

Fundamentos da Programação

Estruturas de controlo condicional

Conteúdo

- Operadores relacionais
- Operadores lógicos
- Estruturas de controlo condicional
 - Decisão (**if**)
 - Seleção (**switch**)

Estruturas de controlo condicional

- Na implementação de um programa é muitas vezes necessário **avaliar** uma determinada **expressão condicional** para **decidir** que **ações** devem ser executadas.
- Estruturas de controlo condicional:
 - Decisão.
 - Seleção.

Expressão condicional

- Permite a execução de um determinado conjunto de ações dependendo do valor devolvido por uma condição lógica.
- Devolve apenas um de dois possíveis valores: **verdadeiro** (true) ou **falso** (false).
 - Em C, o valor **0** (zero) corresponde a **falso**. Qualquer valor diferente de **0** é considerado **verdadeiro**.
 - Noutras linguagens de programação, existe um tipo de dados **booleano** que apenas permite armazenar os valores **true** e **false**.

Operadores relacionais e lógicos



Operadores relacionais

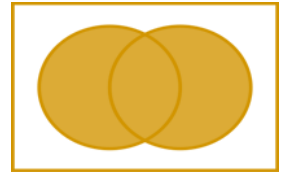
Sequência	Descrição	Operação	Resultado ¹
<code>==</code>	Igual a ...	<code>a == b</code>	Falso
<code><</code>	Menor que ...	<code>a < b</code>	Verdadeiro
<code><=</code>	Menor ou igual que ...	<code>a <= b</code>	Verdadeiro
<code>></code>	Maior que ...	<code>a > b</code>	Falso
<code>>=</code>	Maior ou igual que ...	<code>a >= b</code>	Falso
<code>!=</code>	Diferente de ...	<code>a != b</code>	Verdadeiro

¹Assumindo `a = 0` e `b = 1`

Operadores lógicos

Operador	Significado	Exemplo	Resultado
&&	Conjunção (e)	(cond1 && cond2)	Verdadeiro, se ambas as condições forem verdadeiras. Falso, se alguma delas for falsa.
 	Disjunção (ou)	(cond1 cond2)	Verdadeiro, se alguma das condições for verdadeira. Falso, se ambas forem falsas.
!	Negação	!(cond1)	Verdadeiro, se a condição for falsa. Falso, se a condição for verdadeira.

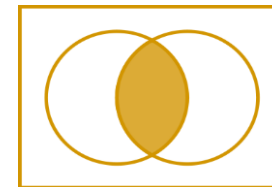
Conjunção lógica (&&)



- Verdadeiro, se ambas as condições forem verdadeiras. Falso, se alguma delas for falsa.
 - Exemplo: `(idade >= 18 && genero == 'M')`

p	q	p && q
V	V	V
V	F	F
F	V	F
F	F	F

Disjunção lógica (||)



- Verdadeiro, se alguma das condições for verdadeira. Falso, se ambas forem falsas.
 - Exemplo: **(nota1 >= 9.5 || nota2 >= 9.5)**

p	q	p q
V	V	V
V	F	V
F	V	V
F	F	F

Negação lógica (!)



- Verdadeiro, se a condição for falsa. Falso, se a condição for verdadeira.
 - Exemplo: **!(nota < 9.5)**

p	!p
V	F
F	V

Decisão



if... else...

- O **if... else...** (se... então... senão...) permitem, com base no resultado da condição, decidir sobre a execução ou não de determinadas instruções ou optar entre duas alternativas.
 - Exemplos:
 - se chove **então** levo guarda-chuva.
 - se chove **então** levo guarda-chuva **senão** levo boné.

if...

```
if (condição) {  
    [instrução 1;  
    instrução 2;  
    ...;  
    instrução n;]  
}
```

Executa o bloco se a condição for verdadeira

```
...  
if (nota >= 9.5) {  
    printf("O aluno teve positiva");  
    ++aprovados;  
}  
...
```

if... else...

```
if (condição) {  
    [instrução 1; ...; instrução n;]  
} else {  
    [instrução 1; ...; instrução n;]  
}
```

Executa o bloco se a condição for **verdadeira**

Executa o bloco se a condição for **falsa**

```
...  
if (nota >= 9.5) {  
    printf("O aluno teve positiva");  
    ++aprovados;  
} else {  
    printf("O aluno teve negativa");  
}  
...
```

if... else...

```
if (condição)
    instrução 1;
else {
    instrução 1;
    instrução 2;
    [...; instrução n;]
}
```

A indentação ajuda a legibilidade, mas não tem significado. Neste caso ajuda a perceber que instruções pertencem ao bloco.

Embora o uso de chavetas seja opcional quando existe apenas uma instrução, é considerado **má prática** não as incluir.
É **obrigatório** quando existe mais do que uma instrução.

```
...
if (nota >= 9.5) {
    printf("O aluno teve positiva");
    ++aprovados;
} else // má prática!!!!
    printf("O aluno teve negativa");
printf("Aprovados: %d", aprovados);
...
```

Estruturas aninhadas

- Por vezes não basta testar uma condição lógica para tomar uma decisão, por isso uma ação pode ser outro **if** (aninhado ou encadeado, *nested*).

```
...
if (teste1 >= 10) {
    printf("passou no 1º exame");
    if (teste2 >= 10) {
        printf("passou no 2º exame.");
    }
} else {
    printf("Tem de ir a exame.");
}
...
```


... else if...

```
if (condição) {  
    [instrução 1; ...; instrução n;]  
} else {  
    if (condição) {  
        [instrução 1; ...; instrução n;]  
    } else {  
        [instrução 1; ...; instrução n;]  
    }  
}
```

```
if (condição) {  
    [instrução 1; ...; instrução n;]  
} else if (condição) {  
    [instrução 1; ...; instrução n;]  
} else {  
    [instrução 1; ...; instrução n;]  
}
```

```
...  
if (nota > 15) {  
    printf("Dispensado do 2º teste!");  
} else if (nota > 10) {  
    printf("Tem de fazer o 2º teste!");  
} else {  
    printf("Reprovado!");  
}  
...
```

Operador ternário (?)

- Sintaxe: **condição ? instrução_se_verdadeiro : instrução_se_falso;**
- Exemplo:

Os seguintes blocos:

```
if (nota >= 9.5) {  
    resultado = 'P';  
} else {  
    resultado = 'R';  
}
```

```
if (num % 2 == 0) {  
    printf("Par");  
} else {  
    printf("Impar");  
}
```

Podem ser escritos da seguinte forma:

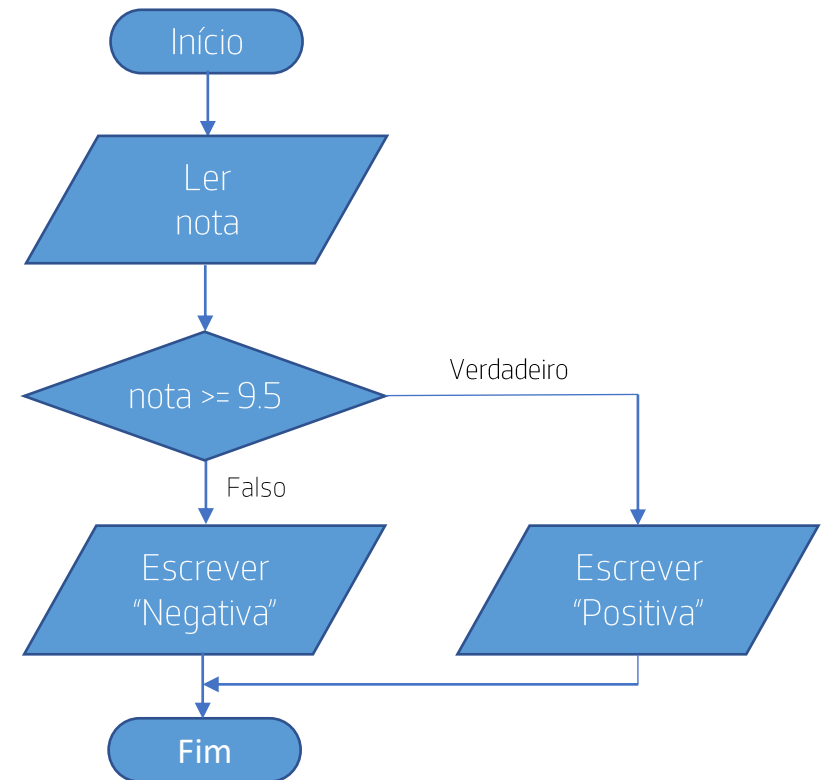
```
resultado = (nota >= 9.5) ? 'P' : 'R'
```

```
(num % 2 == 0) ? printf("Par") : printf("Impar");  
  
printf("%s", (num % 2 == 0) ? "Par" : "impar");
```

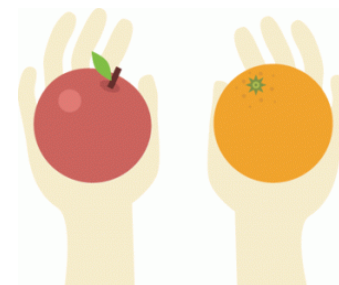
Decisão



```
início  
  ler nota  
  se nota >= 9.5 então  
    escrever "Positiva"  
  senão  
    escrever "Negativa"  
  fimse  
fim
```



Seleção



switch

- O **switch** (escolhe... caso...) permite mediante o valor de uma expressão (cujo resultado seja avaliado num **char**, **int** ou **long**) optar entre um determinado número de alternativas (*cases*) possíveis.

switch

```
switch (expressão) {  
    case expressão1:  
        [instrução 1; ...; instrução n;]  
        [break;]  
    case expressão2:  
        [instrução 1; ...; instrução n;]  
        [break;]  
    ...  
    case expressãoN:  
        [instrução 1; ...; instrução n;]  
        [break;]  
    default:  
        [instrução 1; ...; instrução n;]  
}
```

switch avalia uma variável e depois compara-a com as constantes de tipo inteiro ou carácter de cada **case**.

break (opcional) faz com que o **switch** termine. Se esta instrução não estiver presente no final do **case** selecionado, serão avaliados os **case** seguintes.

default (opcional) permite contemplar os **case** que não verifiquem nenhum dos valores anteriores.

switch

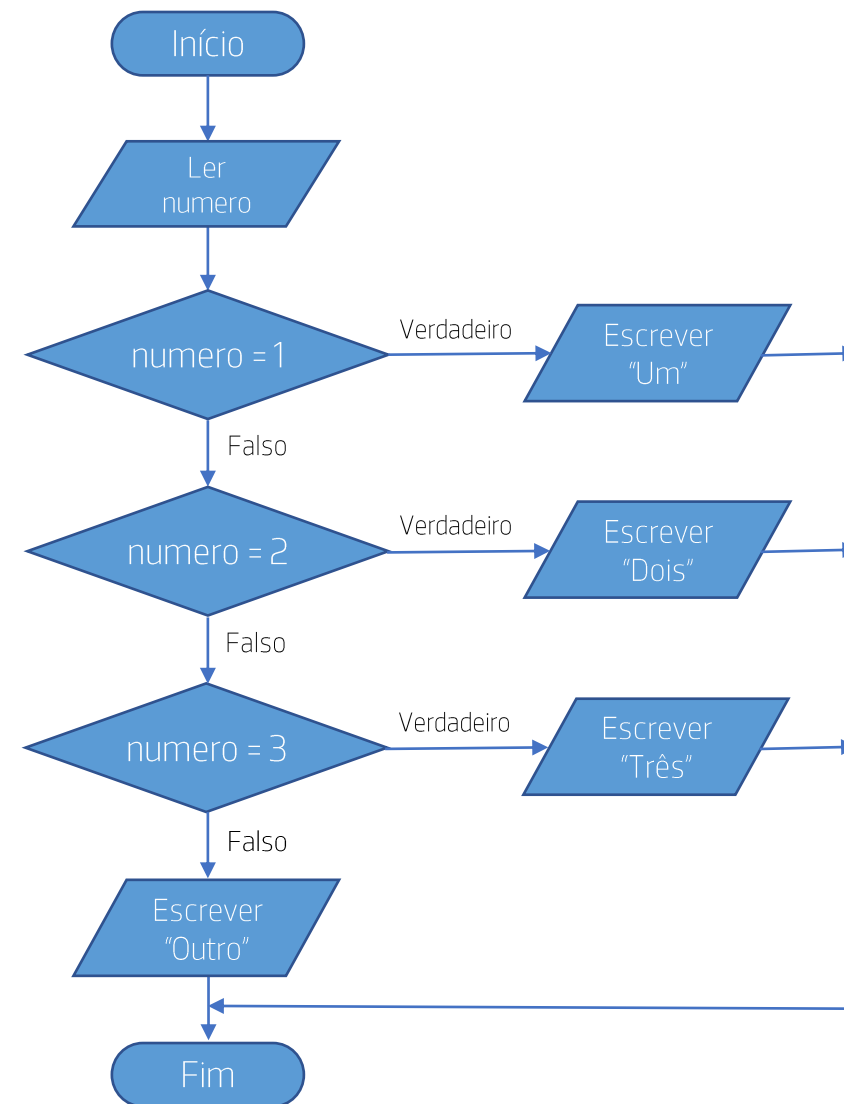
```
...
scanf("%d", &numero);
switch (numero) {
    case 1:
        printf("Um");
        break;
    case 2:
        printf("Dois");
        break;
    case 3:
        printf("Três");
        break;
    default:
        printf("Escolheu outro numero!");
}
...
```

```
...
switch (estado_civil) {
    case 's':
    case 'S':
        printf("Solteiro");
        break;
    case 'c':
    case 'C':
        printf("Casado");
        break;
    case 'v':
    case 'V':
        printf("Viúvo");
        break;
    case 'd':
    case 'D':
        printf("Divorciado");
        break;
}
...
```

Seleção



```
início
  ler numero
  escolhe numero
    caso 1: escrever "Um"
    caso 2: escrever "Dois"
    caso 3: escrever "Três"
    defeito: escrever "Outro"
  fim escolhe
fim
```



Leitura recomendada

- (Capítulo 3) Damas, L. Linguagem C; FCA – Editora de Informática, Lda, 1999; ISBN 9789727221561.

