
AGENDA 14

TÉCNICAS DE
PROGRAMAÇÃO
NO KODULAR



GEEaD - Grupo de Estudos de Educação a Distância
Centro de Educação Tecnológica Paula Souza

GOVERNO DO ESTADO DE SÃO PAULO
EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃO
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS
PROGRAMAÇÃO MOBILE I

Expediente

Autor:

GUILHERME HENRIQUE GIROLI

Atualização técnica:

Rogério Galdiano de Freitas

Revisão Técnica:

Eliana Cristina Nogueira Barion

Revisão Gramatical:

Juçara Maria Montenegro Simonsen Santos

Editoração e Diagramação:

Flávio Biazim



Utilização de métodos

Em determinados momentos da programação de aplicativos, encontramos a necessidade de repetir inúmeras vezes o mesmo código ou trechos da programação em vários locais diferentes do nosso programa. Para evitar que o programador reescreva várias vezes os mesmos códigos em locais diferentes, utilizamos os procedimentos.

Um exemplo de método é a função de “trocaJogador”. Essa função é chamada todas as vezes que um jogador escolhe uma posição do tabuleiro, e passa a jogada para o seu adversário.



Figura 1 - Procedimento trocaJogador

Os procedimentos possuem uma visibilidade, um tipo de retorno e argumentos. A visibilidade “public” é responsável por deixar o método acessível para ser utilizado em qualquer parte do nosso app. Sendo que na plataforma Kodular todos os procedimentos serão públicos.

Cada procedimento pode gerar um valor após a sua execução, para isso é necessário informar qual é o tipo de retorno, como por exemplo, um valor inteiro. No caso do nosso procedimento “trocaJogador” não é necessário retornar nada após a sua execução.

Os argumentos são declarados no próprio procedimento. Os argumentos de um método têm a função de transmitir valores para as rotinas que serão executadas internamente no procedimento. Para a função de “trocaJogador” não foi necessário passarmos nenhum argumento para a sua execução. Mas no procedimento marca_posição utilizamos um argumento btn, que determina qual o botão que o usuário clicou.

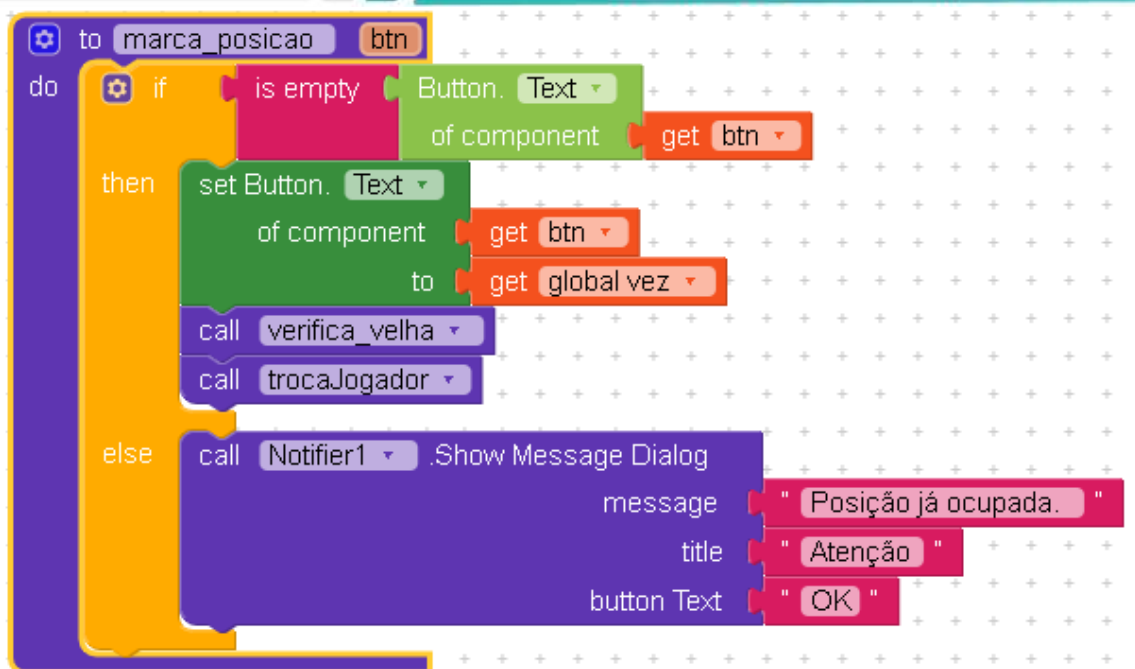


Figura 2 - Exemplo de procedimento com argumento.

Para utilizar um método basta apenas chamar o seu nome, como podemos observar no código que é executado quando o botão “**btn_Comecar**” é pressionado.

Durante a programação de aplicativos nos deparamos com a manipulação de dados na programação. Esses dados obedecem a um tipo e podem ser utilizados para armazenar valores em variáveis e/ou constantes, retornos de métodos, entre outras utilizações.

Encontramos alguns tipos que são utilizados com uma maior frequência no desenvolvimento mobile.

Variável

Variável é um local para armazenamento temporário na memória do dispositivo indicado por um nome previamente escolhido no momento da sua declaração. No ambiente de desenvolvimento Kodular não é necessário um tipo para a variável, A função principal da variável é armazenar valores temporários que serão utilizados nas rotinas do aplicativo, esses valores podem sofrer alterações durante a execução do aplicativo. Veja a seguir o código para gerar uma variável.



Figura 3 - Criando uma variável.

Vetor

Uma variável armazena apenas um valor para cada alocação. Para trabalhar com vários valores de um mesmo tipo relacionados há um processamento comum podemos utilizar um vetor. O vetor é indicado por um nome e um índice de acordo com a quantidade de divisões definidas no ato do seu desenvolvimento.

No exemplo a seguir encontramos uma variável para armazenar uma lista de valores.



Figura 4 - Criando uma lista de valores.

Na sequência declaramos um vetor para armazenar os nomes de três alunos de um grupo de estudos. É importante ressaltar que em um mesmo local vamos armazenar três nomes, separados internamente por um índice. O tamanho do vetor é definido de acordo com a necessidade da programação.



Figura 5 - Criando uma lista com três nomes.

O índice de um vetor sempre inicia no "0". Em uma lista de 3 posições vamos encontrar índices 0, 1, e 2.

Estrutura de decisão

A estrutura de decisão é utilizada para que uma parte do código seja executada apenas se uma determinada condição "booleana" estiver sendo atendida. Caso contrário podemos determinar que o aplicativo execute uma outra parte do código. O código a seguir demonstra a utilização de uma estrutura de decisão "if e else" para alterar o símbolo do jogador. Caso o jogador atual seja o "X", a condição será atendida e alteramos para o "O". Caso o jogador não é o "X", a condição não é atendida e executamos a alteração da variável para "X", uma vez que ela atualmente só pode ser o "O".



Figura 6 - Bloco de programação - trocaJogador.

Estrutura de repetição

Estrutura de repetição é utilizada para que uma parte do código seja executada e repetida durante uma determinada condição. É uma estrutura muito utilizada na programação e uma das suas principais funções é gerar uma economia de código em certas rotinas.

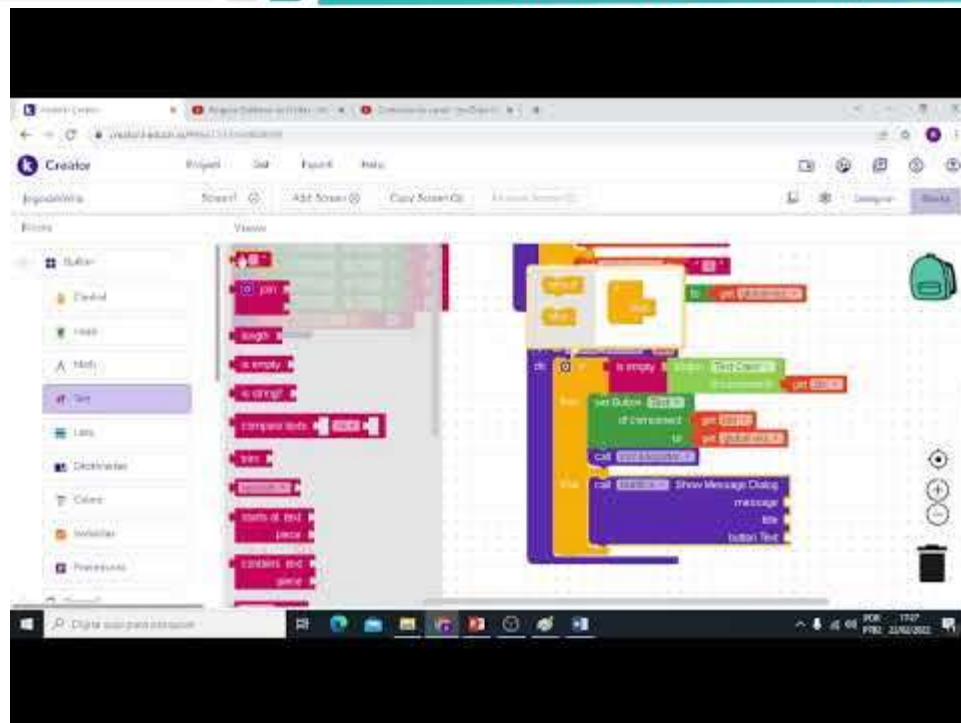


Figura 7- Exemplo de estrutura de repetição.

Agora iremos criar dois procedimentos que terão a função de alternar entre os jogadores e marcar o símbolo do jogador em cada botão clicado no tabuleiro. Portanto o aluno deverá assistir ao vídeo para compreender a programação dos procedimentos.

Agenda 14 - Procedimentos trocaJogador e marca_posicao ,disponível em :

<https://youtu.be/wdYw1HgOUZA>



- Clique no construtor **Procedure** e arraste o objeto **to procedure do**

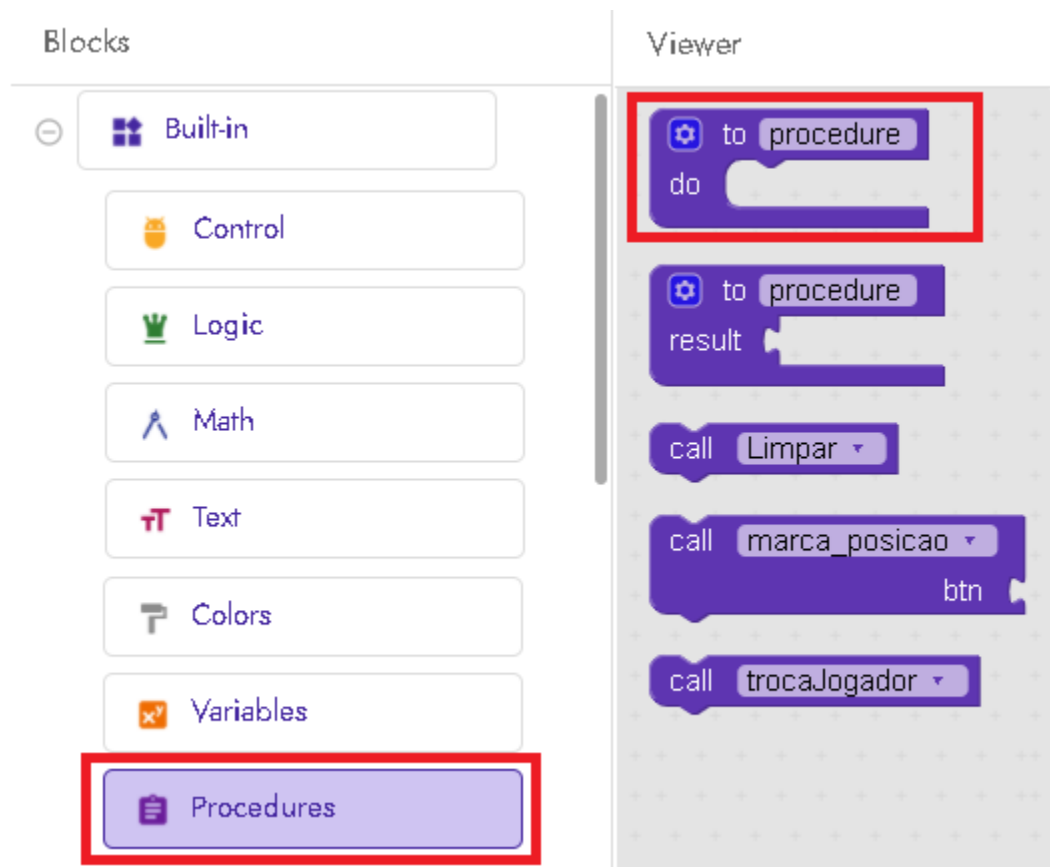


Figura 8 - Criando o procedimento verifica_velha

- Digite o **Verifica_velha** para o procedimento criado anteriormente.
- Clique no construtor **Control**, arraste o componente **IF** e encaixe no procedure **verifica_velha**.



Figura 9 - Inserindo o componente IF.

- Arraste a condição **elseif** por 8 vezes, dentro da condição IF.

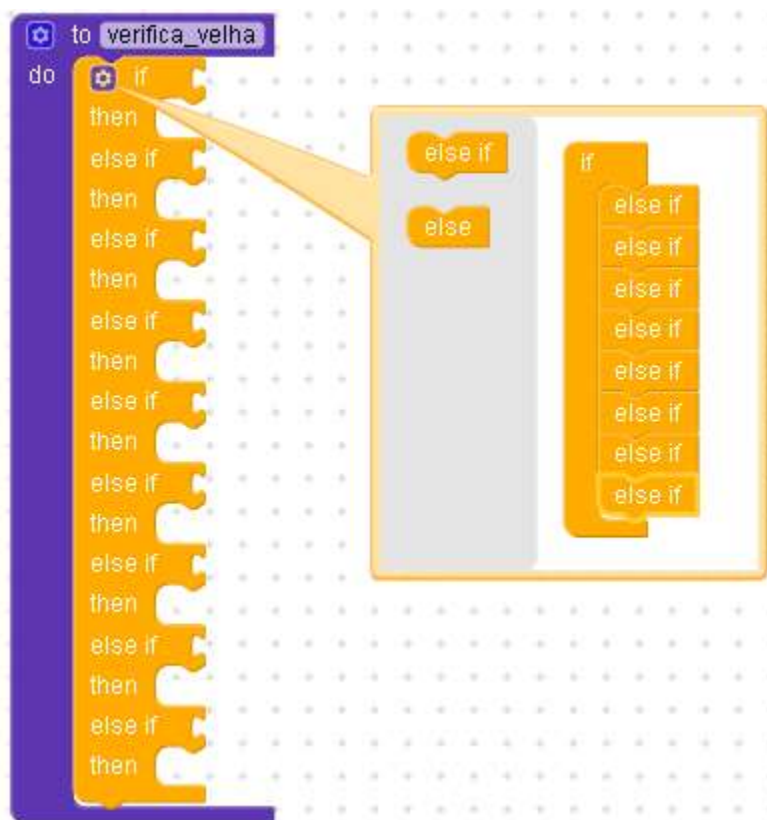


Figura 10 - Personalizando o procedimento verifica_velha.

- Clique no construtor **Logic**, arraste a opção **de teste de igualmente** e encaixe no bloco **IF**.



Figura 11 - Inserindo a condição no bloco de comando IF.

- Clique no construtor **Logic**, arraste novamente o componente de comparação e vamos encadear com a primeira estrutura de igualdade.

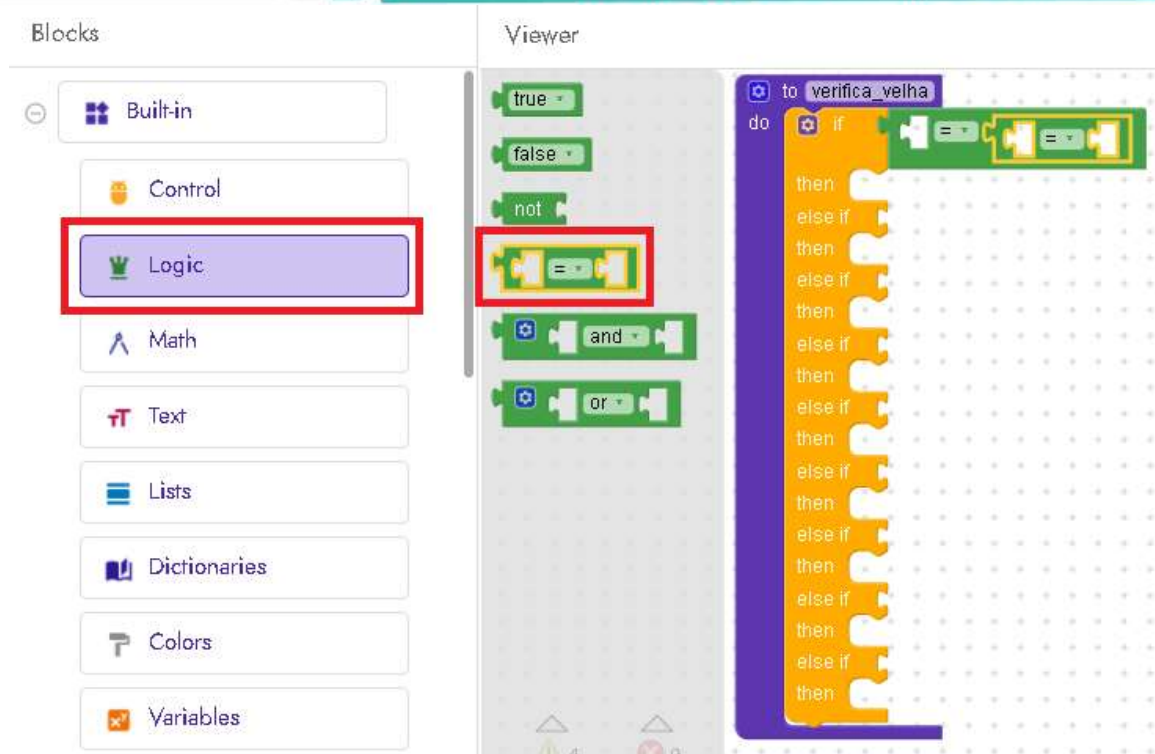


Figura 12- Inserindo o bloco de igualdade encadeada no bloco IF.

- O aluno deverá montar a primeira condição, como a **Figura 38**.

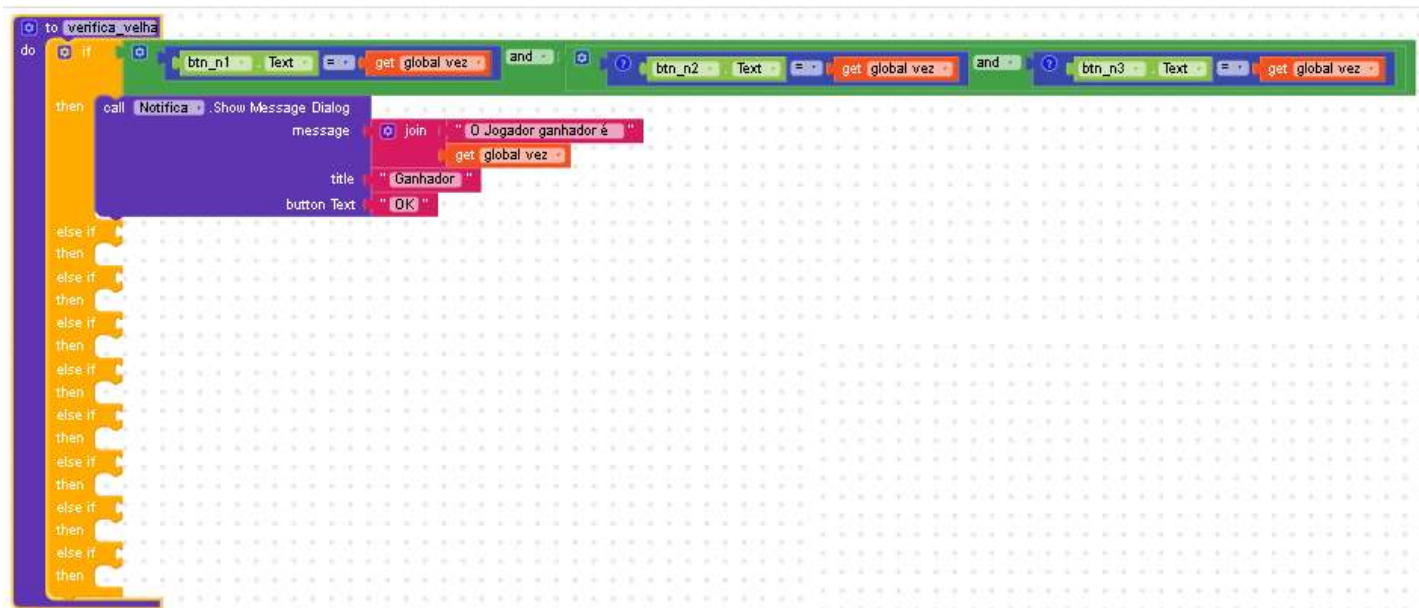


Figura 13 - Montagem do bloco do procedimento verifica_velha.

- Após a montagem do primeira condição, o aluno deverá construir todo o código que segue na **Figura 39**. Pois existe várias formas de ganhar o jogo, seja ela no sentido vertical, horizontal ou diagonal. Portanto a quantidade de código é um pouco grande, mas o resultado no final vale a pena.

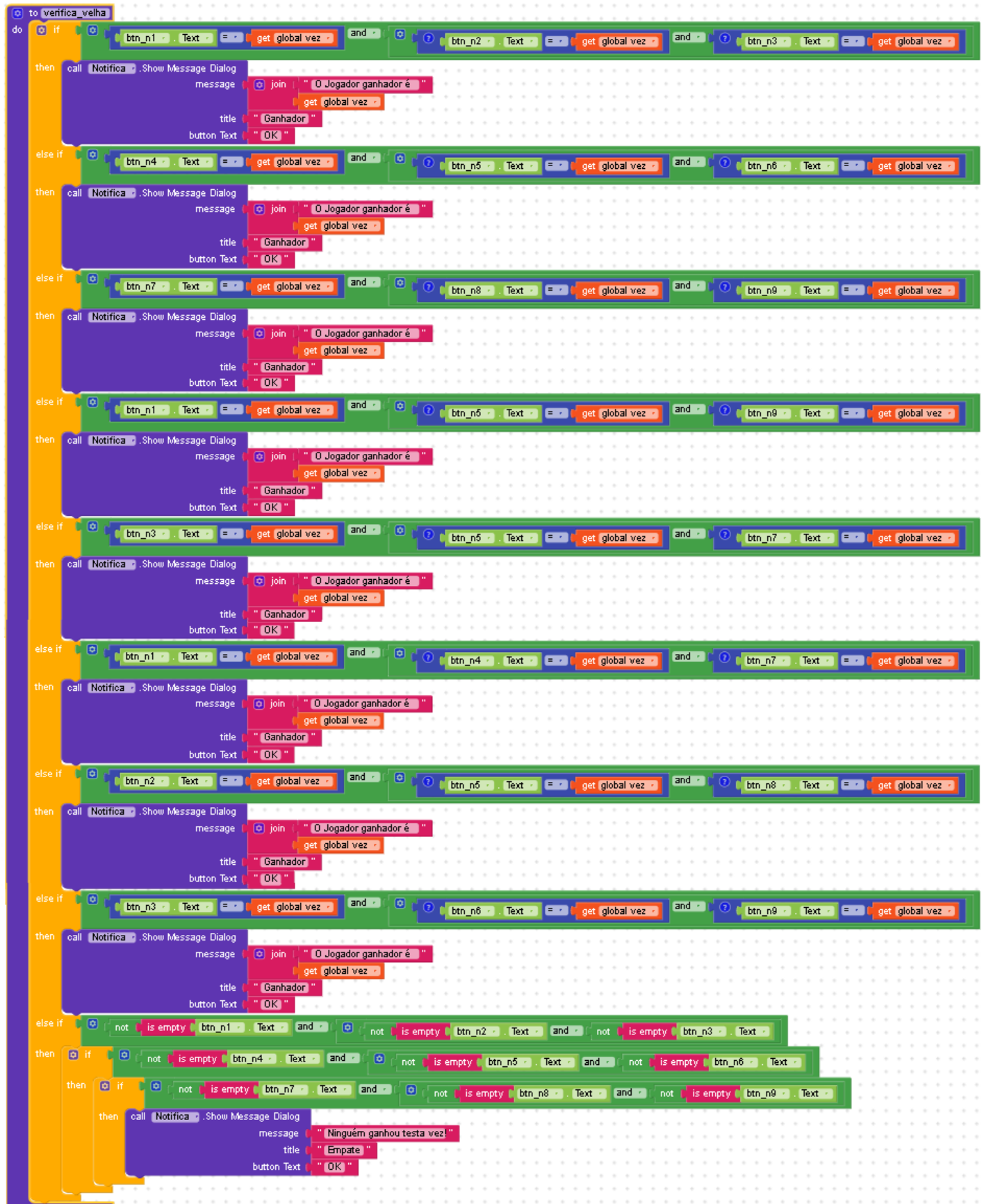


Figura 14 - Procedimento Verifica_Velha Completo.

Sendo assim, ao terminar o código do procedimento `verifica_velha`, o aluno deverá atualizar o código do procedimento **marca_posição**. Pois durante o jogo o aplicativo verifica se alguém ganhou a partida.

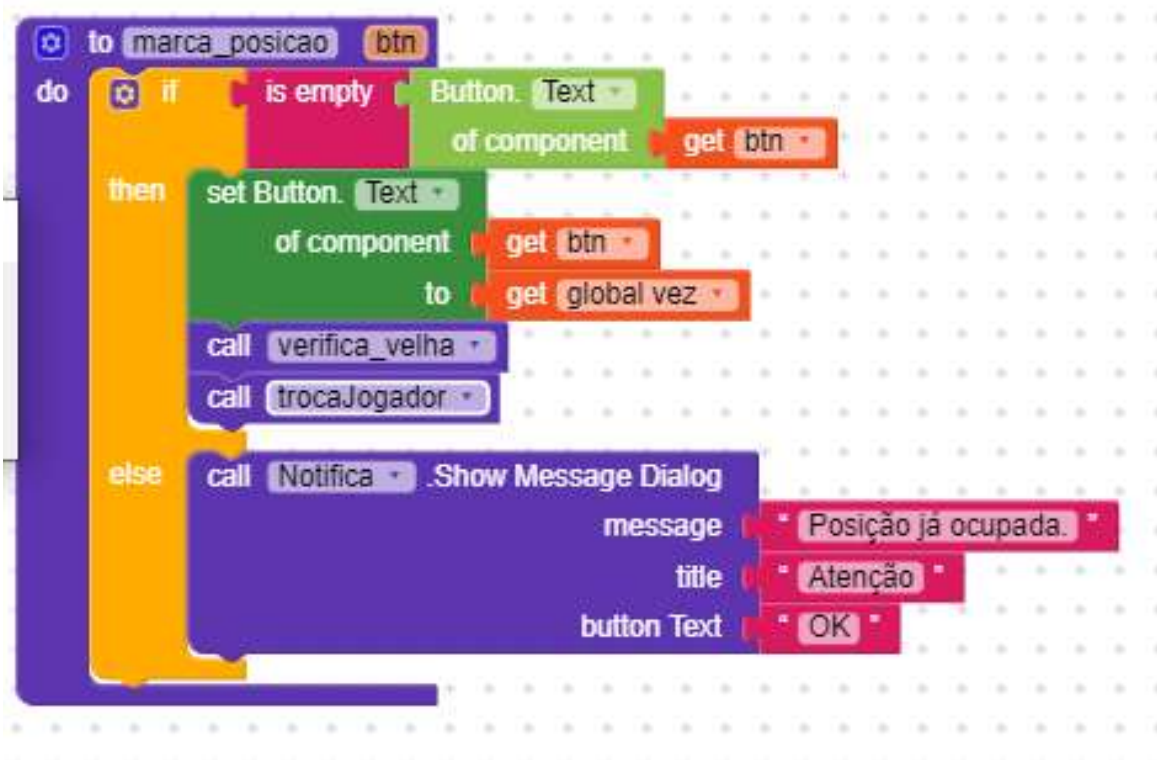


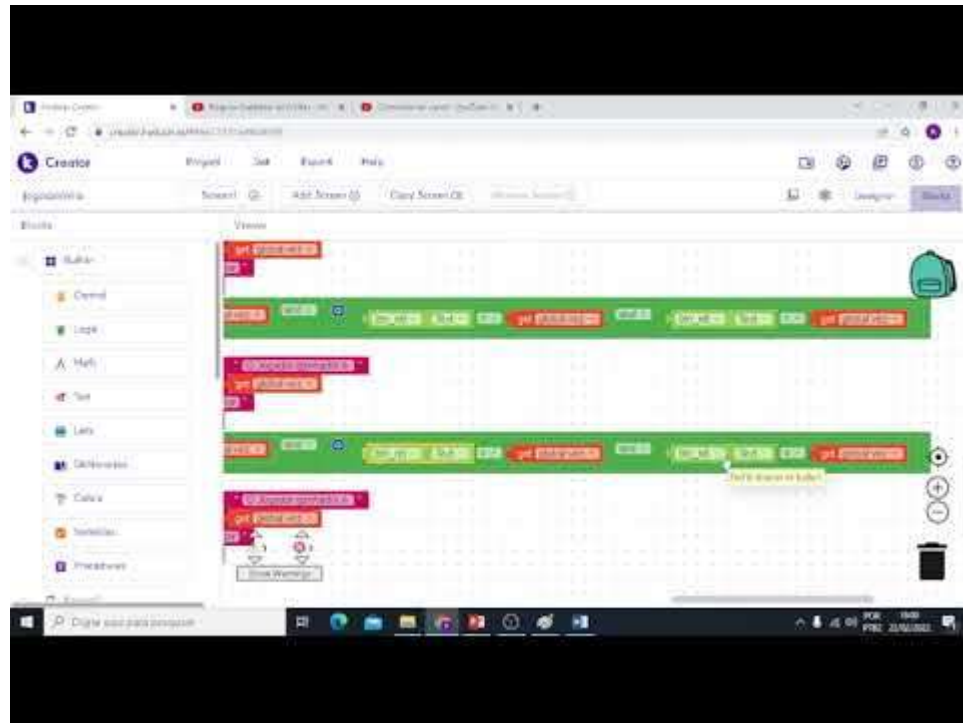
Figura 15 - Atualização procedimento `marca_posicao`.



Assista o vídeo a seguir para compreender algumas rotinas necessárias para o bom funcionamento do aplicativo jogo da velha.

Agenda 14 – Procedimento `verifica_velha`, disponível em:

<https://youtu.be/DifjY5Wa8pM>



Bons Estudos!

Confira o arquivo completo:

[Jogo da Velha.aia](#)