Manual do ArduBlock – Testes

* “Maior que” ( > )



Figura 1 – Bloco Maior que

Este bloco é chamado de “maior que” porque ele retorna verdadeiro para um teste, caso o número do primeiro encaixe for maior que o do segundo encaixe.

Exemplo: No exemplo da Figura 2 estamos usando um bloco se/senão que foi descrito no manual da parte de Controle e dois Leds, um conectado na porta D13 e outro na porta D12, e o nosso bloco “maior que” irá verificar se o primeiro número é maior que o segundo, nesse caso o número 5 é sim maior que o número 1, e isso irá retornar que esse teste é verdadeiro então ele irá acender o LED verde, se caso o segundo número for maior que o primeiro ele irá retornar o valor de falso e então o LED verde irá se manter apagado e o LED vermelho irá acender.

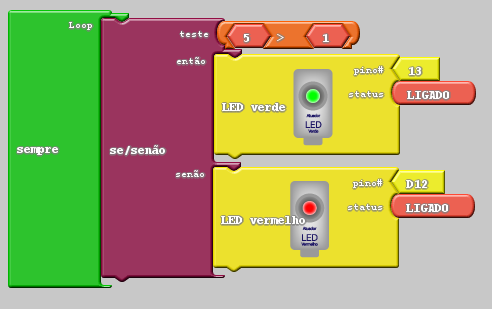


Figura 2 – Exemplo da Aplicação do Bloco “Maior que”

* “Menor que” ( < )



Figura 3 – Bloco Menor que

Este bloco é chamado de “menor que” porque ele retorna verdadeiro para um teste, caso o número do primeiro encaixe for menor que o do segundo encaixe.

Exemplo: No exemplo da Figura 4 estamos usando um bloco se/senão que foi descrito no manual da parte de Controle e dois Leds, um conectado na porta D13 e outro na porta D12, e o nosso bloco “menor que” irá verificar se o primeiro número é menor que o segundo, nesse caso o número 5 é sim menor que o número 10, e isso irá retornar que esse teste é verdadeiro então ele irá acender o LED verde, se caso o segundo número for menor que o primeiro ele irá retornar o valor de falso e então o LED verde irá se manter apagado e o LED vermelho irá acender.

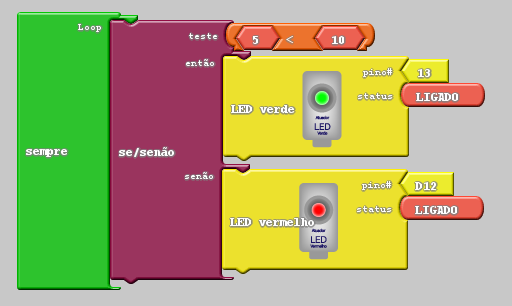


Figura 4 – Exemplo da aplicação do Bloco “Menor que”

* igualdade ( = )

Este bloco é chamado de “igual” porque ele retorna verdadeiro para um teste caso o bloco do primeiro encaixe for igual ao bloco do segundo encaixe. Mas notamos que temos quatro tipos diferentes de encaixe para os blocos de igualdade.

* O primeiro que está representado na Figura 5.A é para utilizar quando queremos comparar dois números, independente se são números inteiros, negativos ou até mesmo fracionários.
* Já na Figura 5.B a comparação é feita para as variáveis booleanas que podem ser divididos em dois tipos, Alto (HIGH) ou Baixo (LOW).
* A Figura 5.C é a comparação entre uma letra e outra letra, simplesmente um caractere.
* E por último na Figura 5.D­­ a comparação é feita entre Strings, que podem ser representadas por palavras contendo mais que apenas um caractere.



Figura 5.A



Figura 5. B

Figura 5. D

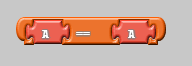


Figura 5. C

* “Maior ou igual que” ( >= )



Figura 6 – Bloco Maior ou Igual que

Este bloco é chamado de “Maior ou igual que” porque ele retorna verdadeiro para um teste caso o número do primeiro encaixe for maior ou igual ao número do segundo encaixe.

Exemplo: No exemplo da Figura 8 estamos usando um bloco se/senão que foi descrito no manual da parte de Controle e dois Leds, um conectado na porta D13 e outro na porta D12, e o nosso bloco “maior ou igual que” irá verificar se o primeiro número é igual que o segundo, nesse caso o número 5 é sim igual ao número 5, e isso irá retornar que esse teste é verdadeiro então ele irá acender o LED verde, se caso o segundo número for maior que o primeiro ele irá retornar o valor de falso e então o LED verde irá se manter apagado e o LED vermelho irá acender.

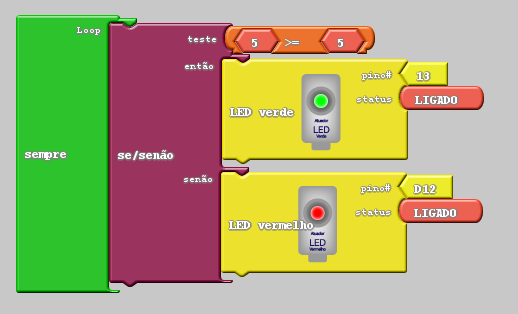


Figura 7 – Exemplo de Aplicação do Bloco Maior ou Igual que

* “Menor ou igual que” ( <= )



Figura 8 – Bloco Menor ou igual que

Este bloco é chamado de “Menor ou igual que” porque ele retorna verdadeiro para um teste caso o número do primeiro encaixe for menor ou igual ao número do segundo encaixe.

Exemplo: No exemplo da Figura 10 estamos usando um bloco se/senão que foi descrito no manual da parte de Controle e dois Leds, um conectado na porta D13 e outro na porta D12, e o nosso bloco “menor ou igual que” irá verificar se o primeiro número é igual ou menor que o segundo, nesse caso o número 5 é sim igual ao número 5, e isso irá retornar que esse teste é verdadeiro então ele irá acender o LED verde, o primeiro número poderia ser também um número menor que 5, por exemplo o 4, 3, 2 entre outros até mesmo números negativos. Se caso o primeiro número for maior que o segundo ele irá retornar o valor de falso e então o LED verde irá se manter apagado e o LED vermelho irá acender.

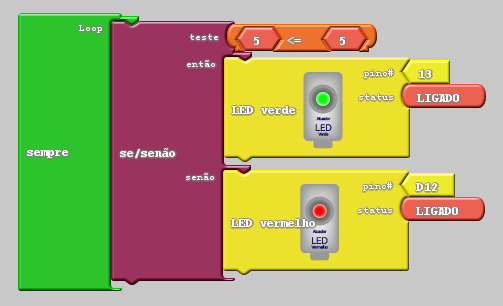


Figura 9 – Exemplo de Aplicação do Bloco Menor ou Igual que

* “Diferente que” ( != )

Este bloco é chamado de “Diferente que” porque ele retorna verdadeiro para um teste caso o bloco do primeiro encaixe for diferente ao bloco do segundo encaixe.

* O primeiro que está representado na Figura 11.A é para utilizar quando queremos comparar a diferença de dois números, independente se são números inteiros, negativos ou até mesmo fracionários.
* Já na Figura 11.B a comparação da diferença é feita para as variáveis booleanas que podem ser divididos em dois tipos, Alto (HIGH) ou Baixo (LOW).
* A Figura 11.C é a comparação da diferença entre uma letra e outra letra, simplesmente um caractere.
* E por último na Figura 11.D a comparação da diferença é feita entre Strings, que podem ser representadas por palavras contendo mais que apenas um caractere.



Figura 10.A



Figura 11.B



Figura 11.C



Figura 11.D

* “E” e “OU”

O bloco “E” é um operador lógico, e resulta em **verdadeiro** apenas se ambos os dois encaixes retornarem verdadeiro, caso contrário ele retorna **falso**. Ou seja, os dois testes precisam ser **verdadeiros** para que o teste E seja verdadeiro.

O Bloco “OU” é um operador lógico, e resulta em verdadeiro quando **UM** dos dois encaixes retorne o estado de verdadeiro, e esse bloco só retornará falso quando os dois encaixes retornarem à condição de falso também.

No exemplo da figura 11 pode ser observado os dois blocos “e” e o bloco “ou”, o bloco “e” só irá retornar verdadeiro quando o pino digital D2 e o D3 estiverem em estado ALTO, caso um deles esteja em estado BAIXO, o bloco “e” retorna falso.

O bloco “ou” irá retornar verdadeiro quando o pino D2 estiver em estado ALTO, OU o pino D3 estiver em estado alto, nesse caso apenas um deles precisa estar em estado ALTO, caso os dois estejam em estado BAIXO, o bloco “ou” retorna falso.

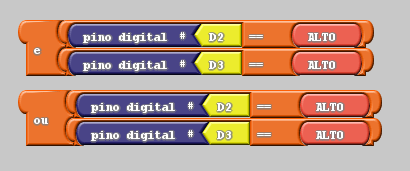


Figura – Bloco “E” e bloco “OU”

* “Não”

O bloco “Não” é um operador lógico, que resulta em verdadeiro se o operador dentro do encaixe for falso, e caso a operação dentro do encaixe for falso, o retorno do bloco “Não” é verdadeiro. Fazendo o contrário do que está recebendo.



Figura 12 – Bloco “Não”

Exemplo: Este exemplo está sendo representado pela Figura 13 onde estamos trabalhando agora com um botão conectado na porta D2, também temos dois LEDS um conectado na porta D12 e outro na D13, mas principalmente queremos entender o bloco **não**, e para isso vamos usar juntamente o bloco de igualdade de booleanos, e usando o exemplo da figura 6.2, se o botão for pressionado irá acender o LED verde e o vermelho irá ficar apagado, caso o botão não esteja pressionado o LED verde irá ficar apagado e o vermelho irá acender, mas arrumando um bloco **não** na frente do bloco de igualdade, podemos observar que se não pressionarmos o botão o LED verde irá acender e o vermelho apagado e quando pressionamos o mesmo, o contrário com os LEDs acontece.

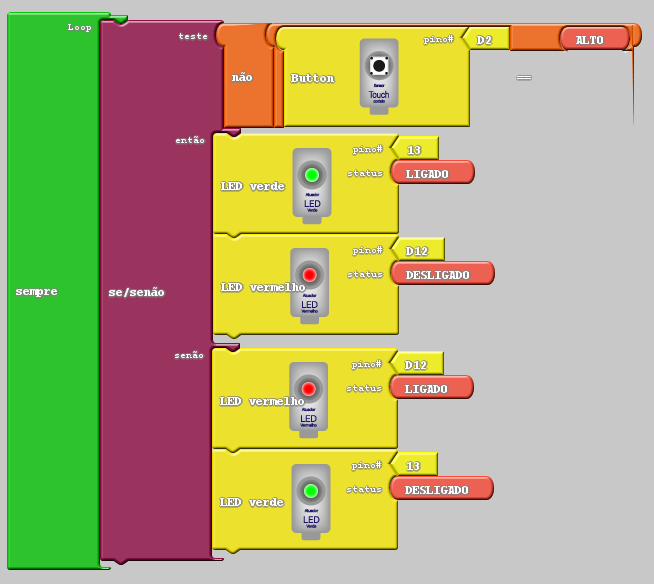


Figura 13 - Exemplo da aplicação do bloco “Não”

* “Compara com Inteiro”

Este bloco irá converter primeiramente o valor de um número em forma de String para um valor inteiro, e depois disso vai retornar verdadeiro caso o número do primeiro encaixe seja igual ao número do segundo encaixe.

String: para esclarecer o entendimento a String é representada em sequencias de caracteres e não pode ser feito operações matemáticas dessa forma, por isso primeiro fazemos a conversão para um valor inteiro.

Números Inteiros: São aqueles números que não apresentam casas decimais, ou seja, números após a vírgula, exemplos são: 1, 2, 3, 4, 5 e assim por diante. Pode ser considerado números inteiros tanto positivo quanto negativo.



Figura 14 - Bloco "compara com inteiro"

Exemplo: Este exemplo está sendo representado pela Figura 15 onde estamos trabalhando agora com dois LEDS um conectado na porta D12 e outro na D13, mas principalmente queremos entender o bloco **compara com inteiro**, no primeiro encaixe podemos observar que podemos arrumar qualquer número inteiro possível, mas com esse número não é possível fazer nem uma operação matemática ou comparativa com ele, pois seu formato está como String, e para que possamos comparar com outro número inteiro devemos primeiro fazer a conversão dele, e é isso que este bloco faz, ele converte a String para Inteiro e faz a comparação com o segundo encaixe, se os dois números forem iguais, ele retorna verdadeiro e se forem diferente ele irá retornar falso, neste exemplo temos que 2 é diferente de 1, então irá retornar falso e o LED vermelho irá se acender e o LED verde irá ficar apagado.

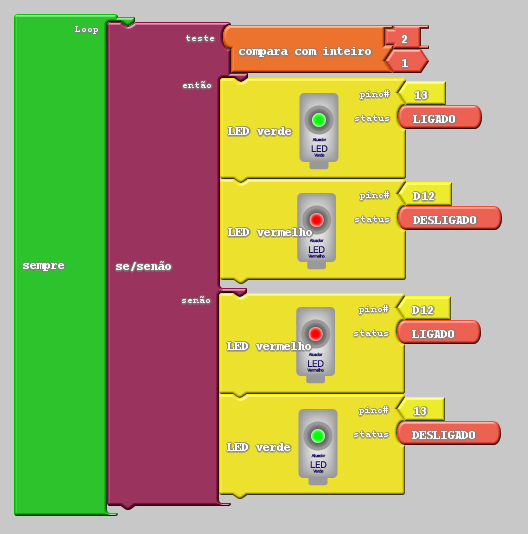


Figura 15 – Exemplo da Aplicação do Bloco “compara com inteiro”