Introdução à Computação Variáveis e Entrada/Saída

Prof. Dr. Marcos Paulino Roriz Junior (marcosroriz@ufg.br)























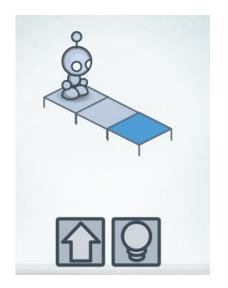


Nível de Abstração

Lighbot (Robô)

- **Pseudocódigo**
- Linguagem de Programação

- **Assembly**
- Linguagem de Máquina (010101)





Fonte: https://www.flickr.com/photos/christiaancolen/20445410340

























Algoritmos

- Linguagem natural → português.
- Diagramas de fluxo → gráfica.
- 3. Pseudocódigo: intermediário
- 4. Linguagem de programação → código.























Linguagem natural

Obtenha 3 valores de pH em pontos diferentes. Calcule a média dos mesmos. Verifique se é maior que 7, se for o pH é básico. Caso contrário, o pH é ácido.























Fluxograma

Símbolo utilizado para indicar o início e o fim do algoritmo.
Símbolo que permite indicar o sentido do fluxo de dados. Serve exclusivamente para conectar os símbolos ou blocos existentes.
Símbolo utilizado para indicar cálculos e atribuições de valores.
Símbolo utilizado para representar a entrada de dados.
Símbolo utilizado para representar a saída de dados.
Símbolo utilizado para indicar que deve ser tomada uma decisão, apontando a possibilidade de desvios.

























	Símbolo utilizado para indicar o início e o fim do algoritmo.	
	Símbolo que permite indicar o sentido do fluxo de dados. Serve exclusivamente para conectar os símbolos ou blocos existentes.	
	Símbolo utilizado para indicar cálculos e atribuições de valores.	
	Símbolo utilizado para representar a entrada de dados.	
	Símbolo utilizado para representar a saída de dados.	
\Diamond	Símbolo utilizado para indicar que deve ser tomada uma decisão, apontando a possibilidade de desvios.	

Obtenha 3 valores de pH em pontos diferentes. Calcule a média dos mesmos. Verifique se é maior que 7, se for o pH é básico. Caso contrário, o pH é ácido.

















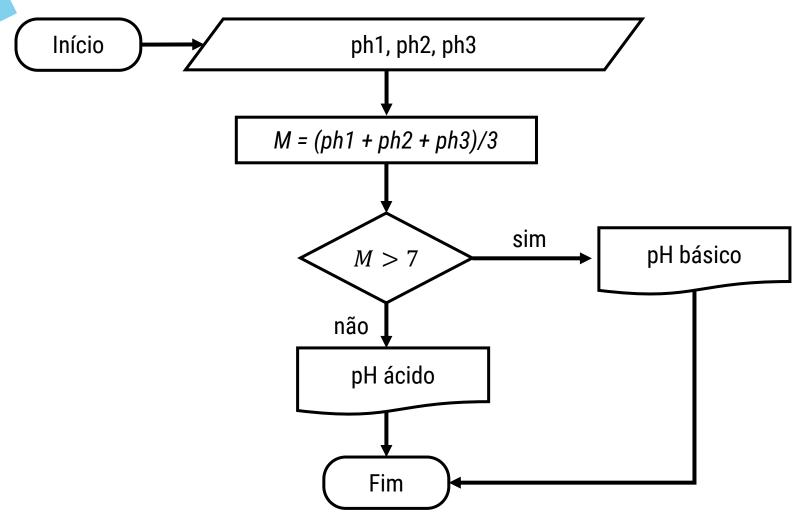








Fluxograma



























Fluxograma Ferramenta Lucidchart (https://lucidchart.com)























Lucidchart https://lucidchart.com

















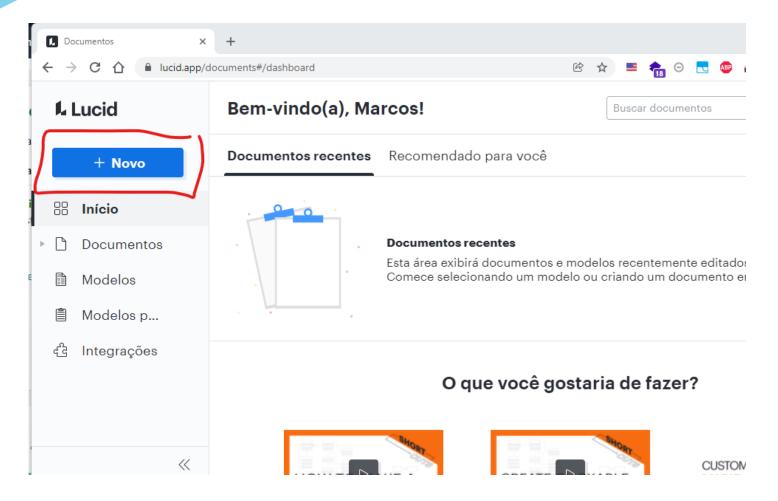








Lucidchart Clicar em novo documento



















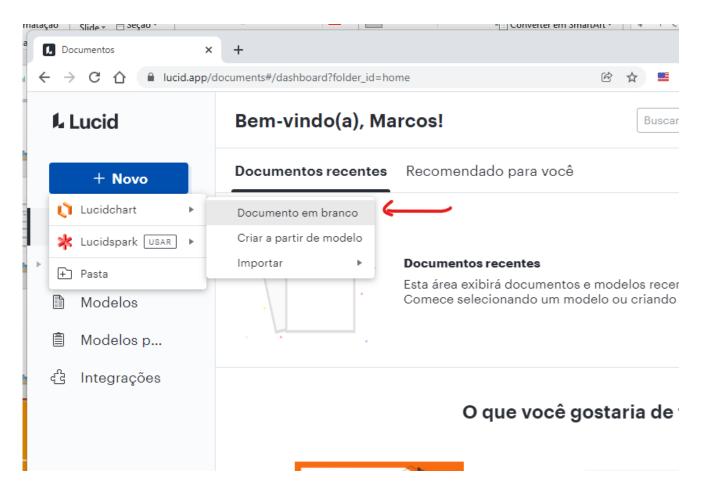








Ludichart Criar documento em branco



















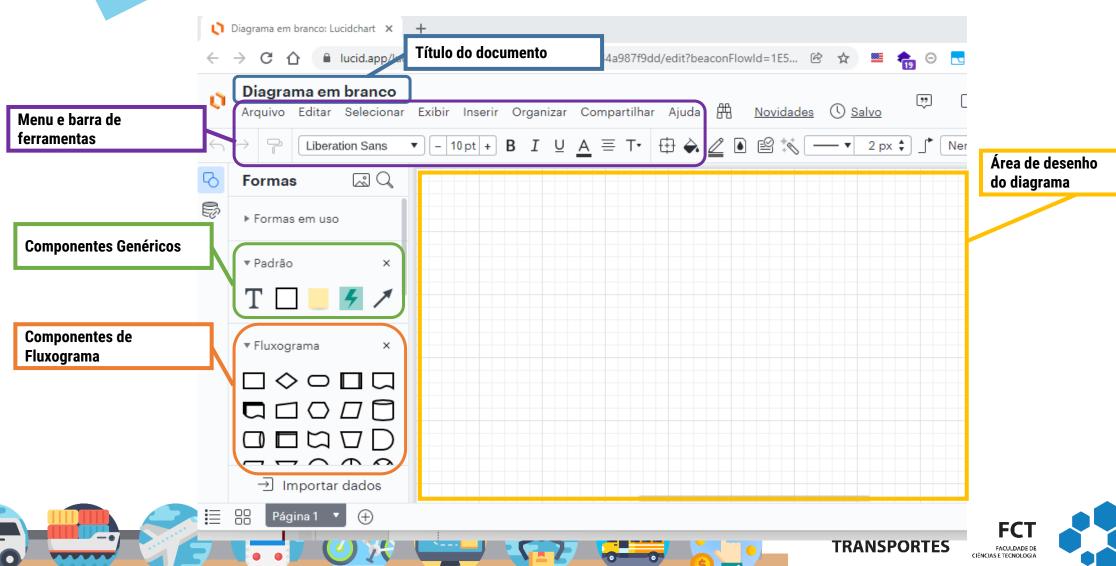






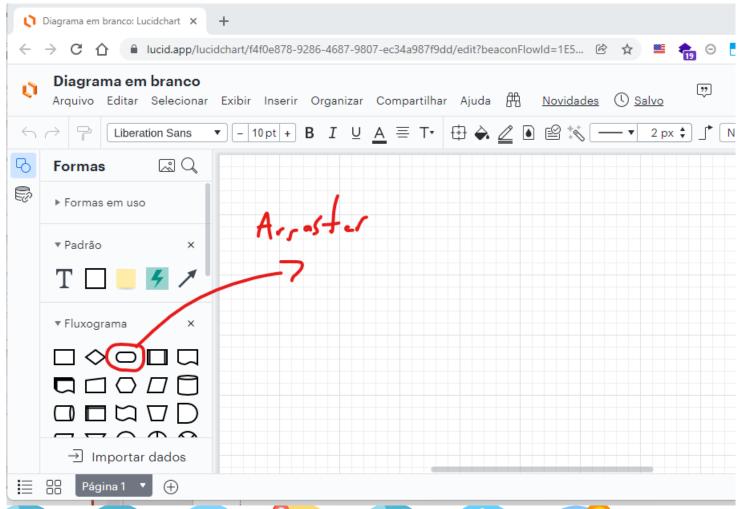


Lucidchart Tela do aplicativo





Lucidchart Adicionando elementos















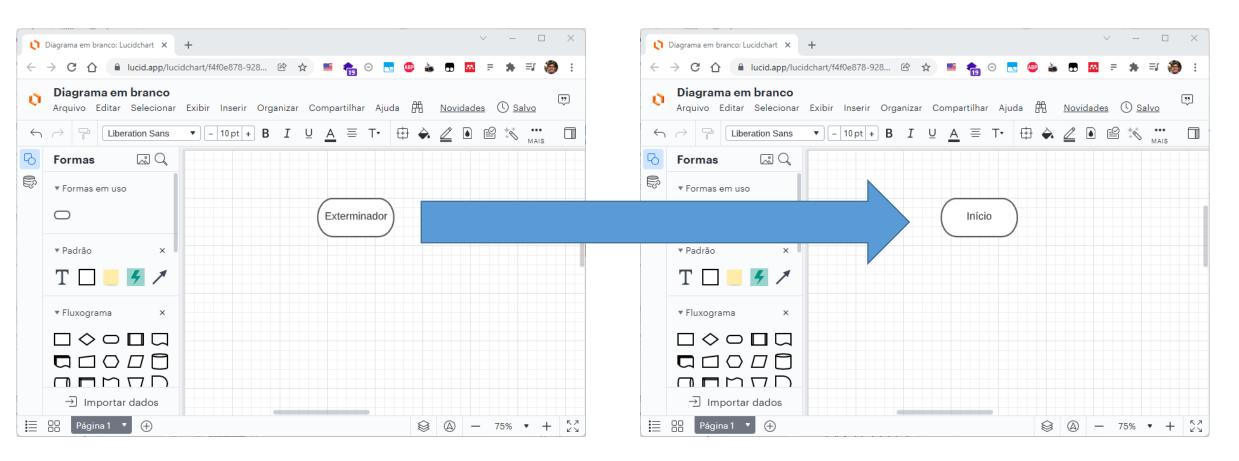








Lucidchart Clica duas vezes no componente para editar o texto



















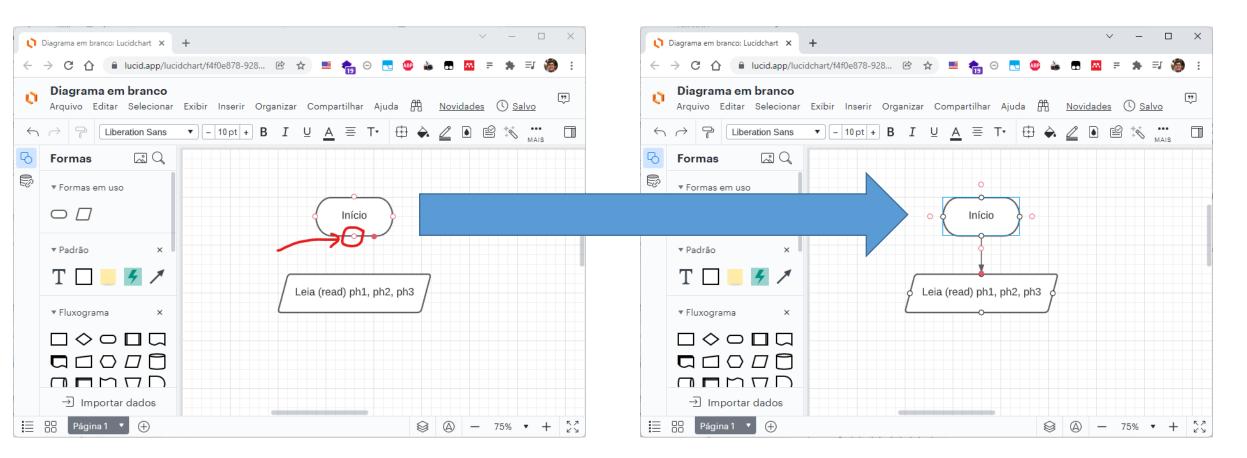








Ludichart Adicionar elemento de fluxo

















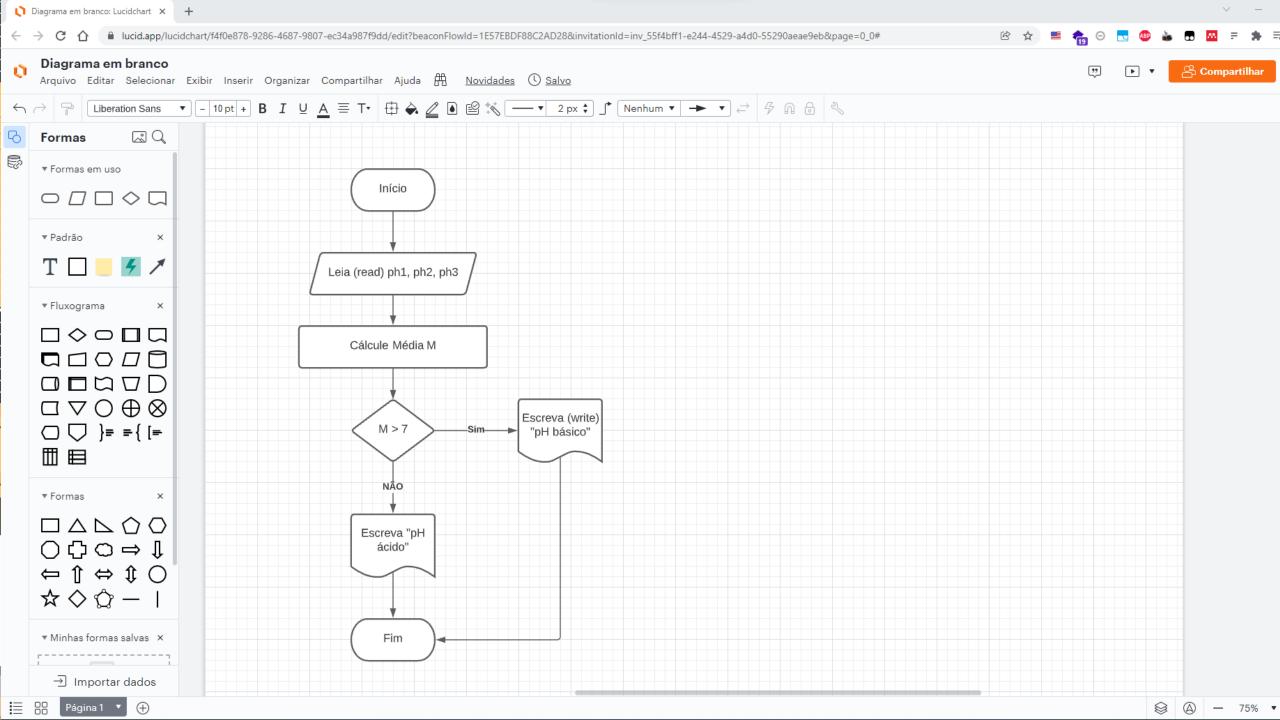




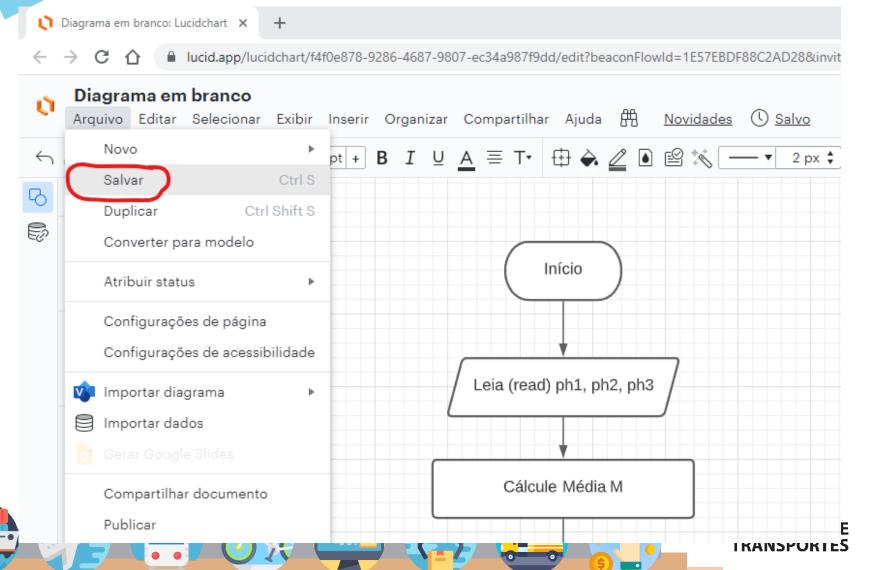








Lucidchart Arquivo -> Salvar



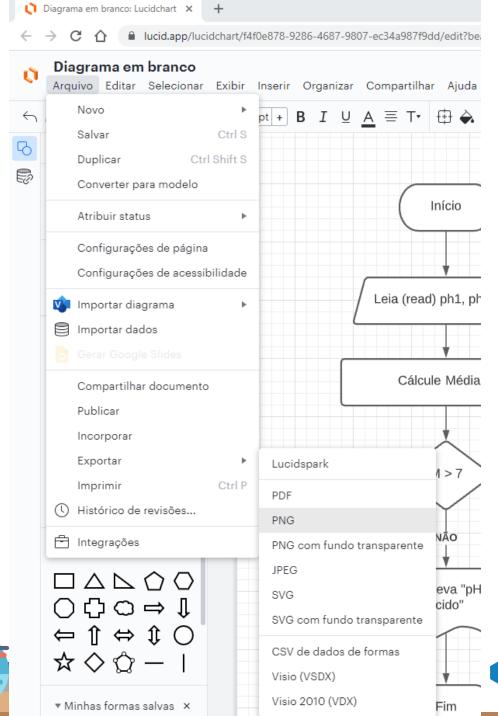




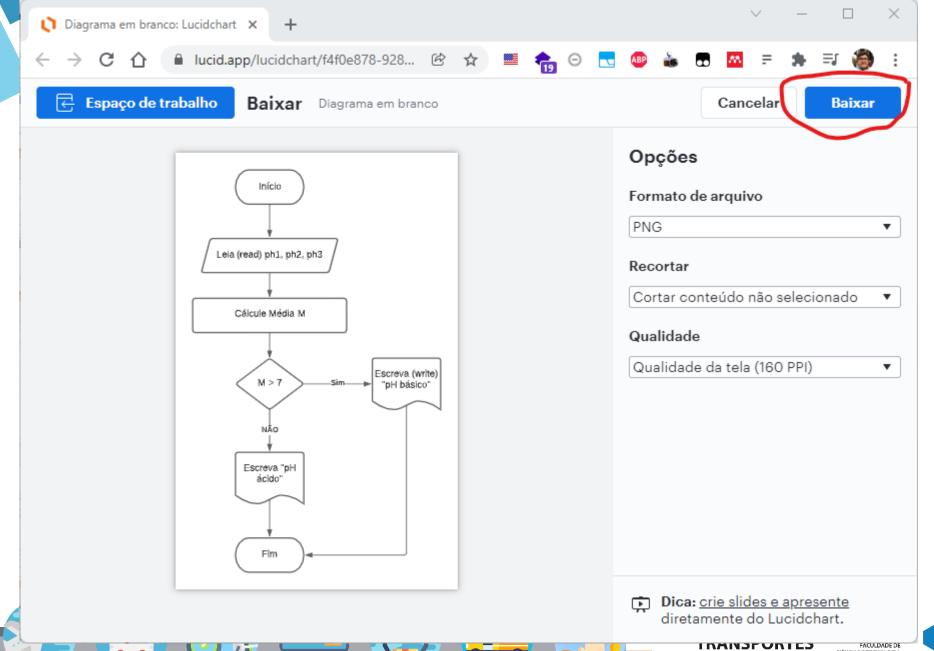


Para exportar o diagrama:

- Arquivo
- Exportar
- Escolha o formato
 - PNG
 - PDF
 - JPEG

































Pseudocódigo



























- Linguagem intermediária
- Próxima da linguagem de programação
 - Pode ser facilmente codificada em programa

























- Representação informal para algoritmos
- 1. Alto nível de abstração
- 2. Informal porém similar às linguagens de programação reais
- 3. Variáveis + notação matemática + estruturas de controle















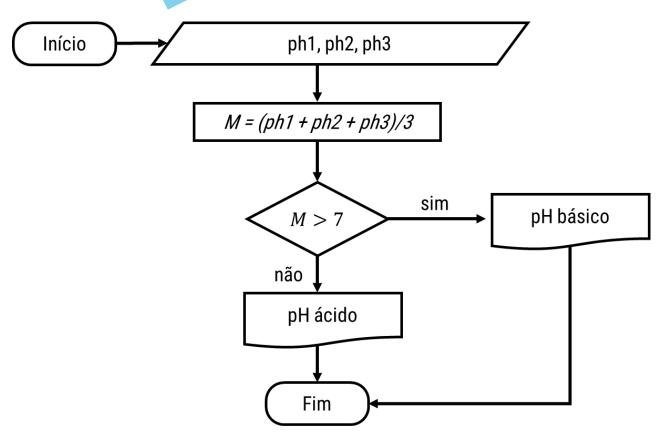








Pseudocódigo



ALGORITMO

- 1. DECLARE ph1, ph2, ph3, M NUMÉRICO
- 2. ESCREVA "DIGITE OS TRÊS VALORES"
- 3. LEIA ph1, ph2, ph3
- 4. $M \leftarrow (ph1 + ph2 + ph3) / 3$
- 5. SE M > 7
- 6. ENTÃO ESCREVA "ph Básico"
- 7. SENÃO ESCREVA "ph Ácido"

FIM_ALGORITMO























Pseudocódigo vs Programa C

ALGORITMO

- 1. DECLARE ph1, ph2, ph3, M NUMÉRICO
- 2. ESCREVA "DIGITE OS TRÊS VALORES"
- 3. LEIA ph1, ph2, ph3
- 4. $M \leftarrow (ph1 + ph2 + ph3) / 3$
- 5. SE M > 7
- 6. ENTÃO ESCREVA "ph Básico"
- 7. SENÃO ESCREVA "ph Ácido"

FIM_ALGORITMO

```
1. #include <stdio.h>
2. int main() {
3.
      float ph1, ph2, ph3, m;
      printf("Digite os valores");
      scanf("%f %f %f", &ph1, &ph2, &ph3);
5.
      m = (ph1 + ph2 + ph3) / 3;
6.
      if (m > 7) {
           printf("ph Básico");
8.
      } else {
9.
          printf("ph Ácido");
10.
11.
12.}
```





















Pseudocódigo vs Programa Python

ALGORITMO

- 1. DECLARE ph1, ph2, ph3, M NUMÉRICO
- 2. ESCREVA "DIGITE OS TRÊS VALORES"
- 3. LEIA ph1, ph2, ph3
- 4. $M \leftarrow (ph1 + ph2 + ph3) / 3$
- 5. SE M > 7
- 6. ENTÃO ESCREVA "ph Básico"
- SENÃO ESCREVA "ph Ácido"

FIM_ALGORITMO

```
1. print("Digite os valores")
2. ph1 = float(input())
3. ph2 = float(input())
4. ph3 = float(input())
5. m = (ph1 + ph2 + ph3) / 3
6. if m > 7:
   print("ph Básico")
8. else:
      print("ph Ácido")
```























Variáveis



















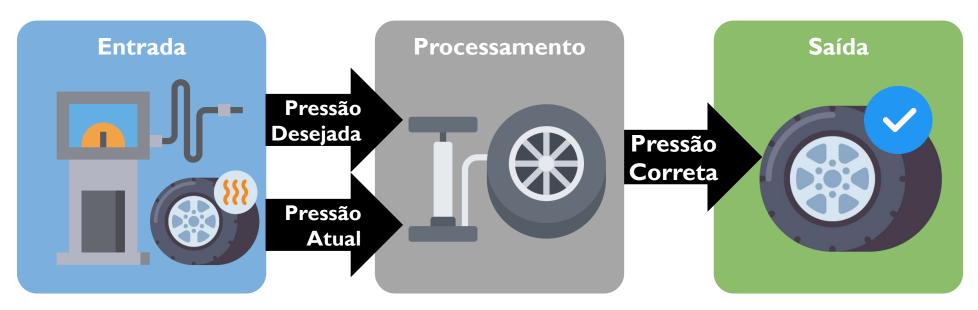






Variável: Motivação

- Programa permite que o usuário escolha um valor
- Deve estar preparado para lidar com o valor (variável) que o usuário escolheu

























- Pode ser usada internamente no programa
- Para representar dados sendo processados
- "Variável representa uma posição de memória, que possui nome e tipo e seu conteúdo pode variar ao longo do tempo, durante a execução de um programa"
- Embora uma variável possa assumir diferentes valores, ela só pode armazenar um valor a cada instante.















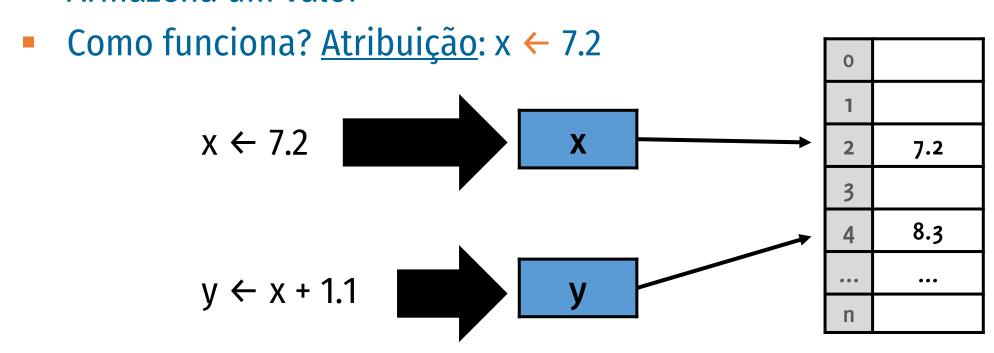






Pseudocódigo: Variáveis

- Local (célula) de memória identificada por um nome.
- Armazena um valor



















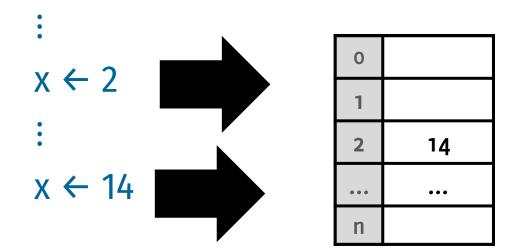
NGENHARIA DE TRANSPORTES





Pseudocódigo: Variáveis

- Local (célula) de memória identificada por um nome.
- Variar o valor armazenado ao longo da execução.



























- Devemos informar ao computador que iremos utilizar uma variável
- Declaração define o nome e o tipo da variável (tipo = valores que ela pode armazenar)
- FORMA: DECLARE ____ TIPO

Exemplos:

- DECLARE X, Y NUMÉRICO
- DECLARE A LÓGICO
- DECLARE NOME CARACTERES























Tipo de Dados/Variáveis

Principais tipos

Tipo	Descrição	Exemplo de valor
NUMÉRICO	Representam números, pode ser dividido em inteiros e reais	10 -42 20.443
LÓGICOS	Possuem apenas dois valores possíveis, verdadeiro ou valso	VERDADEIRO FALSO
CARACTERES	Também chamado de literais, este tipo representam os caracteres (letras)	'o' "marcos" "marcos1900"

Variáveis

- Os caracteres permitidos são: os números, as letras maiúsculas, as letras minúsculas e o caractere sublinhado.
- O primeiro caractere deve ser sempre uma letra ou o caractere sublinhado.
- Não são permitidos espaços em branco e caracteres especiais (@, \$, +, -, %, !).
- Não podemos usar as palavras reservadas nos identificadores, ou seja, palavras que pertençam à linguagem de programação.























Exemplos: DECLARE A, a NUMÉRICO DECLARE Bb, bB NUMÉRICO

- a é diferente de A
- Bb é diferente de bB























Pseudocódigo: Instruções

- Sequência de símbolos que denota um valor ou ação
- Símbolos: variáveis, constantes e operadores
- Ações: Entrada, Saída, Condicionais, Repetições
- Exemplos:

$$x \leftarrow 7.2$$

 $y \leftarrow 8.3$
 $r \leftarrow (x + y)$
ESCREVA $\pi \times r^2$

Programa: sequência de instruções (expressões)

















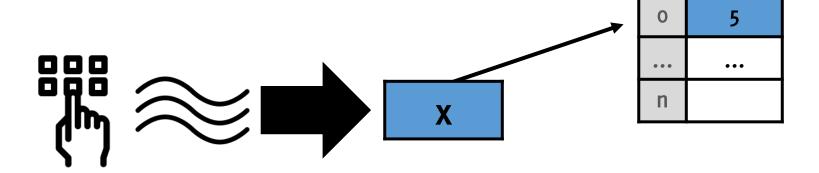




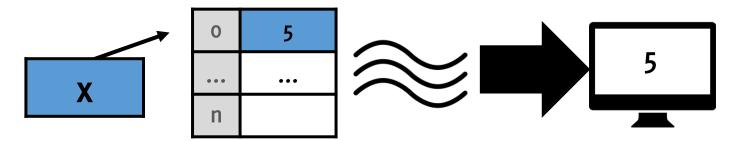


Pseudocódigo: Entrada e Saída

Entrada: <u>LEIA</u> x (read)



Saída: <u>ESCREVA</u> x (write)





















NGENHARIA DE TRANSPORTES







Exemplo (Problema: Pneu)





Fonte: https://www.pedal.com.br/

Calibrar os pneus do carro deve ser uma tarefa cotidiana de todos os motoristas. Para isto, os postos de gasolina possuem uma bomba de ar.

A maioria das bombas atuais são eletrônicas, permitindo que o motorista indique a pressão desejada num teclado.

Ao ser ligada ao pneu, a bomba primeiro lê a pressão atual e calcula a diferença de pressão entre a desejada e a lida.

Com esta diferença ela esvazia ou enche o pneu para chegar na pressão correta.

Sua ajuda foi requisitada para desenvolver o programa da próxima bomba da SBC – Sistemas de Bombas Computadorizadas































Fonte: https://www.pedal.com.br/

Tarefa

Escreva um programa que, dada a pressão desejada digitada pelo motorista e a pressão do pneu lida pela bomba, indica a diferença entre a pressão desejada e a pressão lida.

Exemplo (Problema: Pneu)

Entrada

- A primeira linha da entrada contém um inteiro N que indica a pressão desejada pelo motorista (1 ≤ N ≤ 40).
- A segunda linha contém um inteiro M que indica a pressão lida pela bomba $(1 \le M \le 40)$.

Saída

Seu programa deve imprimir uma única linha, contendo a diferença entre a pressão desejada e a pressão lida.



















ENGENHARIA DE











CALIBRADOR DE POSTO DE GASOLINA

Fonte: https://www.pedal.com.br/

Entrada	Saída
30 18	12

Exemplo (Problema: Pneu)

Entrada	Saída
27 27	0

Entrada	Saída
27	-3
30	































CALIBRADOR DE POSTO DE GASOLINA

Fonte: https://www.pedal.com.br/

Entrada	Saída
30 18	12

Entrada	Saída
27 27	0

Entrada	Saída
27	-3
30	

Entrada

- A primeira linha da entrada contém um inteiro N que indica a pressão desejada pelo motorista (1 ≤ N ≤ 40).
- A segunda linha contém um inteiro M que indica a pressão lida pela bomba (1 ≤ M ≤ 40).

Saída

 Seu programa deve imprimir uma única linha, contendo a diferença entre a pressão desejada e a pressão lida.































Fonte: https://www.pedal.com.br/

Algoritmo

- 1. DECLARE N, M, P NUMÉRICO
- 2. LEIA N
- 3. LEIA M
- 4. $P \leftarrow N M$
- 5. ESCREVA P
 FIM_ALGORITMO

Entrada

- A primeira linha da entrada contém um inteiro N que indica a pressão desejada pelo motorista (1 ≤ N ≤ 40).
- A segunda linha contém um inteiro M que indica a pressão lida pela bomba (1 ≤ M ≤ 40).

Saída

 Seu programa deve imprimir uma única linha, contendo a diferença entre a pressão desejada e a pressão lida.

























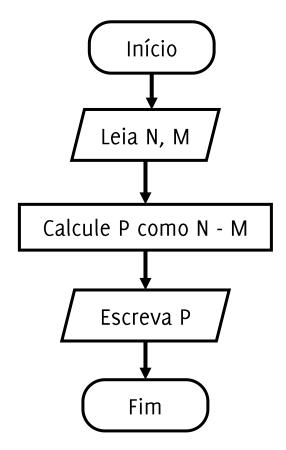




Fonte: https://www.pedal.com.br/

Algoritmo

- 1. DECLARE N, M, P NUMÉRICO
- 2. LEIA N
- 3. LEIA M
- 4. $P \leftarrow N M$
- 5. ESCREVA P
 FIM_ALGORITMO



























Exemplo: (Problema Tomadas)



Fonte: Public Domain Vectors

- A Olimpíada Internacional de Informática (IOI, no original em inglês) é a mais prestigiada competição de programação para alunos de ensino médio; seus aproximadamente 300 competidores se reúnem em um país diferente todo ano para os dois dias de prova da competição.
- Naturalmente, os competidores usam o tempo livre para acessar a Internet, programar e jogar em seus notebooks, mas eles se depararam com um problema: o saguão do hotel só tem uma tomada.



























Fonte: Public Domain Vectors

- Felizmente, os quatro competidores da equipe brasileira da IOI trouxeram cada um uma régua de tomadas, permitindo assim ligar vários notebooks em uma tomada só; eles também podem ligar uma régua em outra para aumentar ainda mais o número de tomadas disponíveis.
- No entanto, como as réguas têm muitas tomadas, eles pediram para você escrever um programa que, dado o número de tomadas em cada régua, determina quantas tomadas podem ser disponibilizadas no saguão do hotel.

























Exemplo: (Problema Tomadas)



Fonte: Public Domain Vectors

Entrada

 A entrada consiste de uma linha com quatro inteiros positivos T_1, T_2, T_3, T_4, indicando o número de tomadas de cada uma das quatro réguas.

Saída

 Seu programa deve imprimir uma única linha contendo um único número inteiro, indicando o número máximo de notebooks que podem ser conectados num mesmo instante.





















Exemplo: (Problema Tomadas)



Fonte: Public Domain Vectors

Entrada	Saída
2 4 3 2	8
Entrada	Saída
6 6 6 6	21
Entrada	Saída
2 2 2 2	5





























Fonte: Public Domain Vectors

Entrada	Saída
2 4 3 2	8
Entrada	Saída
6 6 6 6	21
Entrada	Saída
2 2 2 2	5

Entrada

 A entrada consiste de uma linha com quatro inteiros positivos T_1, T_2, T_3, T_4, indicando o número de tomadas de cada uma das quatro réguas.

Saída

 Seu programa deve imprimir uma única linha contendo um único número inteiro, indicando o número máximo de notebooks que podem ser conectados num mesmo instante.



























Exemplo: (Problema Tomadas)

ALGORITMO

- DECLARE T1, T2, T3, T4, TOTAL NUMÉRICO
- LEIA T1, T2, T3, T4
- $TOTAL \leftarrow T1 + T2 + T3 + T4 3$
- **ESCREVA Total**

FIM_ALGORITMO



Fonte: Public Domain Vectors

Entrada

A entrada consiste de uma linha com quatro inteiros positivos T_1, T_2, T_3, T_4, indicando o número de tomadas de cada uma das quatro réguas.

Saída

Seu programa deve imprimir uma única linha contendo um único número inteiro, indicando o número máximo de notebooks que podem ser conectados num mesmo instante.

























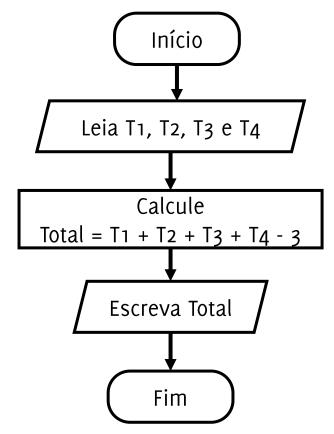


Fonte: Public Domain Vectors

ALGORITMO

- 1. DECLARE T1, T2, T3, T4, TOTAL NUMÉRICO
- 2. LEIA T1, T2, T3, T4
- 3. TOTAL \leftarrow T1 + T2 + T3 + T4 3
- 4. ESCREVA Total

FIM_ALGORITMO

























Linguagem C Concepção do programa e variáveis

















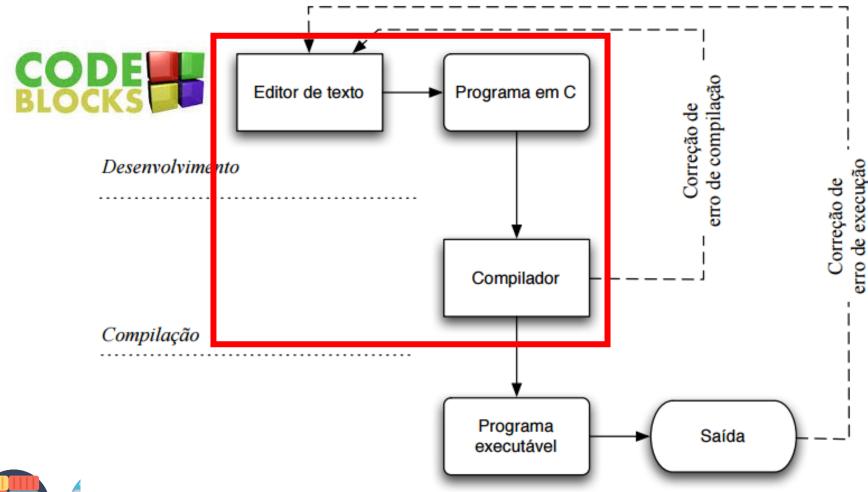








Ciclo de Desenvolvimento











- Ambiente de desenvolvimento Integrado
- Editor para linguagem C e C++
 - Processador de texto que facilita o desenvolvimento de programas em uma linguagem específica (C e C++).
- Compilador (traduz C -> Assembly -> Executável)
 - Embuti o compilador GCC
- Depurador (para tirar erros)
 - Parar execução do programa
 - Verificar valor de variáveis















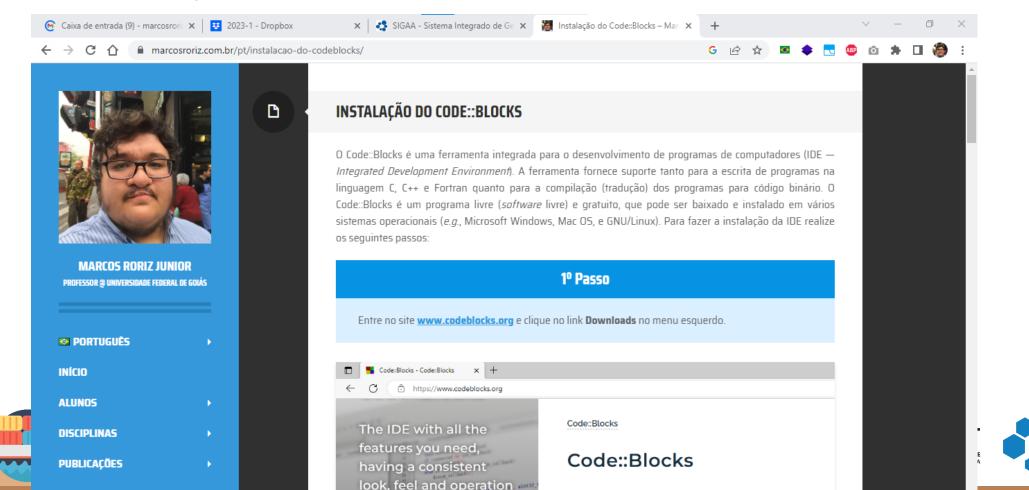




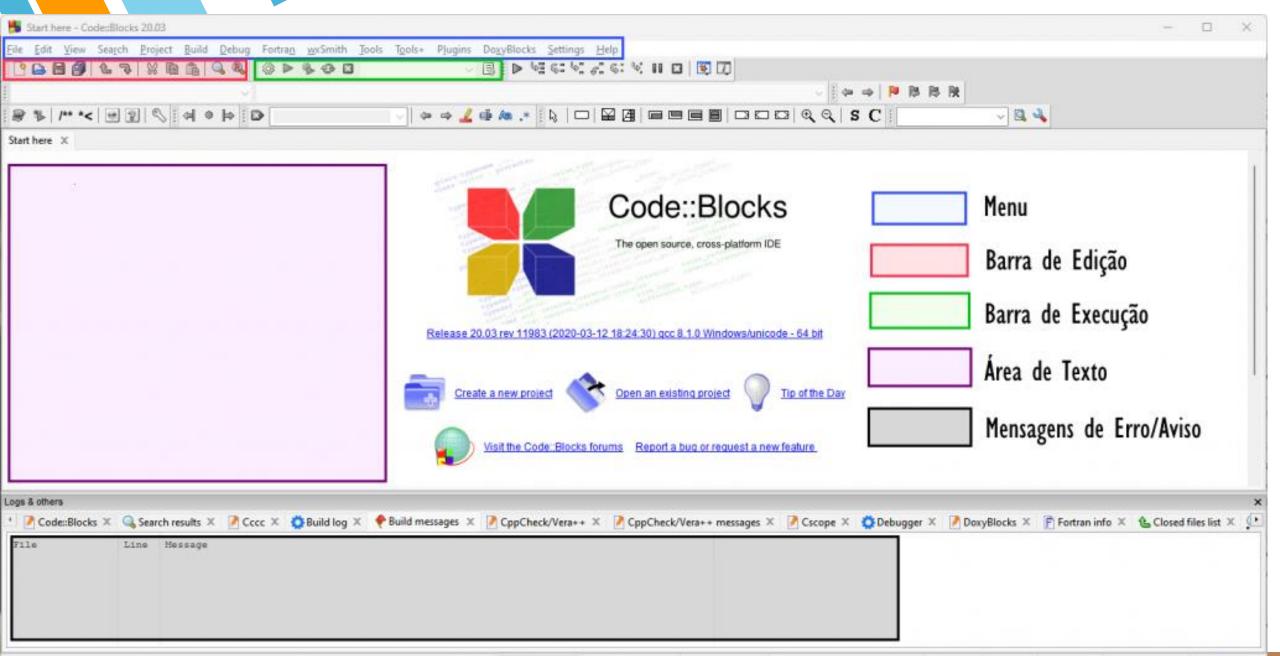


CODE::BLOCKS

Guia de instalação (https://marcosroriz.com.br/pt/introducao-a-computacao/)



Tela Code::Blocks



Item	Descrição
Menu	O menu permite acessar as funcionalidades da ferramenta. Por exemplo, o submenu File permite criar e salvar o seu programa C.
Barra de Edição	A barra de edição fornece atalhos para criar e salvar um programa C. Também inclui funcionalidades para procurar (find) e substituir textos.
Barra de Execução	A barra de execução fornece atalhos para compilar e para executar o programa sendo escrito.
Área de Texto	É a área onde você irá escrever seu programa.
Mensagens de Erro/Aviso	O painel de mensagens de erro/alerta irá lhe notificar de eventuais problemas no seu programa. Por exemplo, caso o seu programa possua algum erro de sintaxe, o compilador irá lhe notificar a linha e o que causou o erro neste painel.















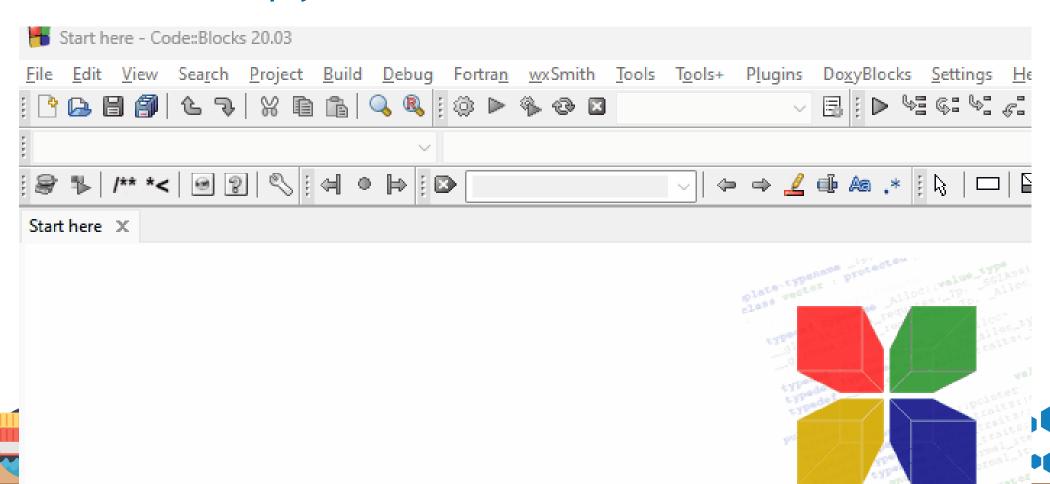






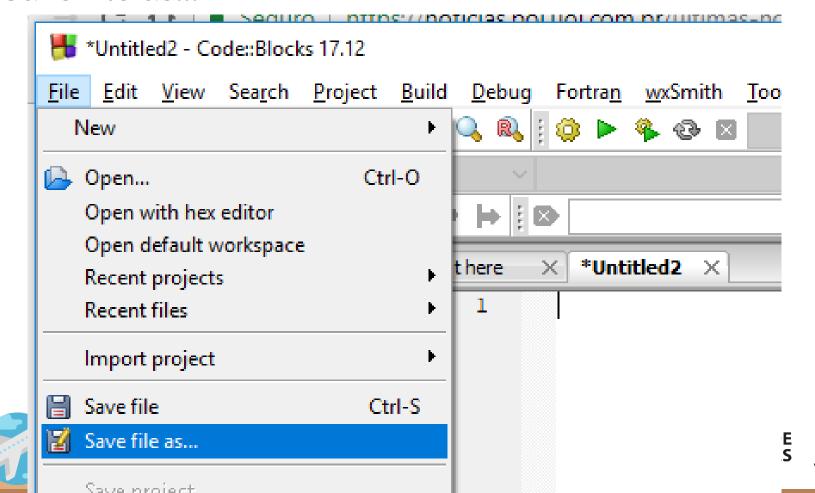
Primeiro programa (passo 1)

File -> New -> Empty file



Primeiro programa (passo 2)

File -> Save file as...

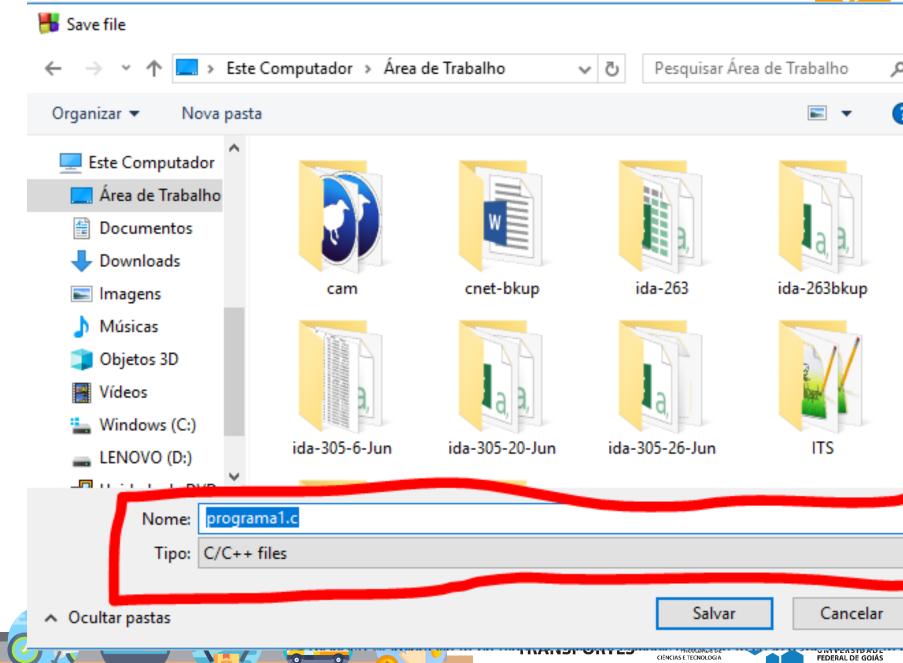




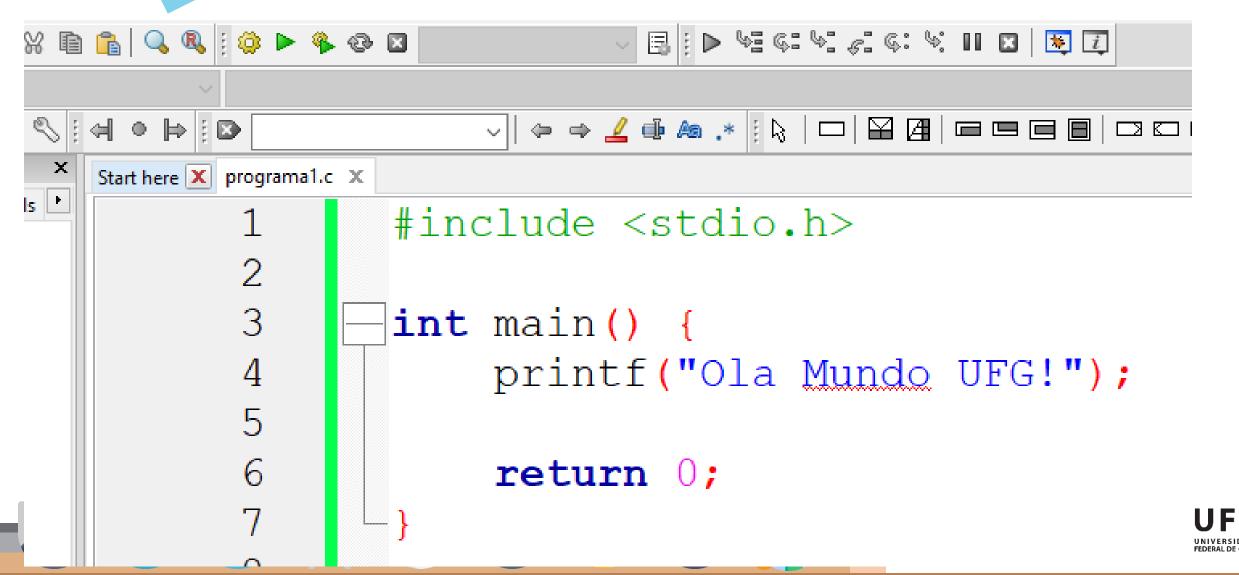


Primeiro programa (passo 3)

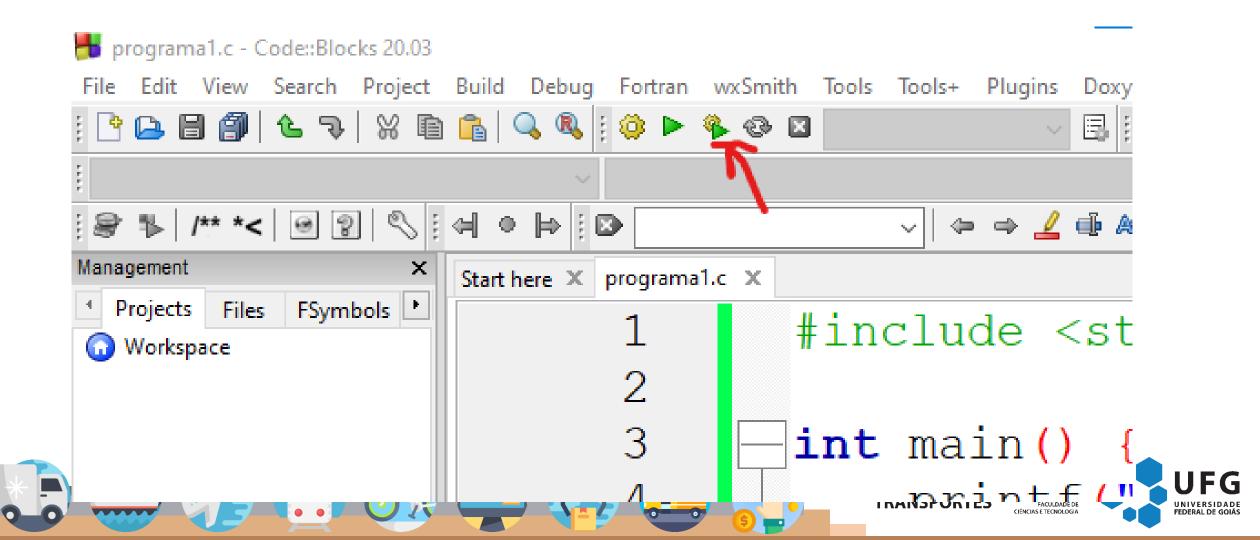
- programa1.c
- Arquivo C



Primeiro programa (passo 4): Código



Primeiro programa (passo 5): Compilar e Executar



Resultado!!!

C:\Users\marco\OneDrive\Documentos\C\programa1.exe

Ola Mundo UFG! Process returned 0 (0x0) execution time : 2 Press any key to continue.



























```
2 programa1.c ×

1 #include <stdio.h>
2
```

- stdio = standard input/output
- Será provavelmente incluída em todo programa C que faremos. Fornece informações ao compilador sobre a função de saída de dados printf, utilizada depois no programa.]
- Bibliotecas (funções feitas por outros programadores)























```
programa1.c X
   #include <stdio.h>
                                            Código do Programa
   int main() ▲{
       printf("Ola Mundo Eng. Transportes!!");
       return
```

- CHAVES { e } delimita as instruções do programa.
- Corpo do programa (principais instruções)





















```
2
3 pint main() {
```

- início
- Principal (main) ponto de início do programa
- Informa ao compilador onde o programa inicia de fato.
- É uma função da linguagem C. Por enquanto simplesmente memorizem que esta linha indica o inicio do programa.

























```
printf("Ola Mundo UFG!");
```

- printf = escreva
- Uma função de saída (implementada na stdio) que permite escrever a informação entre parêntese (parâmetro) na tela.
- Imprime o parâmetro informado na tela.
- Note que instruções terminam com ;





















return 0;

- retorne o valor 0 = <u>fim</u>
- Se o programa retornar 0 indica que foi executado com sucesso, senão ele retorna um valor diferente.













```
programa1.c X
        #include <stdio.h>
3
       int main() {
4
5
            printf("Ola Mur
6
            return 0;
```

Finaliza o corpo do programa























Variáveis



















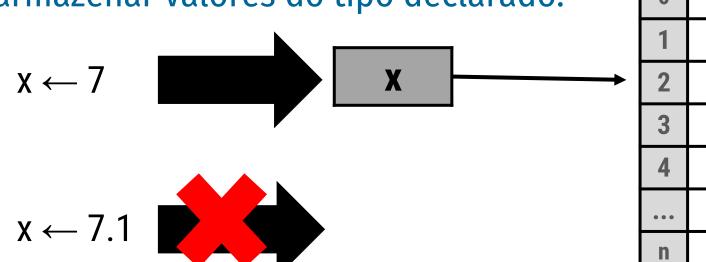






Programa

- int x;
- Declara uma variável inteira (tipo) chamada x.
- Só pode armazenar valores do tipo declarado.



























Programa Variáveis Declaração

- Toda variável tem:
 - um nome
 - um tipo de dado (inteiro, float, caractere, etc)
 - um valor
- Restrição de nomes (letras + números), mas não pode começar com números
- int x__;
- float y201456;
- char ufgGoias;





ipo nome























Tipo de Dados/Variáveis

Principais tipos

	DECLARAÇÃO EM C	Exemplo de valor
NUMÉRICO INTEIRO	int	30, 20, 0, -42
NÚMERO REAL	float double	42.44, -33.1, 0.1, 5.0
LÓGICOS	int	0 = falso 1 = verdadeiro
CARACTERES	char	'o', 'a' "marcos" "marcos1900"











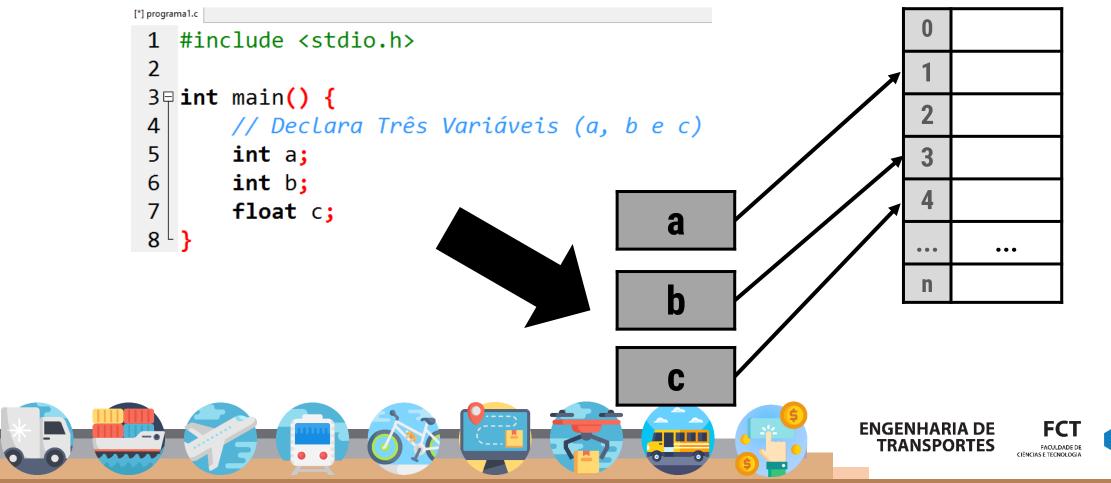






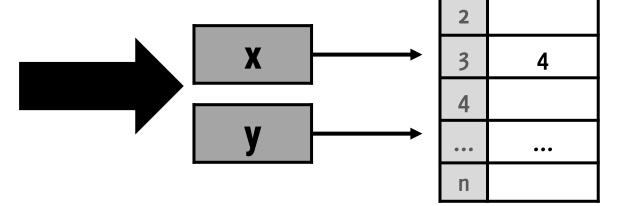
Programa variáveis tipo

Exemplo



Programa variáveis atribuição

- x = expr;
- Atribui o valor resultante da expressão a variável x.
- Tipos de expr e x devem concordar.
- O símbolo = significa atribuição (←) e não igualdade.
- Exemplos:
 - int x = 2;
 - int y = 4;
 - x = -1;

















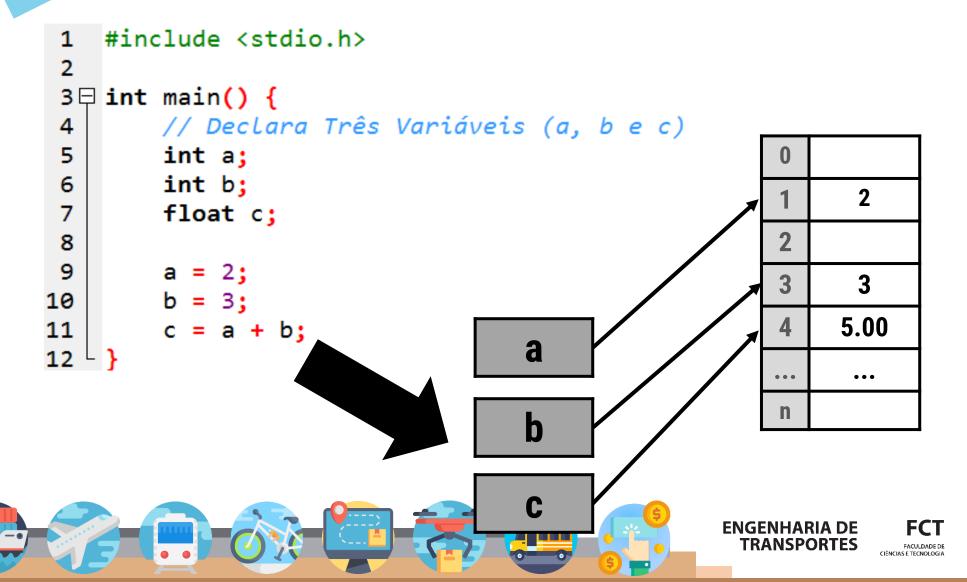








Atribuição



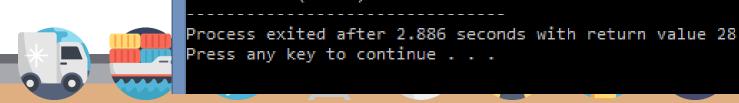
Escrevendo uma variável na tela

```
programa1.c primieroprograma.c
    #include <stdio.h>
 3 pint main() {
         // Declara Três Variáveis (a, b e c)
         int a = 2;
         int b = 3;
         float c = a + b;
         printf("0 valor de (a + b): %f", c);
10 <sup>∟</sup> }
                      5.000
                                               5.000
```



ENGENHARIA DE

```
programa1.c primieroprograma.c
    #include <stdio.h>
 3 □ int main() {
         // Declara Três Variáveis (a, b e c)
         int a = 2;
         int b = 3;
         float c = a + b;
 8
         printf("0 valor de (a + b): %f", c);
 9
10
                       5.000
             C:\Users\Marcos\Desktop\DevCpp\programa1.exe
 0 valor de (a + b): 5.000000
```







Saída = Escrita

- Saída: <u>printf</u> é a função utilizada para saída de dados de um programa em C.
- <u>escreva</u> em pseudocódigo.
- Exemplos:
 - printf("Texto a imprimir\n");
 - printf("Imprime o valor inteiro: %d \n", variável);























Saída = Escrita

- Permite escreve valores em um determinado formato
- printf = print (escreva) + f (formato)

printf ("formato", lista de variáveis);

- Quando não tenho variáveis?
- printf("Conteudo") => Conteudo
- Quando tenho variáveis:
- printf("0 total é = %d", var) => 0 total é 2























Tipo de Formatos

printf ("formato", lista de variáveis);

Formato de variáveis

%с	char
%d ou %i	int
%u	unsigned int
%f	float
%lf	double
%e	formato científico
% g	número real (float ou double)
%s	uma string (sequência de char)















ENGENHARIA DE TRANSPORTES





Tipos de formatos

```
printf ("formato", lista de variáveis);
```

- Especificar casas decimais =
- %.cf = onde c é o número de casas
- %.2f = 2 casas decimais
- Exemplo:
- float k = 3.14134134;
- printf("Valor de k = %.3f", k);























Saída = Escreva

 Saída: <u>printf</u> é a função utilizada para saída de dados de um programa em C.

```
#include <stdio.h>
 3 □ int main() {
        int num = 20;
4
        float k = 50;
 5
 6
        num = num + 30;
8
        k = k + num;
 9
        printf("Bem vindo ao meu programa!\n");
10
        printf("Imprimindo um inteiro: %d\n", num);
11
        printf("Imprimindo um float: %f\n", k);
12
13
```



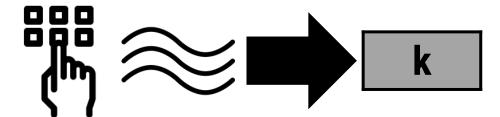


Entrada (Scanf) = LEIA

- Permite ler valores em um determinado formato
- scanf = scan (digitalizar) + f (formato)

scanf("formato", &lista de variáveis);

- int k;
- scanf("%i", &k)
- // valor digitado é armazenado em k

















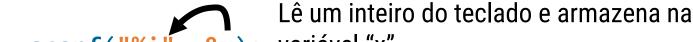






Entrada = LEIA

- Entrada: scanf é a função utilizada para entrada de dados provenientes do usuário.
- <u>LEIA</u> em pseudocódigo
- Exemplos:
 - int x;
 - float k;



- scanf("<mark>%i"</mark>, &x); variável "x"
- scanf("<u>%f"</u>, &k); Lê um real do teclado e armazena na variável "k"























ENTRADA (SCANF) = LEIA

- Múltiplo valores e formatos
- formato = "%i %f %i"

```
scanf("formato", &lista de variáveis);
```

- int k, n;
- float x;
- scanf("%i %f %i", &k, &x, &n);























Entrada e Saída

```
#include <stdio.h>
    int main() {
        int x;
        float k;
        float soma;
8
        printf("Digite o valor de x: ");
        scanf("%d", &x);
10
11
        printf("Digite o valor de k: ");
        scanf("%f", &k);
12
13
14
        soma = x + k;
15
        printf("A soma dos valores (x + k) = %f", soma);
16 L }
```

























```
#include <stdio.h>
 3 □ int main() {
         int x;
         float k;
         float soma;
 6
 8
         printf("Digite o valor de x: ");
         scanf("%d", &x);
10
11
         printf("Digite o valor de k: ");
12
         scanf("%f", &k);
13
14
         soma = x + k;
         printf("A soma dos valores (x + k) = %f", soma);
15
16
   C:\Users\Marcos\Desktop\DevCpp\programa2.exe
   Digite o valor de x: 5
   Digite o valor de k: 7.2
   A soma dos valores (x + k) = 12.200000
   Process exited after 4.912 seconds with return value 38
   Press any key to continue . . .
```





Variáveis e expressões

- Soma +
- Subtração —
- Multiplicação *
- Divisão /
- Resto da Divisão % (apenas entre inteiros)























Variáveis e expressões

- Soma +
- Subtração —
- Multiplicação *
- Divisão /
- Resto da Divisão %

 (apenas entre inteiros)

```
Digite dois numeros: 15

Soma de (x + y) = 22
Subtracao de (x + y) = 8
Multiplicacao (x * y) = 105
Divisao (x / y) = 2
Resto (x / y) = 1

Process exited after 14.16 seconds with return value 18
```

Press any key to continue . . .



Variáveis expressões

- int x = 20;
- int y = 3;
- int a = x + y * x;
- int b = y x * x y;
- int d = x / y;
- int e = x % y;























Conversão de tipos (Casting)

- Permite que uma variável de um tipo seja transformada para outro tipo com o MESMO VALOR
- Por quê? Para permitir que possamos usar operadores adequados
 Ex: Soma de dois conjuntos de caracteres
- N ← 32
- $M \leftarrow 28$
- R ← 32 / 5

Int

R = 6

Float

R = 6.4























Converter / Casting

- int x = 20;
- int y = 3;
- int d = ((float) x) / y;
- int e = x % y;























Programa de Temperatura

- ALGORITMO
- 2. DECLARE F, C NUMÉRICO
- 3. LEIA F
- 4. C <- (F 32) / 1.8
- 5. ESCREVA C
- 6. FIM

```
#include <stdio.h>

int main () {
    float fahrenheit;
    float celsius;

scanf("%f", &fahrenheit);
    celsius = (fahrenheit - 32) / 1.8;
    printf("Temp. Celsius: é %f", celsius);
}
```























Exemplo

 Escreva um algoritmo e um programa que lê uma medida em pés (número inteiro) e medida em polegadas (número inteiro) e imprime a medida correspondente em metros (número real). Um pé equivale a 30,48cm e uma polegada equivale a 2,54cm. Por exemplo:

Entrada:	Saída:
5 10	1.778





















9. Faça um programa que calcule e mostre a área de um triângulo. Sabe-se que: Área = (base * altura)/2.

ALGORITMO Solução:

ALGORITMO DECLARE base, altura, area NUMÉRICO LEIA base, altura area ← (base * altura)/2 ESCREVA area FIM ALGORITMO.









Próxima aula

- Algoritmos e C++
- Mais primitivas básicas
- Condicionais

























Bibliotecas

























Biblioteca

- math.h
- https://cplusplus.com/reference/cmath/
- Constantes M_PI, M_E
- pow(base, expoente)
- sqrt(x)

























Funções matemáticas

Função	Exemplo	Comentário
ceil	ceil(X)	Arredonda um número real para cima. Por exemplo, ceil(3.2) é 4.
cos	cos(X)	Calcula o cosseno de X (X deve estar representado em radianos).
exp	exp(X)	Obtém o logaritmo natural e elevado à potência X.
abs	abs(X)	Obtém o valor absoluto de X.
floor	floor(X)	Arredonda um número real para baixo. Por exemplo, floor(3.2) é 3.
log	log(X)	Obtém o logaritmo natural de X.
log10	log10(X)	Obtém o logaritmo de base 10 de X.
modf	z = modf(X,&Y)	Decompõe o número real armazenado em X em duas partes: Y recebe a parte fracionária e z, a parte inteira do número.
pow	pow(X,Y)	Calcula a potência de X elevado a Y.
sin	sin(X)	Calcula o seno de X (X deve estar representado em radianos).
sqrt	sqrt(X)	Calcula a raiz quadrada de X.
tan	tan(X)	Calcula a tangente de X (X deve estar representado em radianos).

























```
#include <stdio.h>
      #include <math.h>
      int main() {
 4
           // calcula raiz cúbica
 6
           float x, y;
           scanf("%f", &x);
8
9
           y = pow(x, (1.0 / 3.0));
           printf("%f", y);
10
11
           return 0;
12
13
14
```



















ENGENHARIA DE TRANSPORTES





Observação

As funções sin, cos e tan esperam receber argumentos no formato de radianos; para receberem argumentos em graus, siga o exemplo a seguir.

Exemplo para o cálculo do seno de um ângulo fornecido em graus e utilizando uma variável para o valor de π :

```
VALORPI = 3.1415;
scanf("%f%*c",&X); //X EM GRAUS
Y = SIN ((VALORPI * X) / 180);
```





















Biblioteca

```
here X programa1.c X
          #include <stdio.h>
          #include <locale.h>
          int main() {
               setlocale (LC ALL, "Portuguese");
               printf("Olá Mundo UFG!");
               return 0;
```



C:\Users\marco\OneDrive\Documentos\C\programa1.exe

Olá Mundo UFG! Process returned 0 (0x0) execution time : Press any key to continue.





















Obrigado!

Perguntas?

Marcos Roriz (marcosroriz@ufg.br)

ENGENHARIA DE TRANSPORTES

FACULDADE DE CIÊNCIAS E TECNOLOGIA





















Obrigado! Perguntas?

Marcos Roriz (marcosroriz@ufg.br)

ENGENHARIA DE TRANSPORTES

FCT FACULDADE DE CIÊNCIAS E TECNOLOGIA





















