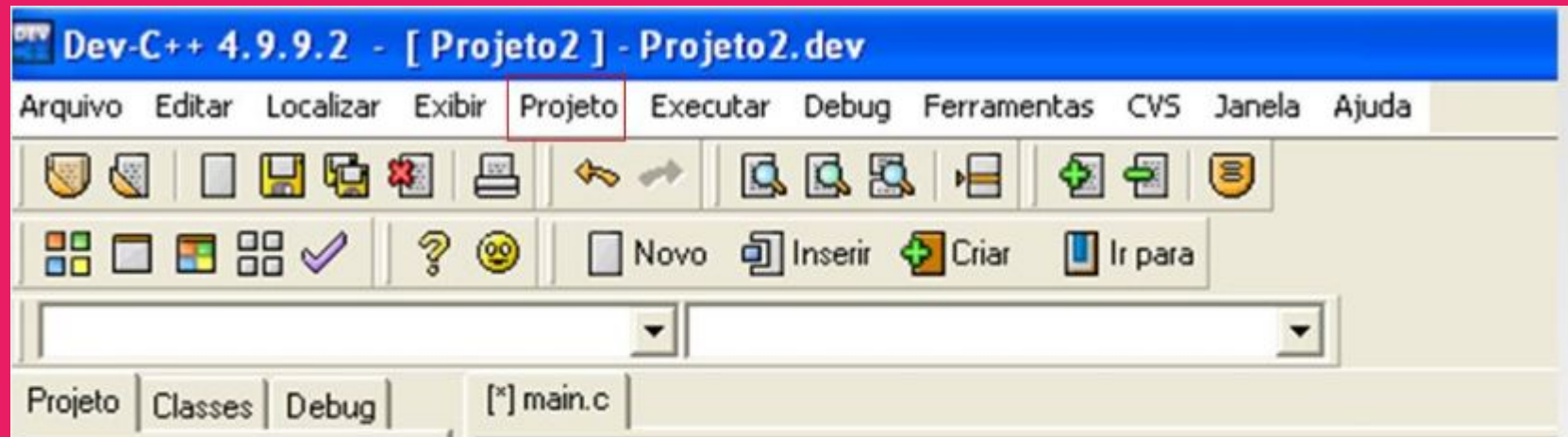
- Painel Principal



- Painel Principal



- Painel Principal



- Adicionar ou remover arquivo ao projeto, usaremos os ícones abaixo para fazer essas ações.

- Painel Principal



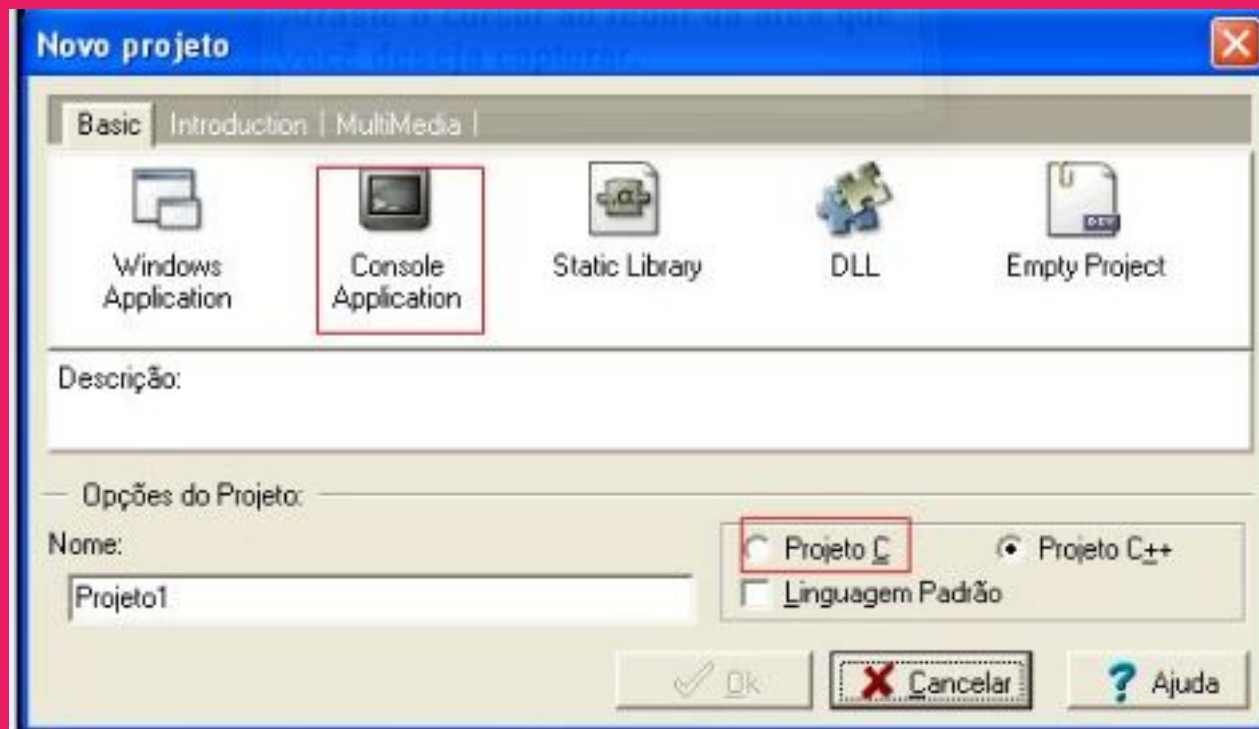
- Execução passo a passo. Nesse primeiro momento não vamos utilizar essa ferramenta.

- Painel Principal



- Algumas configurações de projeto. Nesse primeiro momento usaremos as configurações padrões.

- Meu Primeiro Programa Em C



- Meu Primeiro Programa Em C

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main(int argc, char *argv[])
{
    printf("Ola Mundo!");
    system("PAUSE");
    return 0;
}
```

- Inclui as bibliotecas que contêm as funções de entrada, saída e comando "System".

- Meu Primeiro Programa Em C

- Toda função em C retorna algo e recebe algo como parâmetro.
- Void significa vazio.
- Por padrão a função “main” é executada no início do programa;

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("Ola Mundo!");
    system("PAUSE");
    return 0;
}
```

- Meu Primeiro Programa Em C

- Imprime na tela “Olá Mundo!”.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("Ola Mundo!");
    system("PAUSE");
    return 0;
}
```

- Meu Primeiro Programa Em C

- Essa mensagem serve para travar o console do DOS.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("Ola Mundo!");
    system("PAUSE");
    return 0;
}
```

- Meu Primeiro Programa Em C

- Retorna 0.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    printf("Ola Mundo!");
    system("PAUSE");
    return 0;
}
```

- Operadores

- Atribuição (=)

- Aritméticos

- * e /

- %

+ e -

- Operadores Relacionais

== igual

!= diferente de

> maior que

< menor que

>= maior ou igual

<= menor ou igual

- Operadores

- Operadores Lógicos

- && (e)
- || (ou)
- ! (não)

- Condicionais if... else...

```
if (expressão) {  
    comandos;  
}
```

```
if (expressão) {  
    comandos;  
} else {  
    comandos;  
}
```

```
#include <stdio.h>  
main() {  
    int idade = 15;  
    if (idade < 18) {  
        printf("Invalida\n");  
    }  
    system("pause");  
}
```

```
#include <stdio.h>  
main() {  
    int idade = 21;  
    if (idade < 18) {  
        printf("Invalida\n");  
    } else {  
        printf("OK\n");  
    }  
    system("pause");  
}
```

- Condicionais Par ou Ímpar

```
#include <stdio.h>
main() {
    int a;
    printf("Digite um numero inteiro: ");
    scanf("%d", &a);
    if (a % 2 == 0) {
        printf("O valor eh par.\n");
    } else {
        printf("O valor eh impar.\n");
    }
    system("pause");
}
```


- Controle de Repetição : Exemplo Média de 2 Números

```
#include<stdio.h>

float media2(float a, float b) {
    return ((a + b) / 2.0);
}

int main() {
    float num_1, num_2, media;
    puts("Digite dois numeros:");
    scanf("%f %f", &num_1, &num_2);
    media = media2(num_1, num_2);
    printf("\nA media destes numeros eh %f", media);
    system("pause");
}
```

- Controle Repetição : Exemplo Soma

```
#include <stdio.h>

float soma(float a, float b) {
    return a + b;
}

int main() {
    float n1, n2;
    printf("Digite um numero: ");
    scanf("%f", &n1);
    printf("Digite outro numero: ");
    scanf("%f", &n2);
    printf("Soma: %f\n", soma(n1, n2));
    system("pause");
    return 0;
}
```

Plano de Estudos

1 – Lógica de Programação e Algoritmos

2 – HTML, CSS e JavaScript

3 – Banco de Dados MySQL e Linguagem SQL

4 – Orientação a Objetos (opcional, mas extremamente recomendável)



Slide

@yngridmonike



DÚVIDAS





OBRIGADA !