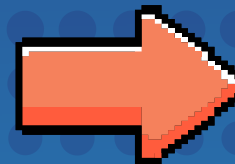


MARIANA GASPAR

Introdução à programação





O QUE É **PROGRAMAR**?

PROGRAMAR É DAR INSTRUÇÕES A UM COMPUTADOR PARA ELE FAZER ALGUMA COISA

MAS HÁ UM PROBLEMA...

COMPUTADORES NÃO FALAM PORTUGUÊS, NEM INGLÊS, NEM ESPANHOL.
ELES SÓ ENTENDEM ALGO MUITO BÁSICO (0 E 1).

OU SEJA, É COMO SE O COMPUTADOR FOSSE ALGUÉM QUE SÓ FALA UMA LÍNGUA SUPER ESTRANHA.





O QUE É **PROGRAMAR**?

PROGRAMAR É ENSINAR O COMPUTADOR COMO SE FOSSE UMA CRIANÇA MUITO LITERAL.

SE NÃO DISSERES TUDO:

- ELE NÃO ADIVINHA
- NÃO IMPROVISA
- NÃO PENSA SOZINHO

TEM DE TER PASSOS EXATOS – ISTO É UM **ALGORITMO**.





O QUE É UM **ALGORITMO**?

UMA SEQUÊNCIA DE PASSOS PARA **RESOLVER UM PROBLEMA**

IMAGINA FAZER UM CAFÉ, O ALGORITMO SERIA:

- COLOCAR ÁGUA
- AQUECER ÁGUA
- COLOCAR CAFÉ
- SERVIR

E ESTE É NECESSÁRIO EM QUALQUER **LINGUAGEM** DE PROGRAMAÇÃO.

exemplo

```
#INCLUDE <STDIO.H>
```

```
INT MAIN( )
```

```
{
```

```
    PRINTF("OLÁ TURMA!\n");
```

```
    RETURN 0;
```

```
}
```

ESTA É A FORMA COMO PEDIMOS AO COMPUTADOR
QUE MOSTRE "OLÁ TURMA" NO ECRA, USANDO C.



O que é uma linguagem
de programação?

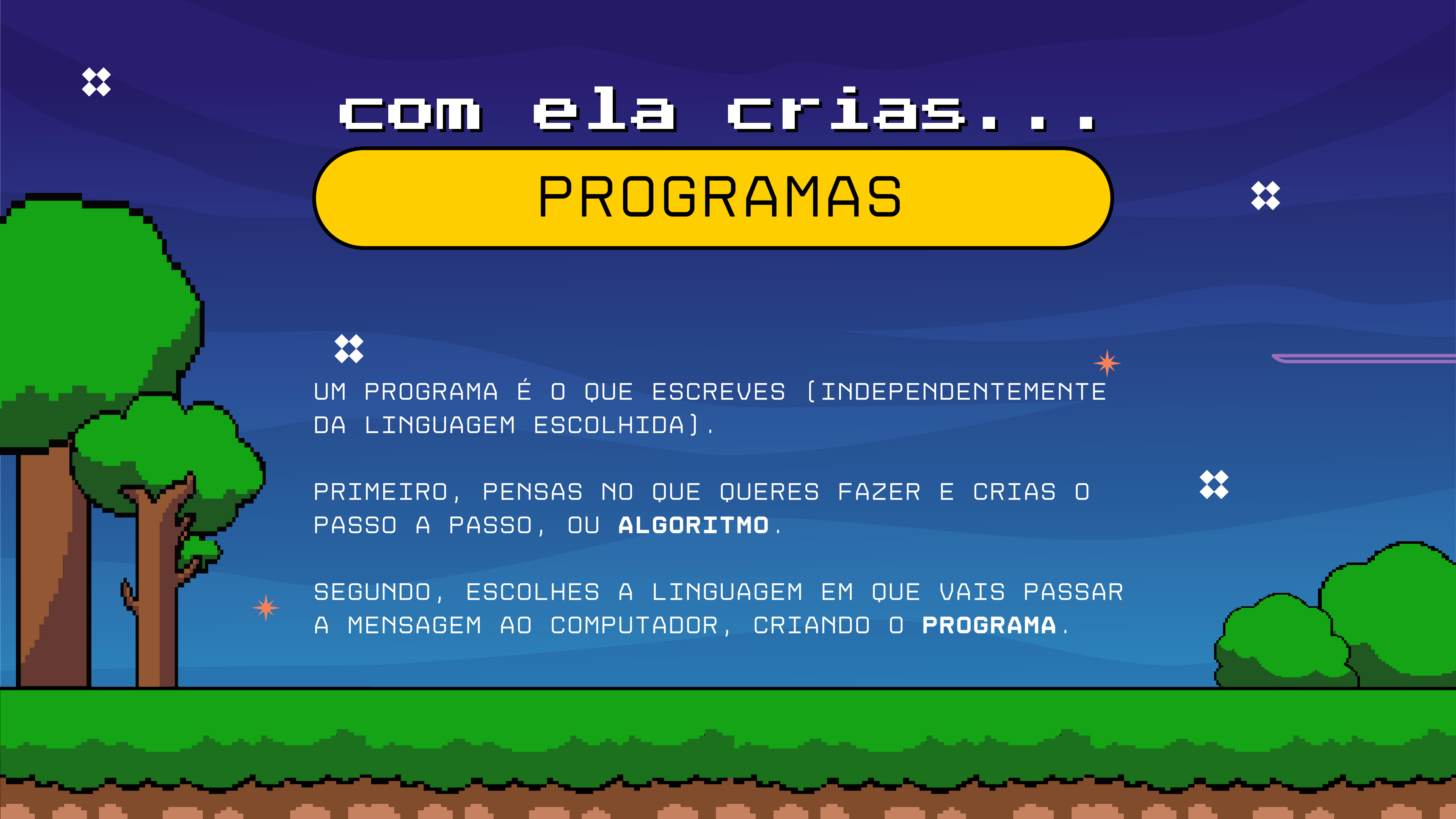


É O IDIOMA QUE
USAS PARA FALAR
COM O COMPUTADOR



C
PYTHON
JAVA
OUTROS...





com ela crias...

PROGRAMAS

UM PROGRAMA É O QUE ESCREVES (INDEPENDENTEMENTE DA LINGUAGEM ESCOLHIDA).

PRIMEIRO, PENSAS NO QUE QUERES FAZER E CRIAS O PASSO A PASSO, OU **ALGORITMO**.

SEGUNDO, ESCOLHES A LINGUAGEM EM QUE VAIS PASSAR A MENSAGEM AO COMPUTADOR, CRIANDO O **PROGRAMA**.



O QUE SÃO VARIÁVEIS?

VARIÁVEIS SÃO “CAIXAS” EM QUE GUARDAS COISAS DENTRO.
EXISTE TIPAGEM DINÂMICA E NÃO DINÂMICA.

NO CASO DA NÃO DINÂMICA, CADA VARIÁVEL SÓ RECEBE
COISAS DO MESMO TIPO.

IMAGINANDO EM SITUAÇÕES DO DIA A DIA:





O QUE SÃO VARIÁVEIS?

NO CASO DA DINAMICA PODEMOS GUARDAR QUALQUER TIPO DE CONTEUDO NA VARIÁVEL.

INTEIROS - GUARDA NÚMEROS SEM VÍRGULA (EXEMPLO: 2)

REAL - GUARDA NÚMEROS DECIMAIS (EXEMPLO: 2,5)

TEXTO - GUARDA CONJUNTOS DE CARACTERES (EXEMPLO: "OLÁ TURMA")

BOOLEANO - GUARDA OPÇÕES DE VERDADEIRO OU FALSO (EXEMPLO: IS_PRESENT?) **TRUE**

NO CASO DA TIPAGEM NÃO DINAMICA, NUMA VARIÁVEL DE INTEIROS NÃO CONSEGUIMOS ARMAZENAR TEXTO, PORQUE NÃO SÃO VARIÁVEIS DO MESMO TIPO.

Operadores



SÃO USADOS EM DIFERENTES
SITUAÇÕES EM PROGRAMAÇÃO

+

*

-

=

/

>



PODEMOS USAR EM OPERAÇÕES

❖ matemáticas

SOMA

$$\begin{array}{|c|c|c|} \hline 5 & + & 3 \\ \hline \end{array}$$

SUBTRAÇÃO

$$\begin{array}{|c|c|c|} \hline 4 & - & 2 \\ \hline \end{array}$$

DIVISÃO

$$\begin{array}{|c|c|c|} \hline 4 & / & 2 \\ \hline \end{array}$$

MULTIPLICAÇÃO

$$\begin{array}{|c|c|c|} \hline 4 & * & 2 \\ \hline \end{array}$$

PODEMOS USAR EM OPERAÇÕES

Lógicas

[DUAS OPÇÕES]

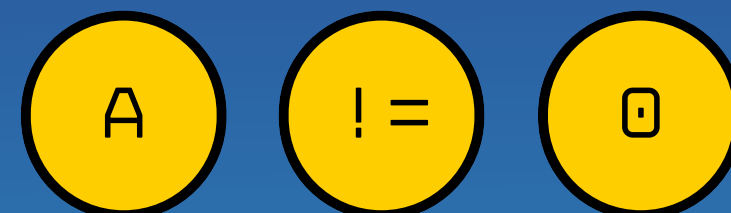
AND [E]



OR [OU]



DIFERENTE DE



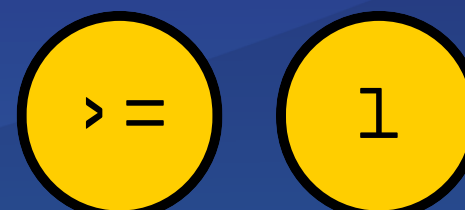
PODEMOS USAR EM OPERAÇÕES

Lógicas

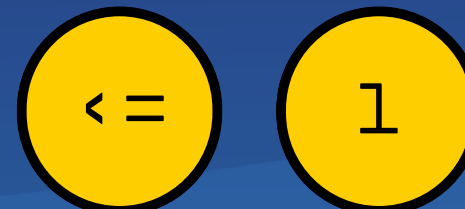
[DUAS OPÇÕES]



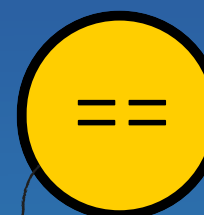
MAIOR OU IGUAL QUE



MENOR OU IGUAL QUE



IGUAL



ATENÇÃO QUE ISTO NÃO
É O MESMO QUE $A = 0$



Estes operadores
permitem a tomada de

DECISÕES

Decisões

ACONTECEM QUANDO USAMOS



IF

ELSE IF

ELSE



PODEMOS USAR PARA SABER COMO



decidir

```
#INCLUDE <STDIO.H>
```

```
INT MAIN() {  
  INT NOTA = 14;  
  
  IF [NOTA >= 18]  
  {  
    PRINTF("EXCELENTE\n");  
  }  
  ELSE IF [NOTA >= 10]  
  {  
    PRINTF("APROVADO\n");  
  }  
  ELSE  
  {  
    PRINTF("REPROVADO\n");  
  }  
  RETURN 0;  
}
```



Mas e se forem tarefas

CÍCLICAS?

Ciclos



SERVEM PARA REPETIR UMA
TAREFA VÁRIAS VEZES
AUTOMATICAMENTE

WHILE

FOR



While

APLICA-SE ENQUANTO A CONDIÇÃO SE CONFIRMAR

VAMOS PLANEAR:

```
ENQUANTO ALUNOS_ESTAO_ATENTOS == VERDADEIRO  
ESCREVER "ELAH!"  
SENÃO  
ESCREVER "SILENCIO"
```

ENQUANTO SE
CONFIRMAR

```
{ WHILE [ALUNOS_ATENTOS == TRUE]  
  {  
    PRINTF("ELAH!\n");  
  }  
  PRINTF("SILENCIO\n");  
  
  RETURN 0;
```

For

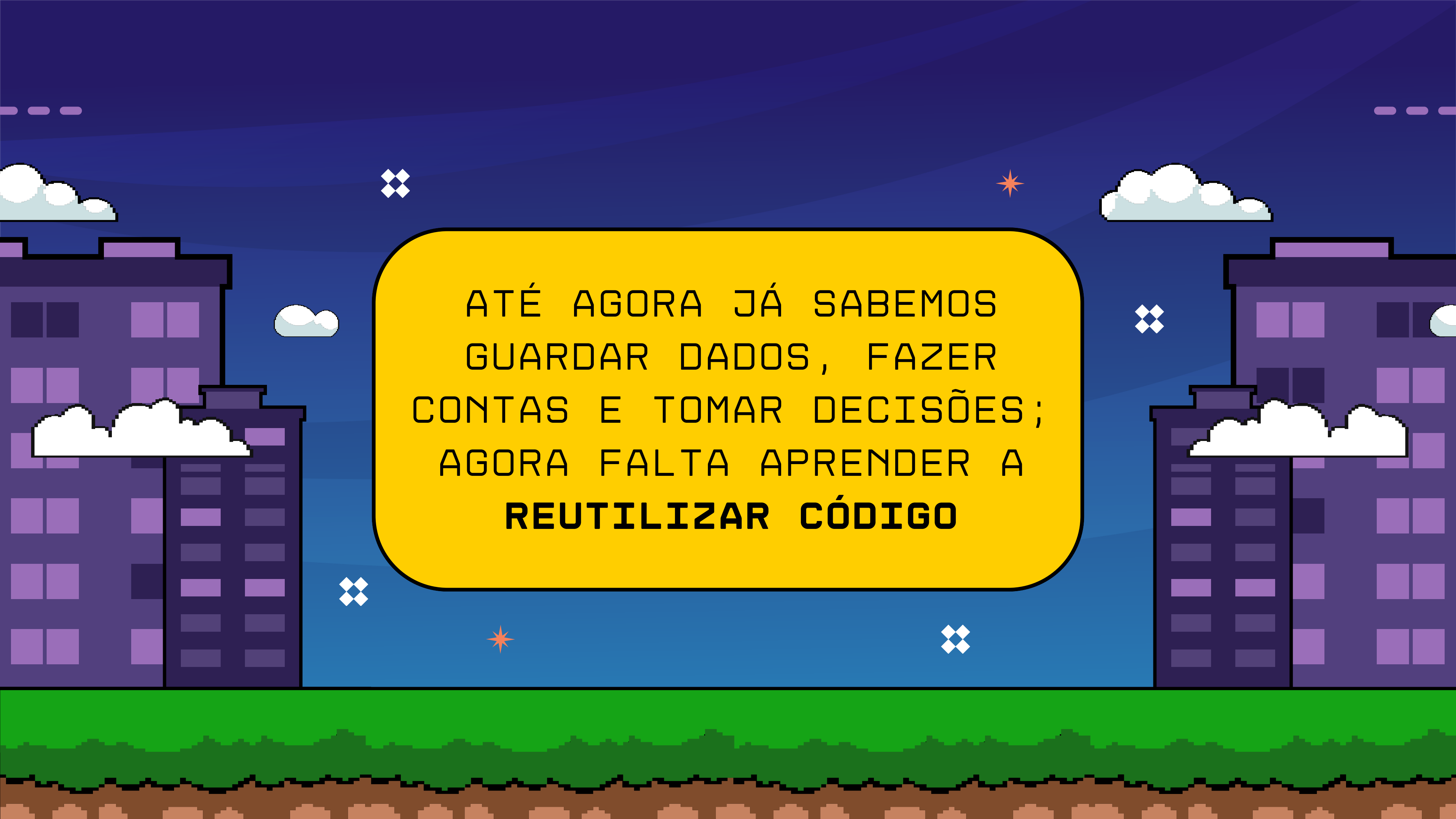
FAZ ISTO X VEZES

VAMOS PLANEAR:

VAMOS COLOCAR A PRESENÇA DOS ALUNOS(DO ALUNO 1 AO 5).

```
INT MAIN() {  
    INT I;  
    INT ALUNOS = 5;
```

```
    FOR [I = 1; I <= ALUNOS; I++] {  
        PRINTF("ALUNO %D PRESENTE\n", I);  
    }
```

A pixel art illustration of a city skyline at night. The sky is dark blue with stylized clouds, stars, and diamond-shaped light patterns. Two tall, purple buildings with many windows are on either side of a central yellow text box. The foreground consists of a green grassy area and a brown ground line.

ATÉ AGORA JÁ SABEMOS
GUARDAR DADOS, FAZER
CONTAS E TOMAR DECISÕES;
AGORA FALTA APRENDER A
REUTILIZAR CÓDIGO



O QUE SÃO FUNÇÕES?

FUNÇÕES SÃO BLOCOS DE CÓDIGO COM NOME QUE SERVEM PARA REUTILIZAR TAREFAS QUE FAZEMOS MUITAS VEZES.

EM VEZ DE REPETIR O MESMO CÓDIGO VÁRIAS VEZES, TU:

- ESCRIVES UMA VEZ
- DÁS UM NOME
- E DEPOIS SÓ “CHAMAS”



Exercício

NÓS TEMOS ESTADO
A USAR UMA
FUNÇÃO, CONSEGUES
IDENTIFICA-LA?

PARA QUE ISTO SEJA
POSSIVEL NÓS ENVIAMOS
INPUTS E RECEBEMOS **OUTPUTS**

INPUTS

OUTPUTS

Inputs

O COMPUTADOR RECEBE INFORMAÇÃO DO UTILIZADOR

TU ESCREVES ALGO NO TECLADO
O COMPUTADOR RECEBE A INFORMAÇÃO

Outputs

O COMPUTADOR MOSTRA INFORMAÇÃO AO UTILIZADOR

MOSTRA-TE A MENSAGEM NO ECRÃ



Resumo final

- **PROGRAMAR** É DAR INSTRUÇÕES CLARAS AO COMPUTADOR.
- UM **ALGORITMO** É UMA SEQUÊNCIA DE PASSOS PARA RESOLVER UM PROBLEMA.
- AS **VARIÁVEIS** GUARDAM INFORMAÇÃO.
- OS **OPERADORES** PERMITEM FAZER OPERAÇÕES E COMPARAÇÕES.
- AS **CONDIÇÕES** AJUDAM O COMPUTADOR A DECIDIR E OS CICLOS A REPETIR TAREFAS.
- AS **FUNÇÕES** PERMITEM REUTILIZAR CÓDIGO



❖ `printf`
❖ `("Obrigada!")` ❖

ACOMPANHA-ME NO LINKEDIN!

