

# Módulo 17

# Tratamento

# de Erros

Tanjil Shahid Khan Nº21

Escola Filipa de Vilhena

Ano Letivo 2020/2021

Curso Técnico de Gestão e Programação de Sistemas  
Informáticos

# O que é um erro

Os erros são comuns na parte da programação, que pode ocorrer falhas e comportamentos anormais durante a execução do código.

# Tipos de Erros

Existem 3 tipos de erros. Os erros de sintática, os erros de lógica e os erros de tempo de execução.



## **Erro de sintática**

Os erros de sintática são erros onde teve algumas falhas de acordo com as regras da gramática da sua linguagem de programação.



# **Erro de lógica**

Erros lógicos acontece onde o resultado final não é a mesma que o programador esteve a espera que fosse.

Os erros lógicos podem ser evitados com testes.

# **Erro de tempo de execução**

Erros de tempo de execução, também conhecido como erro de runtime, é o erro é o mais complicado de se resolver porque não se sabe a origem da causa do erro runtime.

# Erros comuns em C



- Esquecer o ponto e virgula ;



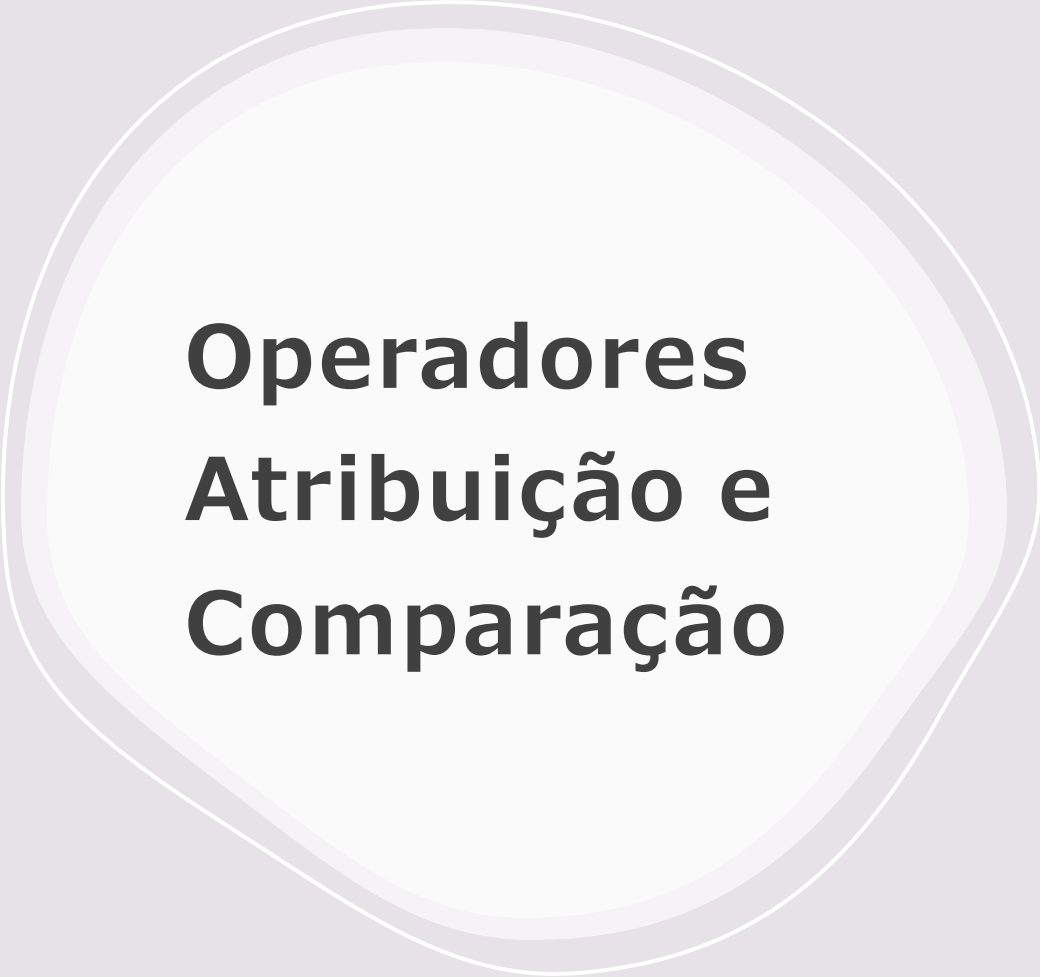
- Esquecer de fechar as aspas  
""



- Escrever mal alguma palavra



- Esquecer de fechar as  
chavetas {}



# **Operadores Atribuição e Comparação**

Os programadores podem enganar-se quando querem comparar variáveis, em vez de usar o operador de comparação, enganam-se em usar o operador de atribuição.



# ***Case sensitive***

A linguagem C possui case sensitive, as palavras chaves como por exemplo `int`, `if`, `for` e etc..., devem ser sempre em minúsculas.



# Variáveis

O erro comum de lógica que se pode cometer é usar as variáveis antes de ser declaradas. Isso vai gerar um erro em que a variável não existe. É mais precisamente um erro de lógica.

# O break

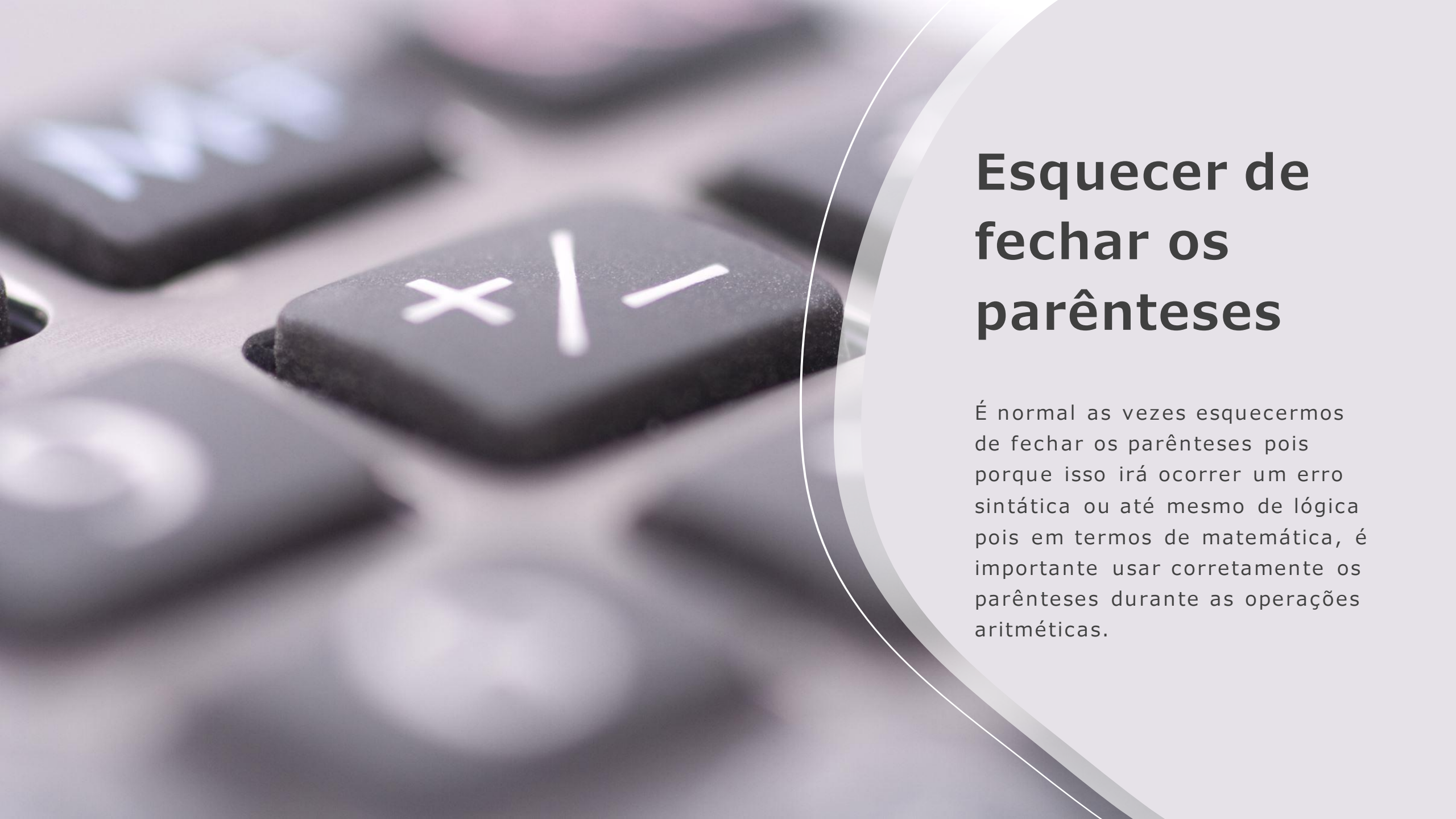
É comum também esquecer do comando break durante o bloco switch case. O programa vai ficar sempre a continuar a executar para os próximos cases até terminar o bloco.

# Ciclos infinitos

É comum acontecer ciclos infinitos pois porque a razão é onde nós esquecemos de atualizar para incrementar o contador que controla a condição de paragem de um ciclo infinito.

## Usar ; erradamente

As vezes, é normal usar o ; nos locais indesejados como por exemplo em condições `if()`; não devemos meter o ponto e virgula, em vez de usar o ponto e virgula, devemos usar as chavetas. Também no `#include <iostream>`; é um local indesejado de usar o ponto e virgula

A close-up photograph of a computer keyboard. The central focus is on a single key that features three symbols: a multiplication sign (x), a division sign (/), and a subtraction sign (-). The key is dark with white symbols. Other keys, including the arrow keys, are visible in the background but are out of focus. A large, semi-transparent white circle with a thin grey border is positioned on the right side of the image, containing the title and a paragraph of text.

# Esquecer de fechar os parênteses

É normal as vezes esquecermos de fechar os parênteses pois porque isso irá ocorrer um erro sintática ou até mesmo de lógica pois em termos de matemática, é importante usar corretamente os parênteses durante as operações aritméticas.

# Não agrupar a condição IF

---

Pode ocorrer um erro de lógica devido a não agrupar bem as condições if. Por exemplo:

```
if(a < b)
```

```
    cout << "a maior que b";
```

```
    cout << "o b e menor que a";
```

# Não agrupar a condição IF

---

No código anterior, isso é a mesma coisa que

```
if(a < b)
```

```
    cout << "a maior que b";
```

```
cout << "o b e menor que a";
```



# Não agrupar a condição IF

---

O que era melhor era provavelmente usar as chavetas

```
if(…){
```

```
    Instrucao1;
```

```
    Instrucao2;
```

```
}
```