

Disciplina: Introdução a Programação

Aula 3: Expressões e tabela da verdade

Apresentação

Na a aula anterior, falamos brevemente sobre os operadores. Nesta aula, veremos de que forma utilizamos esses operadores na criação de expressões que utilizam não somente números, mas também variáveis. Como já estudamos, as variáveis são essenciais na programação. As expressões também são muito utilizadas para cálculos e para tomada de decisões.

Para finalizar, abordaremos as tabelas da verdade, ou tabelas-verdade, que são uma ferramenta muito utilizada para verificar todos os resultados possíveis para uma expressão lógica. Compreender as expressões lógicas e os possíveis resultados que elas podem retornar é muito importante para que, no futuro, você consiga construir os testes lógicos a serem avaliados por estruturas mais complexas utilizadas na programação de computadores.

Bons estudos!

Objetivos

- Identificar os diferentes tipos de expressão;
- Analisar a forma como são escritas as expressões;
- Avaliar expressões lógicas para construção de tabelas da verdade.

Operadores

Na aula anterior, conhecemos os operadores que nos permitirão realizar atribuições e construir expressões aritméticas, relacionais e lógicas. Vimos que esses operadores podem ser [unários](#)¹ ou [binários](#)² , e aprendemos, também, que eles podem ser do tipo aritmético, relacional, lógico, de atribuição, de incremento e de decremento.

Vamos rever esses operadores com mais detalhes?

Leia o texto “[Operadores <galeria/aula3/docs/operadores.pdf>](#)” para lembrá-los.

Vamos, agora, ver de que maneira eles são utilizados nos programas e algoritmos que iremos construir.

Expressões matemáticas

Para que exista uma expressão matemática, ou operação algébrica, é necessário que existam, no mínimo, dois valores ou variáveis e um operador.

Observe os exemplos a seguir, que representam expressões matemáticas:

12 - 7	14 / 2
35 x 3	14 + 2

Para que sejam usadas em um programa ou algoritmo, essas expressões precisam sofrer algumas adaptações. Vejamos:

O sinal de multiplicação: Conforme vimos anteriormente, em programação não é possível usar o “.” ou o “x” para representar essa operação. No lugar deles, devemos utilizar o sinal de “*” (asterisco).	Quando o computador resolver uma expressão matemática, por mais simples que ela seja, ele chegará a um resultado. Esse resultado precisa ser armazenado em uma variável ou constante.
--	---

Para que as expressões matemáticas nos exemplos possam ser utilizadas em um algoritmo ou programa, precisam ser escritas conforme a seguir:

Subtra = 12 - 7	Divide = 14 / 2
Mult = 35 * 3	Soma = 14 + 2

Veja que, agora, os resultados das expressões já têm um local para serem armazenados: nas variáveis “Subtra”, “Divide”, “Mult” e “Soma”, respectivamente.

Um operador que está disponível em diversas linguagens de programação é o módulo, representado pelo sinal “%”.

A operação de módulo é responsável por retornar o resto de uma divisão. Observe os exemplos:

Resto1 =120%2

Resto2 = 310%3

A operação de **módulo** é responsável por retornar o resto de uma divisão. Desse modo, a variável “Resto1” armazenará o valor 0, e a variável “Resto2” armazenará o valor “10”.

Lembre-se de que os operadores têm o que chamamos de **precedência** — responsável por definir a ordem em que as operações serão executadas, quando uma expressão combina várias delas.

Observe a tabela de precedência a seguir:

Precedência dos operadores matemáticos	
1º.	Multiplicação, divisão, módulo
2º.	Adição, subtração

Dica

Assim como são usados na Matemática, os parênteses também podem ser utilizados em algoritmos e programas. Quando eles estiverem presentes, resolva primeiro o conteúdo dentro dos parênteses.

Expressões relacionais

As expressões relacionais são construídas de modo a avaliar a relação entre os termos da expressão.

Uma expressão desse tipo é composta por termos de um mesmo tipo e por operadores relacionais. O resultado de uma expressão relacional será sempre um valor lógico:



Vejamos um exemplo:

Exemplo

Considere as variáveis N1=15 e N2=29:

- Maior = (N1 > N2) → Maior armazena falso
- Menor = (N1 < N2) → Maior armazena verdadeiro.
- Diferente = (N1 != N2) → Diferente armazena verdadeiro.

As expressões relacionais são muito utilizadas na programação, pois elas nos ajudam a avaliar condições.



Se você precisa escrever um algoritmo que, a partir da idade do usuário, informe se ele pode ou não votar, é necessário avaliar se a idade é maior ou igual a 16 (idade>=16).



Se o algoritmo precisa informar se um aluno foi aprovado ou reprovado em uma instituição de ensino em que a média mínima é 7, você precisa avaliar se a média é maior ou igual a 7 (Media>=7).

Percebeu como são importantes as expressões relacionais?

Expressões lógicas

As expressões lógicas, também conhecidas como expressões booleanas, são construídas com o auxílio dos operadores lógicos e são responsáveis por avaliar uma ou mais condições. O resultado de uma expressão lógica será sempre um valor lógico.



Para avaliar uma expressão lógica é preciso conhecer o que retornam os operadores lógicos quando avaliam uma relação.

Vejamos um exemplo:

Exemplo

Imagine que você precisa escrever um algoritmo que avalie se um número armazenado na variável “Num” é positivo e par.

+

Para ser positivo, o número precisa ser maior do que zero.

2

Para ser positivo, o número precisa ser maior do que zero.

Uma solução para esse problema é mostrada a seguir:

se (Num>0) && (Num%2==0)
escreva (“O número é par e positivo!”)

→

Expressão lógica:
(Num>0) && (Num%2==0)

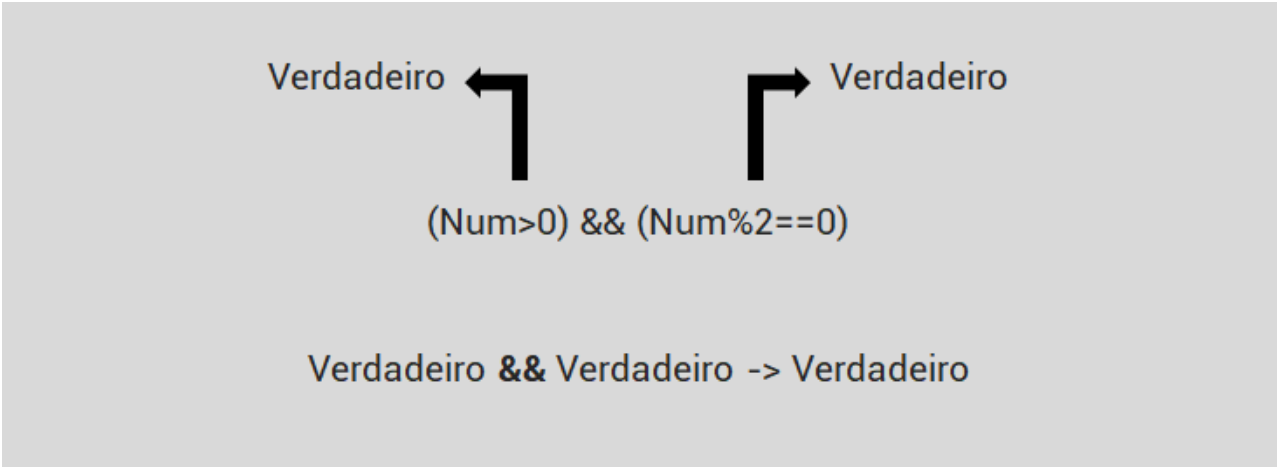
Veja que, na situação proposta, precisamos avaliar duas condições relacionais: (Num>0) e (Num%2==0). Para fazer com que as duas sejam avaliadas, precisamos construir uma expressão lógica com o auxílio do operador “e” (&&).

Agora vamos avaliar duas situações diferentes.

1

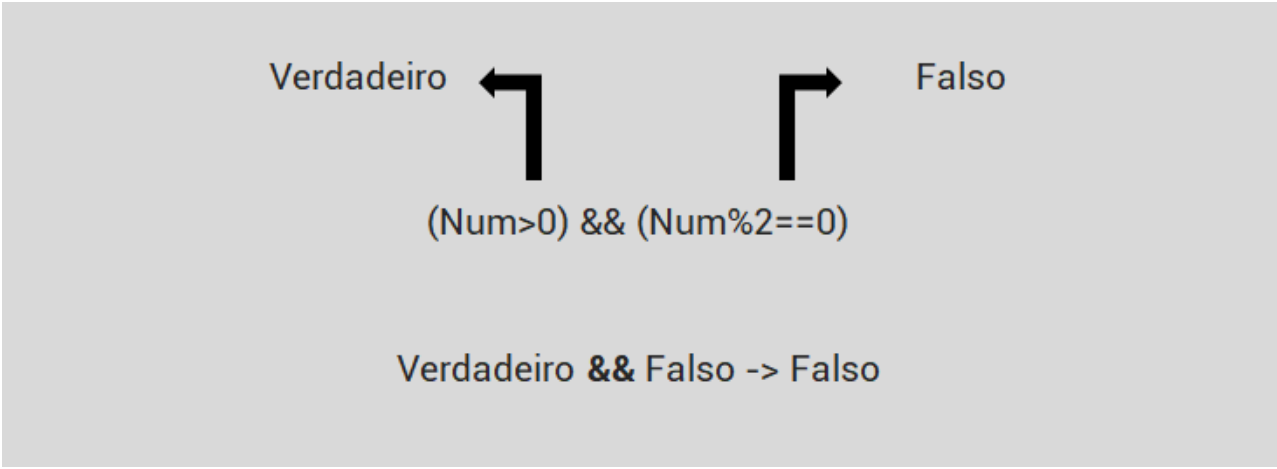
Considere que a variável “Num” está armazenando o valor 10. Ao avaliarmos a expressão lógica para esse caso, veremos que ambas expressões relacionais retornam verdadeiro.

Num = 10
(Num>0) ➡ Esta expressão relacional retorna verdadeiro.
(Num%2==0) ➡ Esta expressão relacional retorna falso.



2 Agora imagine que a variável “Num” está armazenando o valor 15. Neste caso, uma expressão relacional é verdadeira, mas a outra é falsa. Observe:

Num = 15
(Num>0) ➡ Esta expressão relacional retorna verdadeiro.
(Num%2==0) ➡ Esta expressão relacional retorna falso.



Observe que:

_____ 1 _____

_____ 2 _____

Quando as duas expressões avaliadas pelo operador lógico “&&” são verdadeiras, o resultado final da expressão lógica é verdadeiro.

Quando um valor é verdadeiro e o outro é falso, a expressão lógica retorna falso.

Isso acontece porque há regras para que os operadores lógicos definam o valor a retornar.

Agora observe as tabelas a seguir.

Operador && (E)		
A	B	A && B
Verdadeiro	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Falso
Falso	Verdadeiro	Falso
Falso	Falso	Falso

Operador (OU)		
A	B	A B
Verdadeiro	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Verdadeiro
Falso	Verdadeiro	Verdadeiro
Falso	Falso	Falso

Operador ! (NÃO)	
A	!A
Verdadeiro	Falso
Falso	Verdadeiro

Essas tabelas apresentam todas os resultados possíveis para quando os operadores avaliam dois valores lógicos.

1

O operador “E” precisa que os dois valores avaliados sejam verdadeiros para que o resultado final também seja verdadeiro.

2

O operador “OU” precisa que os dois operadores avaliados sejam falsos para que o resultado final também seja falso.

Assim como no caso dos operadores matemáticos, os operadores lógicos também possuem precedência, conforme a tabela a seguir:

Precedência dos operadores lógicos	
1º.	Não (!)
2º.	E (&&)
3º.	Ou ()

Atenção

Conhecer a precedência dos operadores é essencial quando é necessário resolver uma expressão na qual vários operadores são combinados.

Vejamos uma situação:

Considere as variáveis A = verdadeiro, B = verdadeiro e C = falso e a expressão lógica A && B || !C. Observe as etapas a serem seguidas na resolução:

1º. Passo

Substituímos as variáveis por seus valores equivalentes.



Verdadeiro && Verdadeiro || !Falso

2º. Passo

Respeitando a precedência, resolvemos !C.



Verdadeiro && Verdadeiro || Verdadeiro

3º. Passo

Em seguida, resolvemos A && B.



Verdadeiro || Verdadeiro

4º. Passo

Por fim, resolvemos a relação “ou” entre o resultado de A && B com !C.



Verdadeiro

Nesse exemplo, resolvemos a expressão lógica para o caso das variáveis A = verdadeiro, B = verdadeiro e C = falso. Há situações, entretanto, em que desejamos descobrir todos os resultados possíveis para os diferentes valores das variáveis. Nesse caso, é necessário montar uma tabela da verdade.

Atividade

1 - Resolva as expressões a seguir para A = 8, B = 9 e C = 12 e informe o que estará armazenado na variável “Resultado”:

- a) Resultado = (A%2) + C – B
- b) Resultado = A * B + A + C – B
- c) Resultado = (C – A) / 2 + B
- d) Resultado = A >= B
- e) Resultado = C != A
- f) Resultado = B <= C

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

Tabela da verdade

Uma **tabela da verdade**, ou **tabela-verdade**, é uma tabela matemática capaz de apresentar todos os resultados possíveis para uma expressão lógica, a partir das diferentes combinações de valores para as variáveis avaliadas. Observe a tabela da verdade para a expressão A && B com !C, utilizada no exemplo anterior.

A	B	C	!C	A && B	A && B !C
V	V	V	F	V	V
V	V	F	V	V	V
V	F	V	F	F	F
V	F	F	V	F	V
F	V	V	F	F	F
F	V	F	V	F	V
F	F	V	F	F	F
F	F	F	V	F	V

As colunas da tabela são definidas segundo as variáveis da expressão lógica e os passos para resolução da mesma. Cada linha da tabela apresenta uma possibilidade de combinação para as variáveis e os passos para a solução da expressão até que se chegue a um resultado final. Vejamos:

Na **primeira** linha:

Quando A

=

Verdadeiro

Quando B

=

Verdadeiro

Quando C

=

Verdadeiro

O resultado da expressão será:

Verdadeiro

Na **segunda** linha:

Quando A

=

Verdadeiro

Quando B

=

Verdadeiro

Quando C

=

Falso

O resultado da expressão também é:

Verdadeiro

Na **terceira** linha, quando as variáveis alteram os valores para:

Quando A

Quando B

Quando C

=

=

=

Verdadeiro

Falso

Verdadeiro

O resultado final para a expressão é:

Falso

Para descobrir quantos resultados possíveis existem para uma expressão lógica, é preciso identificar a quantidade de variáveis que compõem essa expressão e descobrir quantas possibilidades diferentes de combinação de verdadeiro e falso elas geram.

A quantidade de variáveis será expoente de uma base 2 e o resultado dessa potenciação dirá quantos resultados diferentes existem para a expressão. Observe:

1º. Passo

Identificar a quantidade de variáveis.



A && B || !C → Três variáveis: A, B e C

2º. Passo

Utilizar a quantidade de variáveis como expoente de uma base 2.



2³ → Quantidade de variáveis da expressão

3º. Passo

Resolver a potenciação para descobrir a quantidade de combinações possíveis.



2³ = 8 → Quantidade de combinações possíveis

A quantidade de combinações possíveis nos ajudará a definir quantas linhas irão compor a **tabela da verdade**.

Exemplo

Leia o texto “[Exemplo – Quantidade de variáveis <galeria/aula3/docs/exemplo.pdf>](#)” para conhecer um exemplo que demonstra a quantidade de variáveis.

Atividade

2 - Resolva as expressões lógicas a seguir para A = V, B = F, C = F e D = V e informe o que estará armazenado na variável “Resultado”:

- a) Resultado = !A || B && C || D
- b) Resultado = A && B || !(C && D)
- c) Resultado = !A || B || C && D && A

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online

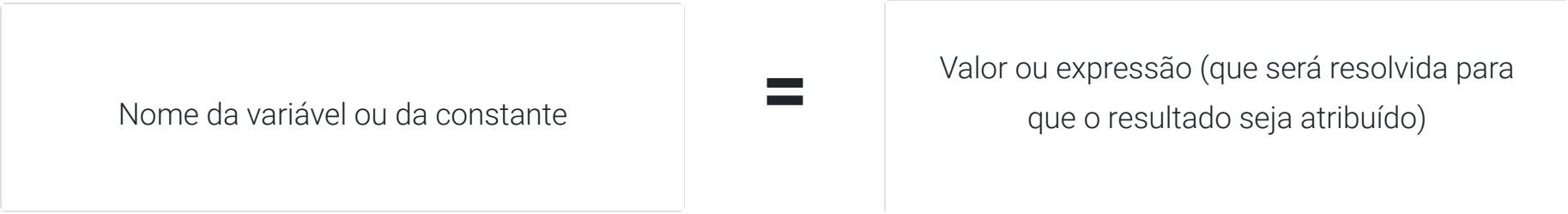
Operadores de atribuição

Os operadores de atribuição são responsáveis por armazenar valores em constantes e variáveis. Observe:

Resultado = 0 → Atribui 0 à variável “Resultado”;
Pi = 3.14 → Atribui 3.14 à constante “Pi”;
Nome = “Thomás” → Atribui “Thomás” à variável “Nome”;
Soma = N1 + N2 → Atribui a soma de A+B à variável “Soma”.

Nos exemplos anteriores, vimos exemplos utilizando o operador de atribuição simples, que é o sinal de igualdade (“=”).

O padrão para atribuição é o seguinte:



Existem ainda outros operadores de atribuição que são combinados aos operadores matemáticos para efetuarem um cálculo e armazenarem o resultado em uma variável:

1

Operador de adição e atribuição

Soma += 5 → Equivale à expressão Soma = Soma + 5

2

Operador de subtração e atribuição

Subtra -= 10 → Equivale à expressão Subtra = Subtra – 10

3

Operador de divisão e atribuição

Divide /= 2 → Equivale à expressão Divide = Divide / 2

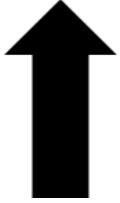
4

Operador de multiplicação e atribuição

Multiplica *= 3 → Equivale à expressão Multiplica = Multiplica * 3

Operadores de incremento e decrementar

Em programação, incrementar e decrementar significam, respectivamente, aumentar e diminuir o valor de uma variável.



O operador de **incremento aumenta** em uma unidade o valor da variável.



Enquanto o operador de **decremento diminui** em uma unidade o valor da variável.

Eles estão exemplificados a seguir:

_____ 1 _____

Operador de incremento

Total ++ ➡ Equivale à expressão Total = Total + 1

_____ 2 _____

Operador de decremento

Total -- ➡ Equivale à expressão Total = Total - 1

Atividade

3 - Utilizando os operadores especiais de atribuição, reescreva as expressões a seguir:

- a) Total = Total * 5
- b) Pontos = Pontos / 8
- c) Itens = Itens + 3
- d) Desconto = Desconto – 4

4 - Para cada uma das expressões lógicas a seguir, monte a tabela da verdade equivalente:

- a) !A || B && A
- b) A && B || !C
- c) !A || B || C && D && A

Notas

Unários ¹

Quando atuam sobre somente um operando.

Binários ²

Quando atuam sobre dois operandos.

Referências

Próxima aula

- Comandos de entrada e saída;
- Problemas simples que podem ser resolvidos com programas;

Explore mais

- Algoritmos e programas para solução dos problemas apresentados.
-

Você já construiu uma galáxia e praticou suas habilidades artísticas com o auxílio de pequenos programas de computador. Pode parecer brincadeira, mas esses exercícios estão sendo essenciais para que você pratique o raciocínio lógico e a estruturação de seu pensamento para solução de problemas. Vamos continuar os desafios? Desta vez, você vai ajudar os zumbis do jogo [Plants vs. Zombies <https://studio.code.org/s/course3/stage/8/puzzle/1>](https://studio.code.org/s/course3/stage/8/puzzle/1) a colher plantas.

Atenção! Aqui existe uma videoaula, acesso pelo conteúdo online