



# Programação

## Introdução

Objetivo

# Programar computadores

Programar computadores

**estructuras  
de datos**

Programar computadores

estruturas  
de dados

**iteração e  
recursão**

Programar computadores

estruturas  
de dados

iteração e  
recursão

**abstração**

Programar computadores

estruturas  
de dados

iteração e  
recursão

abstração

**organização e  
modularização**

O que faz um computador?

**faz  
cálculos**

O que faz um computador?

faz  
cálculos



nativos a uma  
linguagem



definidos pelo  
programador



O que faz um computador?

faz  
cálculos



nativos a uma  
linguagem

**lê e guarda  
dados**



definidos pelo  
programador

O que faz um computador?

**computadores só sabem o que  
lhes dizemos**

faz  
cálculos



nativos a uma  
linguagem

lê e guarda  
dados



definidos pelo  
programador

Conhecimento declarativo

**factos**

Conhecimento declarativo

factos

a raiz de um número  $x$  é  $y$  tal  
que  $y * y = x$

Conhecimento imperativo

**receita**

a raiz de um número  $x$  é  $y$  tal que  $y * y = x$

**VS**

calcular raiz de  $x$

1) começar com um número ao acaso  $g$  menor que  $x$

2) se  $g * g$  estiver suficientemente perto de  $x$  (erro < precisão mínima), paramos e dizemos que  $g$  é a raiz

3) caso contrário, calcula-se novo  $g2 = (g + x/g) / 2$

4) repetir até estarmos perto o suficiente

O que é uma receita?

**sequência de  
passos**

O que é uma receita?

sequência de  
passos

**controle do  
fluxo**



O que é uma receita?

sequência de  
passos

controle do  
fluxo

**saber quando  
parar**

O que é uma receita?

sequência de  
passos

controle do  
fluxo

saber quando  
parar

**= algoritmo**

algoritmo

forma de resolver  
um problema

VS

**programa**

**expressão  
concreta de um  
algoritmo**

Computador simplificado

Memória

# Computador simplificado

Memória



The diagram illustrates a simplified computer architecture. At the top is a dark red rounded rectangle labeled 'Memória'. Below it is a larger, dark blue rounded rectangle containing a dark red rounded rectangle labeled 'Unidade de controlo'. Two red arrows connect the 'Unidade de controlo' to the 'Memória': one pointing down from the control unit to memory, and one pointing up from memory to the control unit.

Unidade de  
controlo

# Computador simplificado

Memória

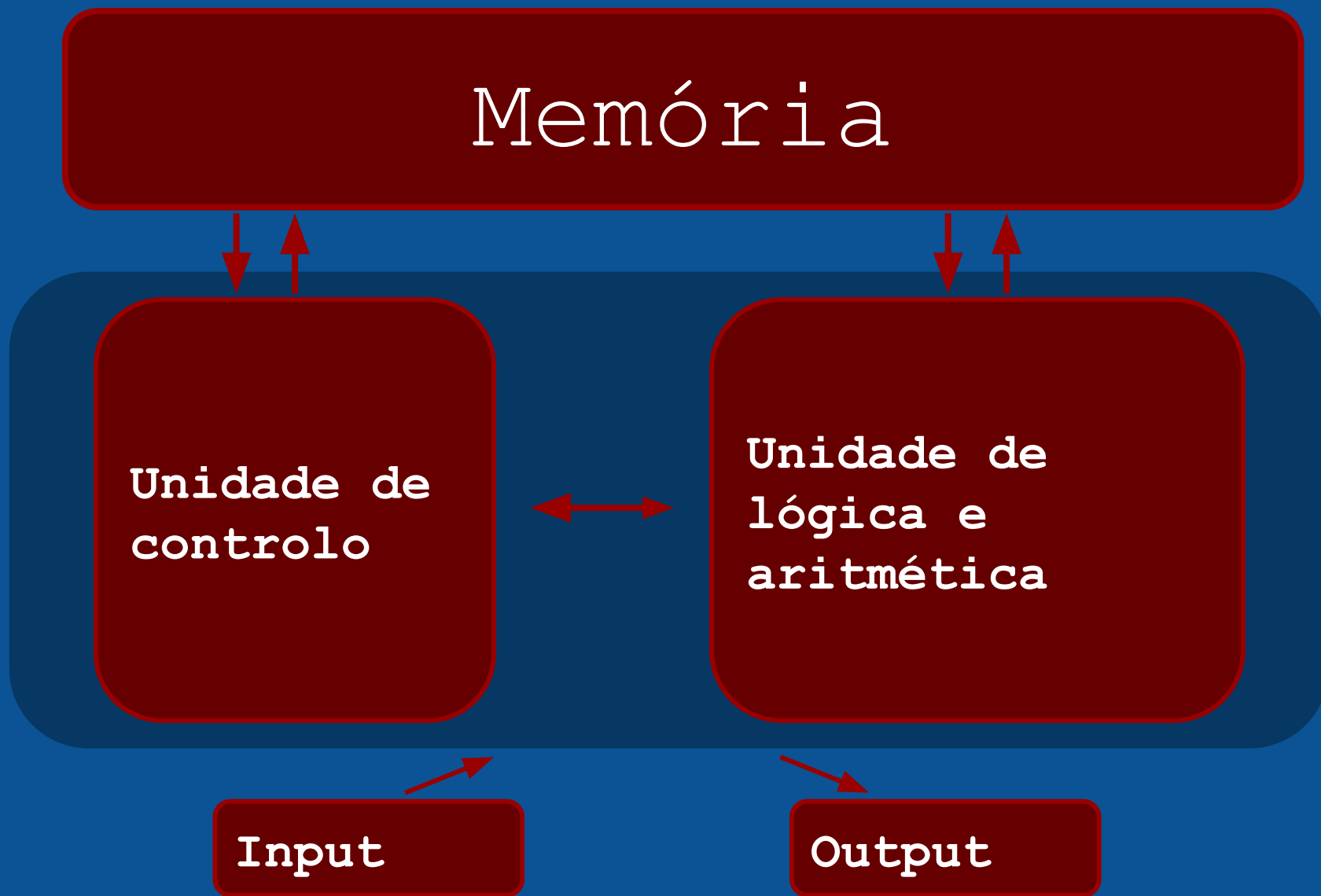
```
graph TD; Mem[Memória]; subgraph CPU; UC[Unidade de controlo]; ALU[Unidade de lógica e aritmética]; UC <--> ALU; end; Mem <--> UC; Mem <--> ALU;
```

The diagram illustrates a simplified computer architecture. At the top is a red rounded rectangle labeled 'Memória'. Below it is a larger dark blue rounded rectangle containing two red rounded rectangles: 'Unidade de controlo' on the left and 'Unidade de lógica e aritmética' on the right. A horizontal double-headed red arrow connects the two units within the CPU. Vertical double-headed red arrows connect the 'Memória' block to both the 'Unidade de controlo' and the 'Unidade de lógica e aritmética' blocks, indicating bidirectional communication.

Unidade de  
controlo

Unidade de  
lógica e  
aritmética

# Computador simplificado



# Computador simplificado

passos a seguir      dados

```
double square_root(double x, double guess, double error){
    double new_guess;
    if (absolute_value(guess * guess - x) < error) return
    guess;

    new_guess = (guess + x / guess) / 2;

    return square_root(x, new_guess, error);
}

int main(void){
    double x = 16;
    double raiz = square_root(x, 1, 0.1)
    printf("Square root of %lf is %lf\n", x, raiz);
}
```

executa  
passos na  
ordem  
certa

realiza  
operações  
primitivas

+ - / \* < >  
move dados

16

4.002258



Código a programa

raiz de 16?

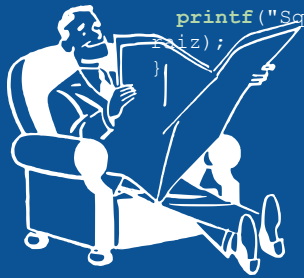


```
double square_root(double x, double guess,
double error){
    double new_guess;
    if (absolute_value(guess * guess - x) <
error) return guess;

    new_guess = (guess + x / guess) / 2;

    return square_root(x, new_guess, error);
}
```

```
int main(void){
    double x = 16;
    double raiz = square_root(x, 1, 0.1)
    printf("Square root of %lf is %lf\n", x,
raiz);
}
```



compilador



```
7f454c46211000000000203e01000
90440000004000000000a811000000
0000400380904001e01b060005000
400000000400400000004004000000
f81000000f8100000080000000300
040003820000000382400000038240
000001c000000001c0000000100000
0005000000000000000400000000
0004a00000004a000000002000
00600010e00000010e600000
0000003820000000402000000
0002000600028e00000028e
```

raiz de 16 =  
4.002258



Quais são elementos  
essências da  
programação?

Quais são operações  
básicas que um  
computador consegue  
fazer?

O que é um algoritmo?