**Aula III - DevAppII**

**Lógica Dart**

**Operadores Aritméticos - Continuação**

**Operadores Lógicos e Relacionais**

**If - Else**

**Switch Case**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Fontes:**

<http://davesbalthazar.com.br/029-operadores-em-dart-tutorial-de-dart/>

<https://www.flutterbrasil.com/14-dart-iniciante-operacoes-logicas>

<https://dart.dev/guides>

<https://dartpad.dev/?null_safety=true>

<https://medium.com/flutter-comunidade-br/introdu%C3%A7%C3%A3o-a-linguagem-de-programa%C3%A7%C3%A3o-dart-b098e4e2a41e>

<https://dev.to/madebyluque/introducao-a-programacao-com-dart-aji>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Operadores Aritméticos - Continuação**

**Revisando:**

**Operadores aritméticos**

O Dart suporta os operadores aritméticos usuais, conforme mostrado na tabela a seguir.

**Operador Significado**

**+ Adicionar**

**– Subtrair**

**-expr Unário menos, também conhecido como negação (inverter o sinal da expressão)**

**\* Multiplicar**

**/ Dividir**

**~ / Divide, retornando um resultado inteiro**

**% Obtém o restante de uma divisão inteira (módulo)**

|  |
| --- |
| **Exemplo:**  **assert (2 + 3 == 5);**  **assert (2 – 3 == -1);**  **assert (2 \* 3 == 6);**  **assert (5/2 == 2,5); // Resultado é um duplo**  **assert (5 ~ / 2 == 2); // Result é um int**  **assert (5% 2 == 1); // Restante**  **assert (‘5/2 = $ 5 ~ / 2 r $ 5% 2’ == ‘5/2 = 2 r 1’);** |

**Operadores Lógicos e Relacionais**

Os operadores lógicos são importantes no Dart porque permitem comparar variáveis e fazer algo com base no resultado dessa comparação. Quando estamos estudando lógica de programação é comum termos dúvidas nos operadores. Entre os operadores, os que mais geram dúvidas são os operadores lógicos.

Por trás dos operadores lógicos existe uma série de conhecimentos teóricos, porém esse não será o foco da nossa aula. O objetivo aqui é bem mais prático, falando diretamente para utilização de operadores lógicos em programação.

**Valores Booleanos (Verdadeiro e Falso)**

Em programação existe um tipo de dado **chamado Booleano**. Esse tipo guarda uma informação que indica **se algo é verdadeiro ou falso**.

Um exemplo!

Vamos supor que você receba a idade do usuário do seu programa e deseja saber **se ela é maior que 18 anos.** O resultado da análise da idade só pode retornar dois valores: **verdadeiro, se ele tiver mais de 18 anos, ou falso, se ele tiver menos de 18.**

No caso acima podemos ter uma visão prática do uso do dado booleano, apesar de não envolver diretamente operadores lógicos.

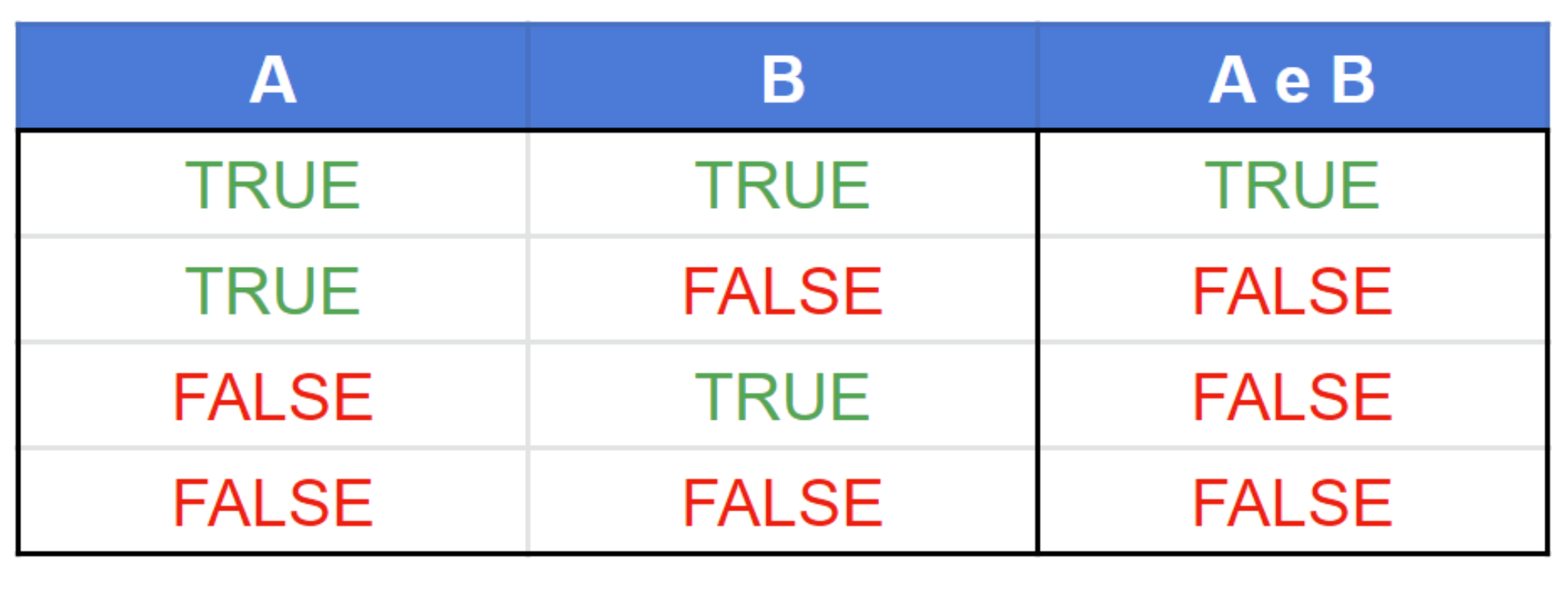
**Onde entram os operadores lógicos e relacionais???**

Os operadores lógicos entram quando precisamos realizar operações sobre um ou dois valores **booleanos (verdadeiro ou falso)**. Vamos pensar de maneira prática novamente. Imagine que agora nós precisamos verificar se a idade do usuário é maior que 18 anos e menor que 22. Pensando em como resolver o problema acima, precisamos realizar 2 comparações, uma para saber se a idade é maior que 18 e outra para saber se a idade é menor que 22.

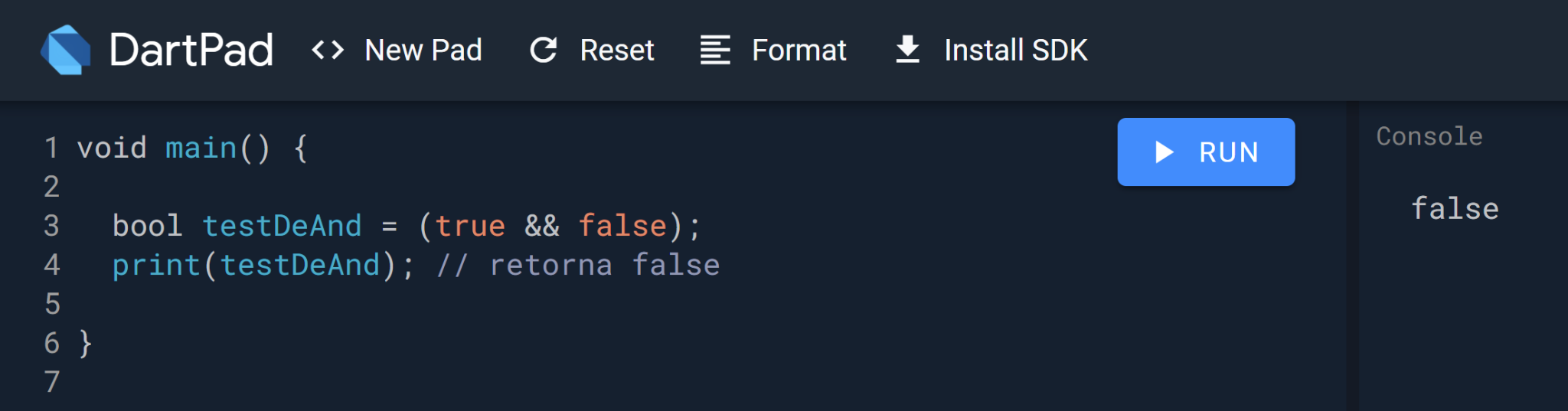
|  |
| --- |
| **Sabemos que cada operação dessa retornará um valor verdadeiro ou falso, dependendo da idade do usuário, mas ainda não temos a resposta se a idade está entre 18 e 22.**  **Para realizar essa análise precisamos realizar uma operação sobre os valores booleanos que foram retornados pelas comparações e para isso usamos um operador lógico.** |

**Operador E (AND) - &&**

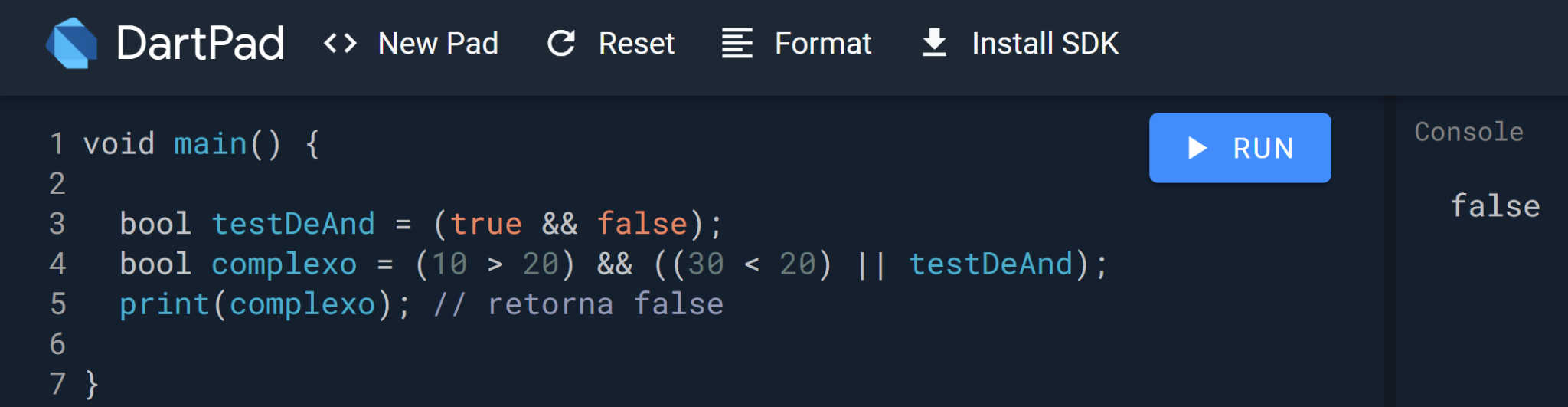
O primeiro operador lógico que veremos será o operador **E**. Este operador retorna verdadeiro quando os dois valores de entrada passados para ele são verdadeiros, caso contrário ele retorna falso. Veja a tabela verdade dele abaixo:



**Operador usado no código fonte: &&**

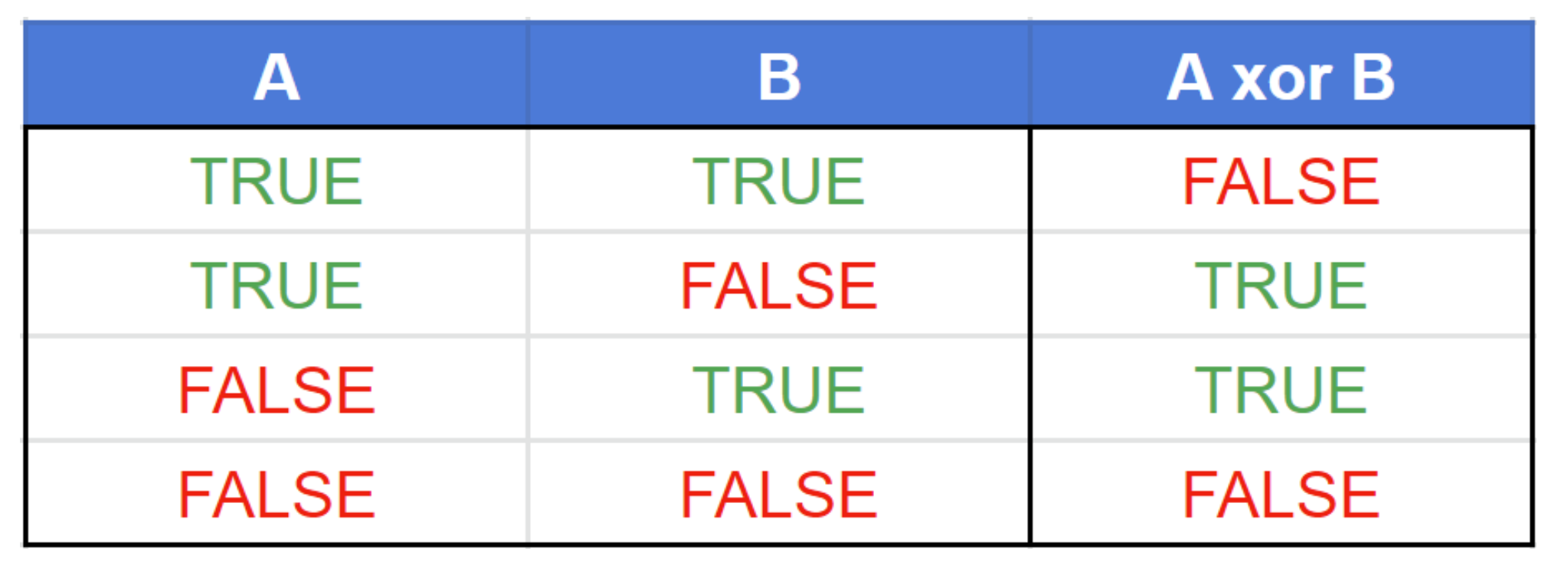


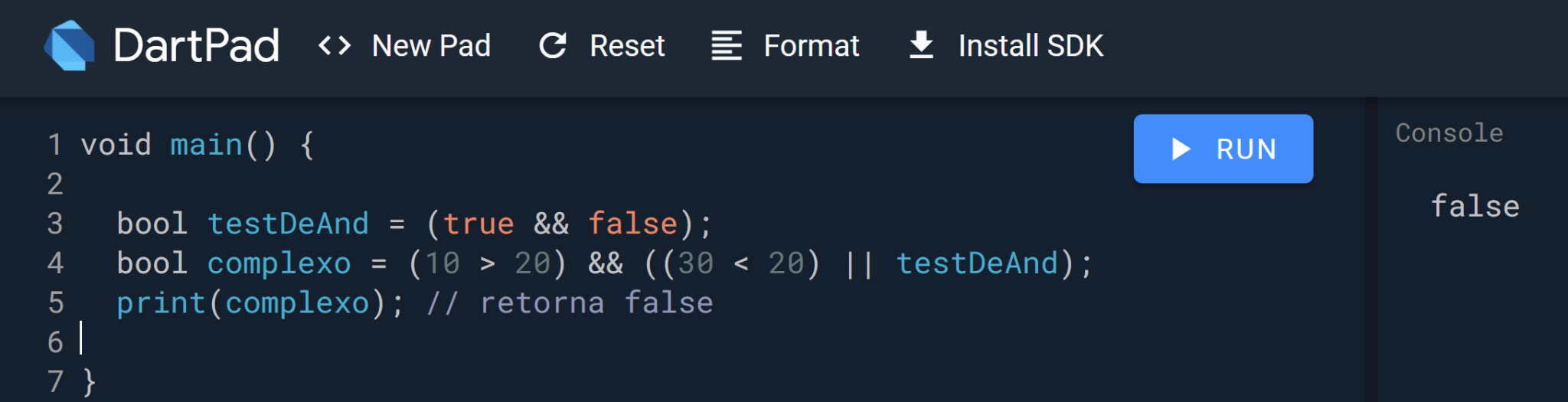
**Usando && e ||:**



**Operador OU - ||**

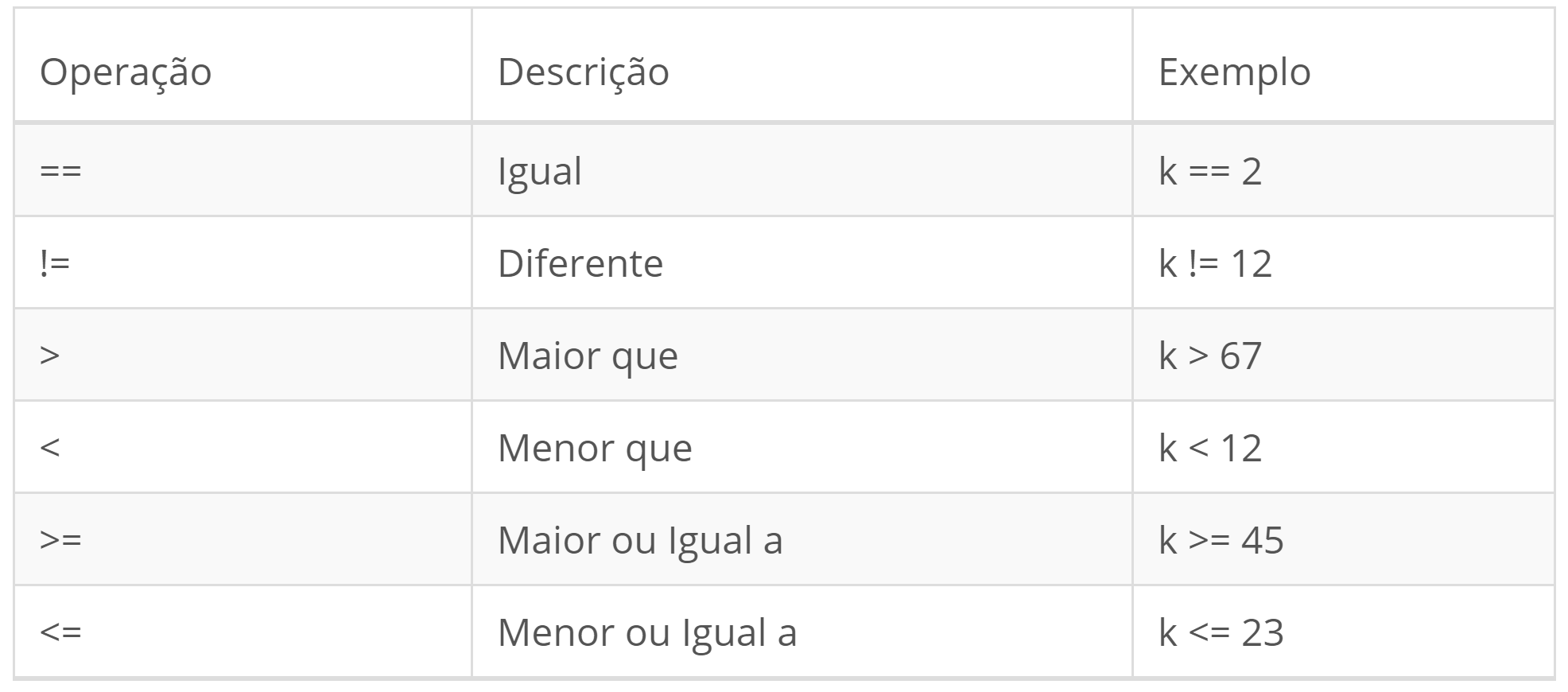
O operador OU Exclusivo retornará verdadeiro quando um dos dois valores é verdadeiro, porém retorna falso se os 2 valores forem verdadeiros. Ele também retorna falso se os dois valores de entrada forem falsos.





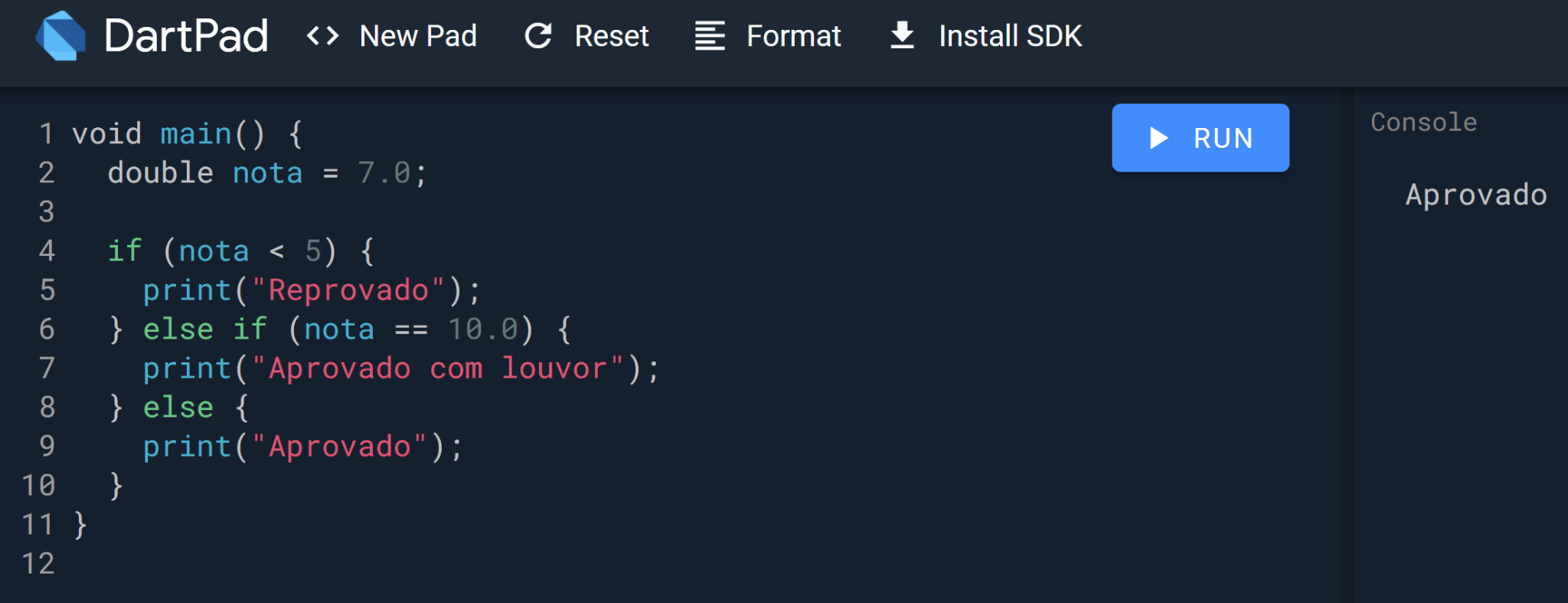
**Operadores Relacionais**

Os seguintes operadores podem ser usados para comparações no Dart. O resultado de cada expressão é sempre um **bool**, o que significa que é true ou false - verdadeiro ou falso.



Sempre precisamos tomar decisões, para isso podemos utilizar condicionais.

Exemplo:

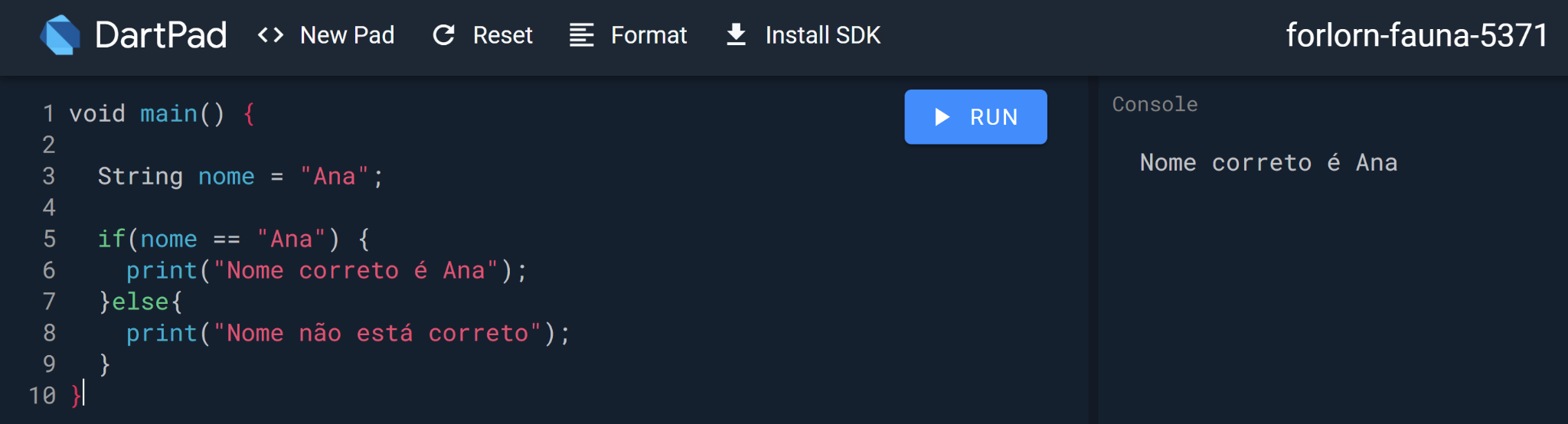


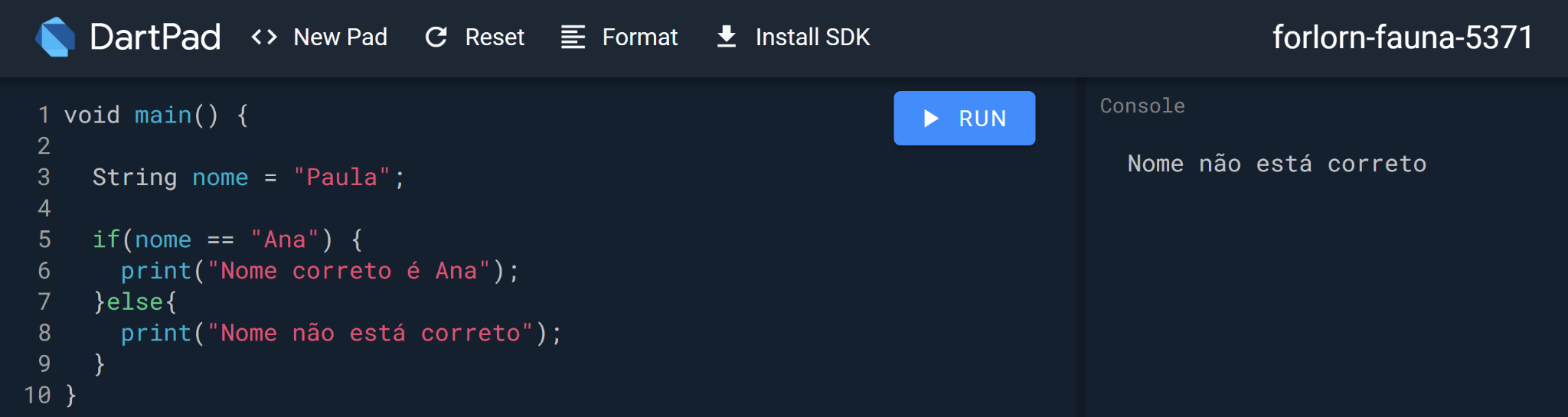
**IF Else:**

**Escolhendo o fluxo com if e else**

A primeira estrutura de controle de fluxo que veremos se chama if e ela permite executar um trecho de código caso uma expressão lógica seja verdadeira, ou ignorá-lo se ela for falsa.

Por exemplo, quando usamos o operador ternário, que nos permite selecionar entre um de dois valores. A sintaxe desse operador é a seguinte:





**Swicth Case**

Switch é outro tipo de condicional que é muito utilizado quando temos constantes. É interessante para usar o switch quando temos uma variável que assume valores definidos. Em regra, deve-se utilizá-la com case (define qual o valor da variável) e breck (termina o comando). E o mais legal e diferencial de outras linguagens é que o O Dart suporta cláusulas de case vazias!

