

COMPUTAÇÃO
NA ESCOLA



TUTORIAL DO JOGO
CAÇA MOSQUITO

OFICINA DE APP INVENTOR

O QUE VAMOS APRENDER?

JOGO CAÇA MOSQUITO

OBJETIVO

Caçar o mosquito voando pela tela

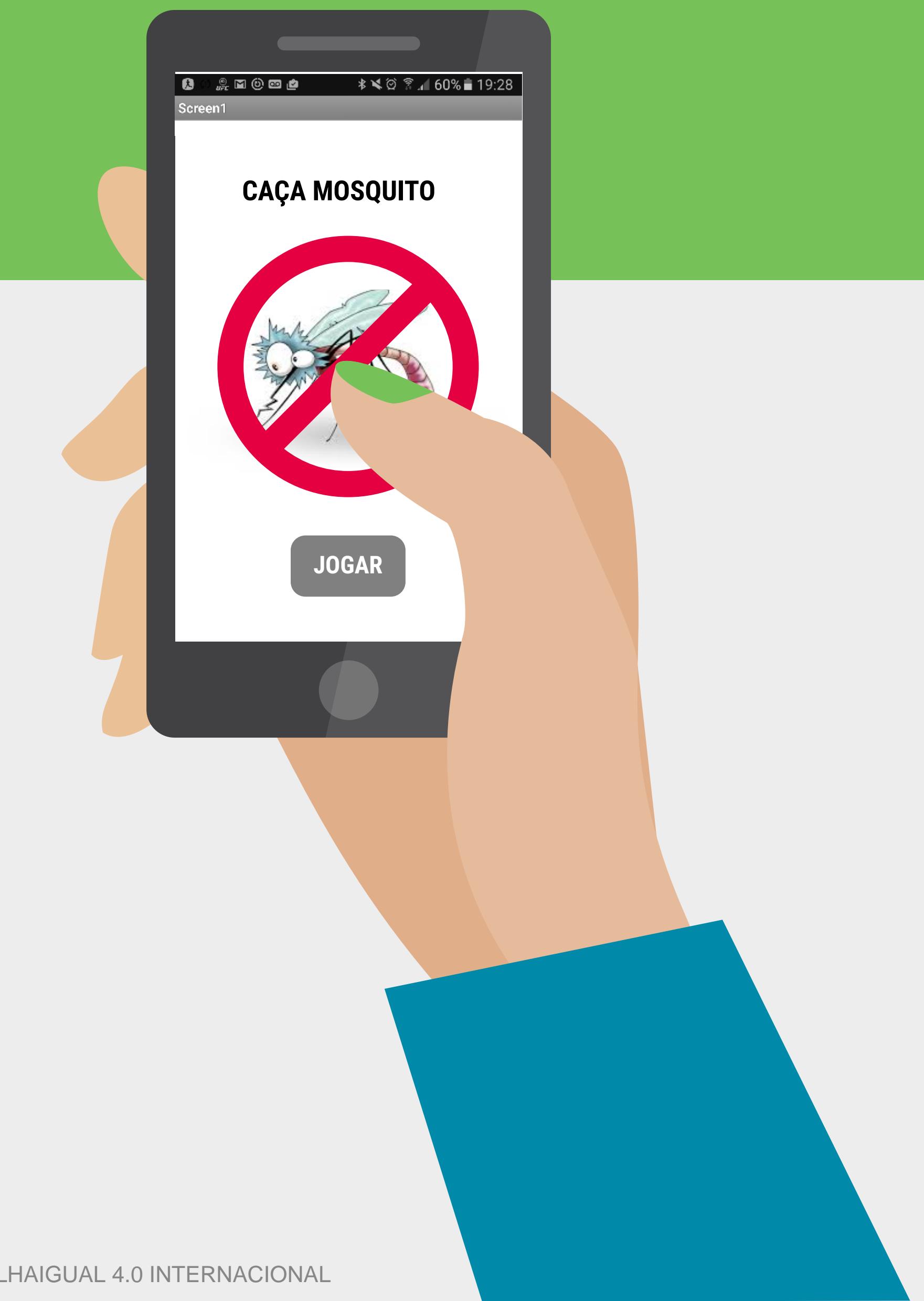
INFORMAÇÕES

Mosquito terá 3 vidas.

Jogador tem 10 segundos para eliminar mosquito.



Fazer um mosquito se mover pela tela.
Adicionar vida ao mosquito.
Adicionar um tempo para matar o mosquito.
Incluir níveis de dificuldade ao jogo.



O QUE É UM APP NO CELULAR?

“App” (abreviação de “aplicação”) é a palavra que descreve os programas desenvolvidos para dispositivos móveis, incluindo smartphones, tablets e até as tecnologias wearable – vestíveis – como os relógios. Existem uma grande variedade de categorias de aplicativos e em cada categoria temos aplicativos que fazem coisas diferentes.



APPS SÃO SOFTWARES

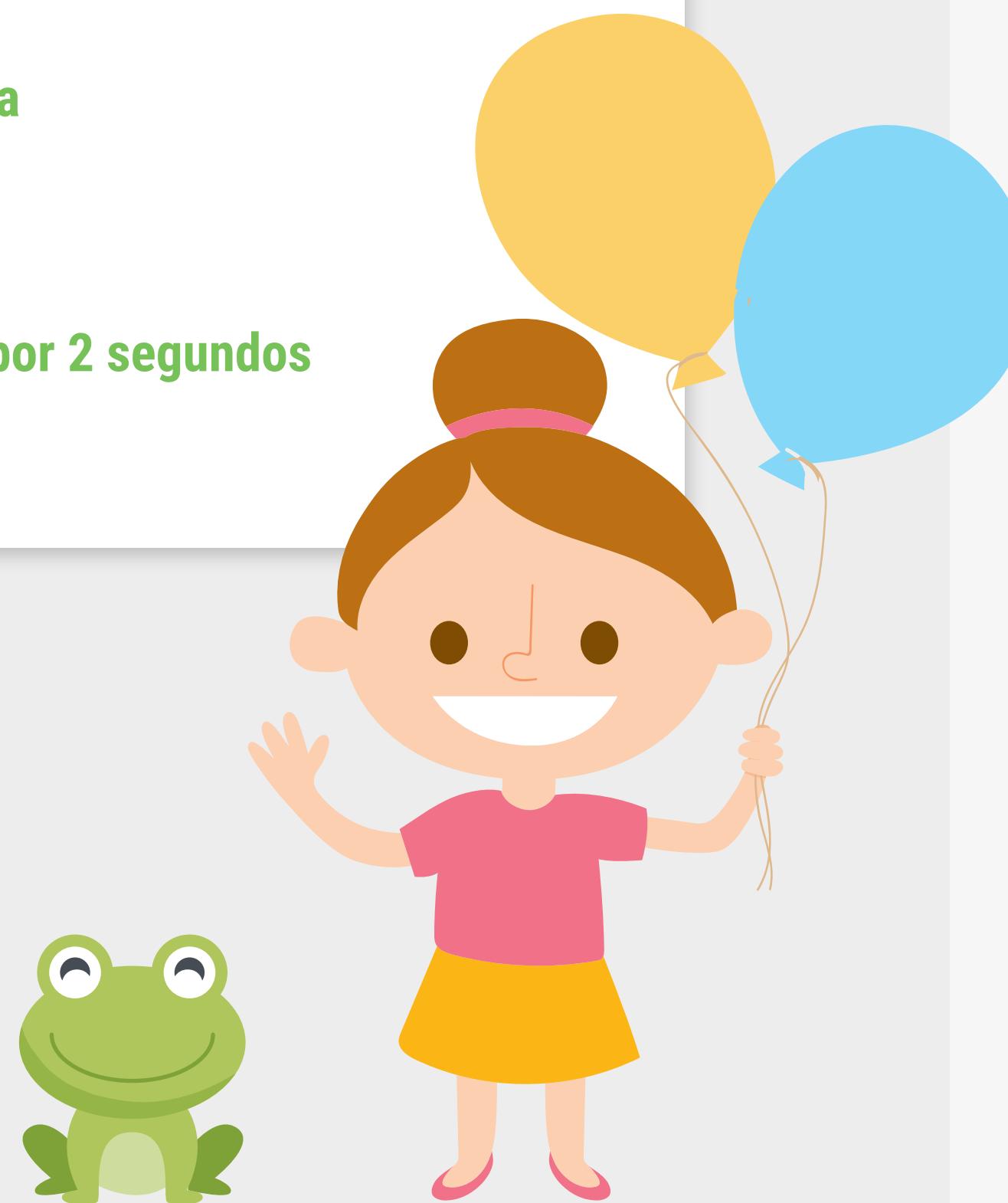
```
print("Railway placer 1.0 running.")  
print("How long would you like your railway?")  
length = tonumber(read())  
print("Creating railway "..length.." long.")  
print("Place fuel in slot 16 and rails")  
print("in slot 15. Press any key to begin.")  
os.pullEvent("char")  
if turtle.getFuelLevel() <= 5 then  
    turtle.select(16)  
    turtle.refuel(1)  
end  
function placeRail()  
    turtle.select(15)  
    turtle.placeDown()  
    turtle.forward()  
    sleep(0.1)  
end
```



ALGORITMO

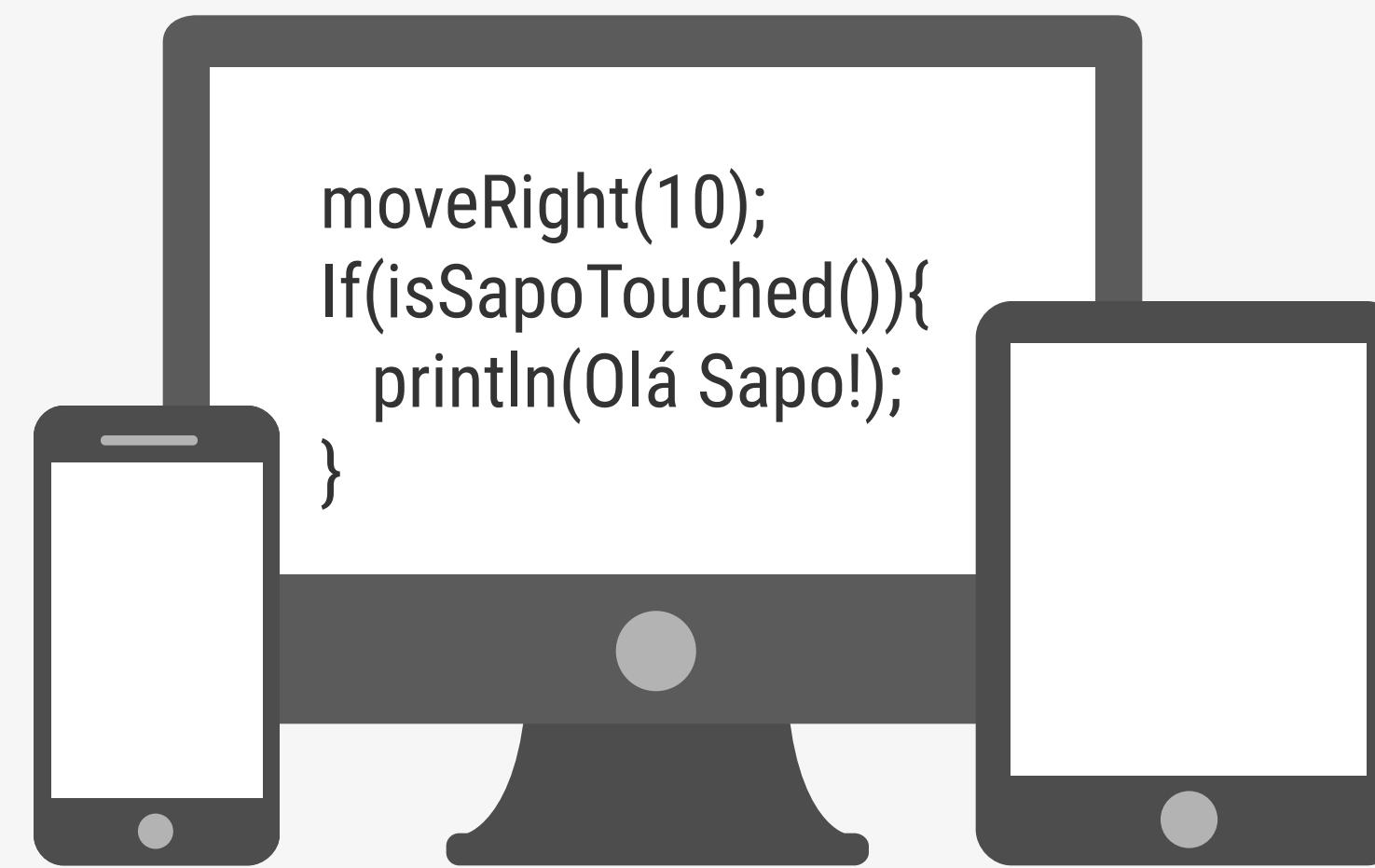
Um algoritmo é um conjuntos de instruções para realizar uma determinada tarefa.

- Ande 10 passos a direita
- Se toca no sapo
- Então diga “Olá sapo!” por 2 segundos

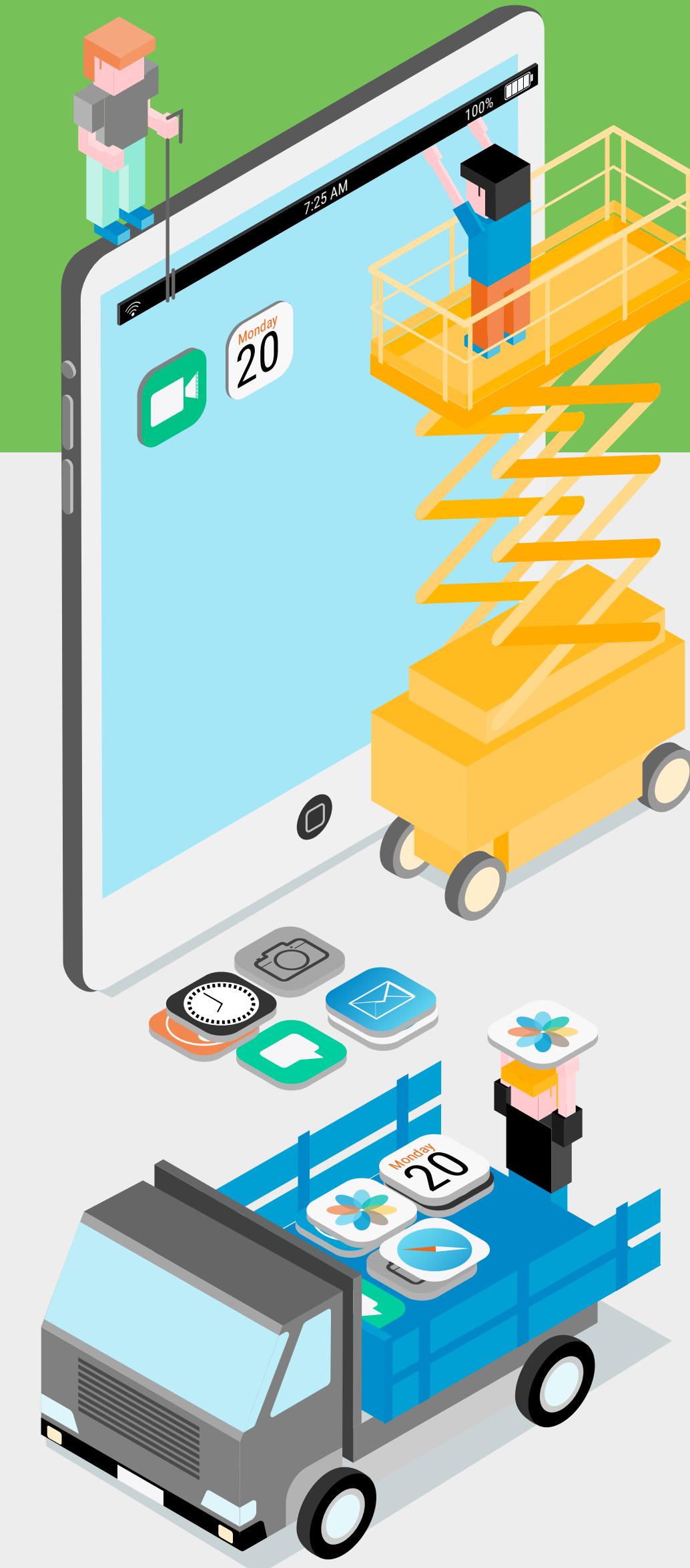
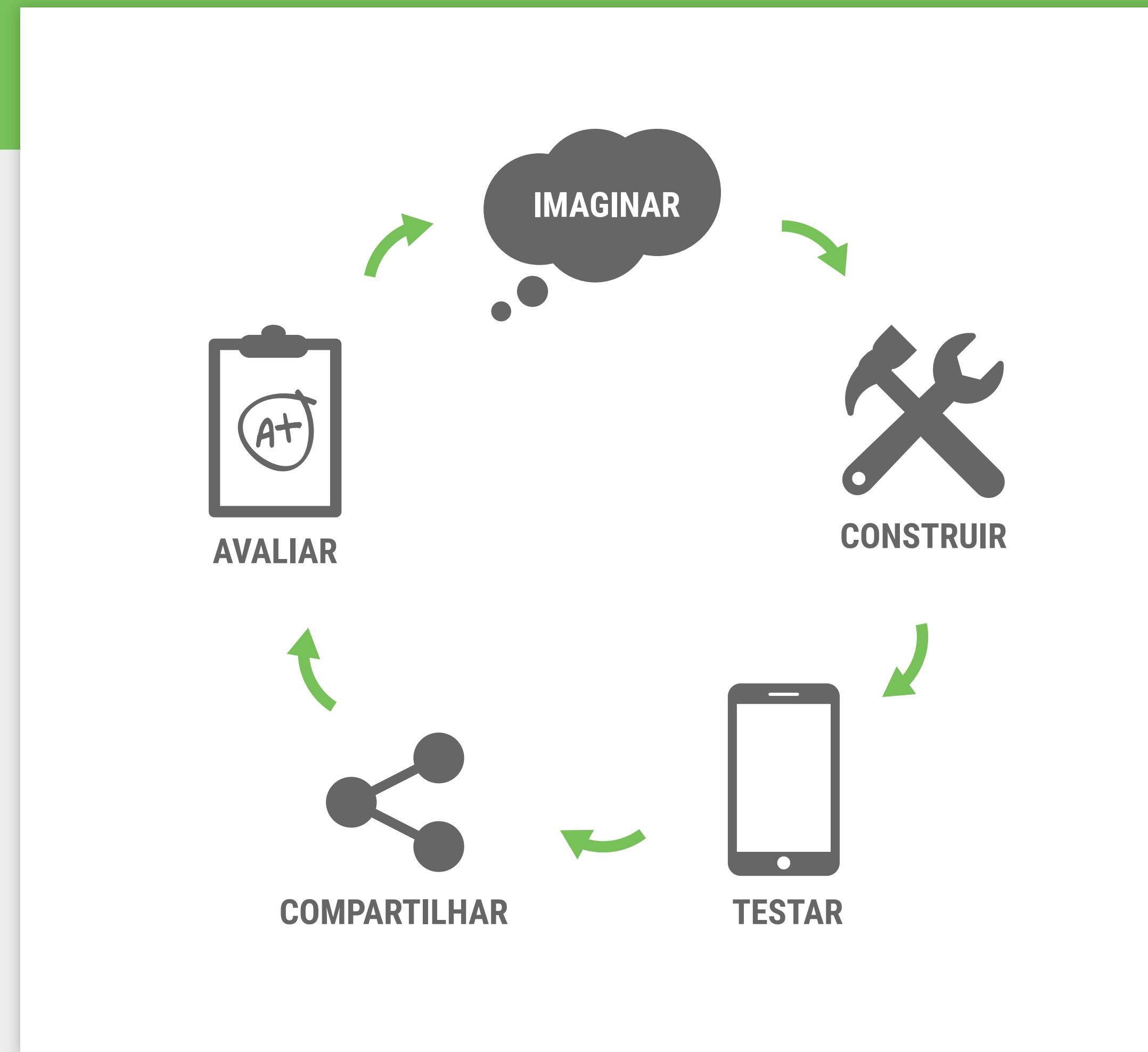


SOFTWARE

Um progra-ma de software é um algoritmo, o qual foi traduzido para uma linguagem que o computador entende.



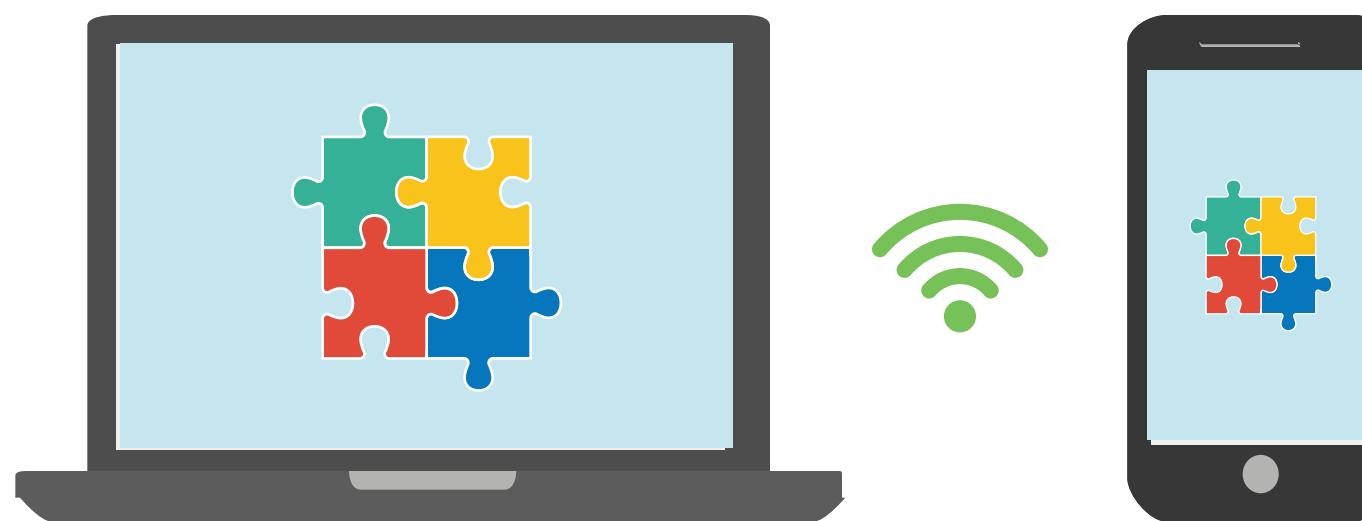
COMO FAZER APPS DE CELULAR



CONFIGURANDO SEU APP INVENTOR

OPÇÃO 1: RECOMENDADA

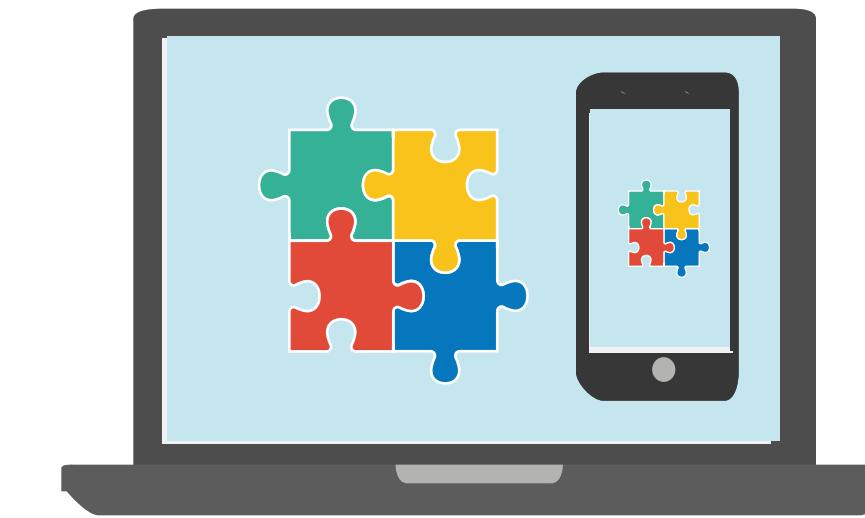
Conecte seu computador e seu dispositivo Android na mesma rede wi-fi



Construa um projeto no seu computador e teste em tempo real no seu dispositivo Android

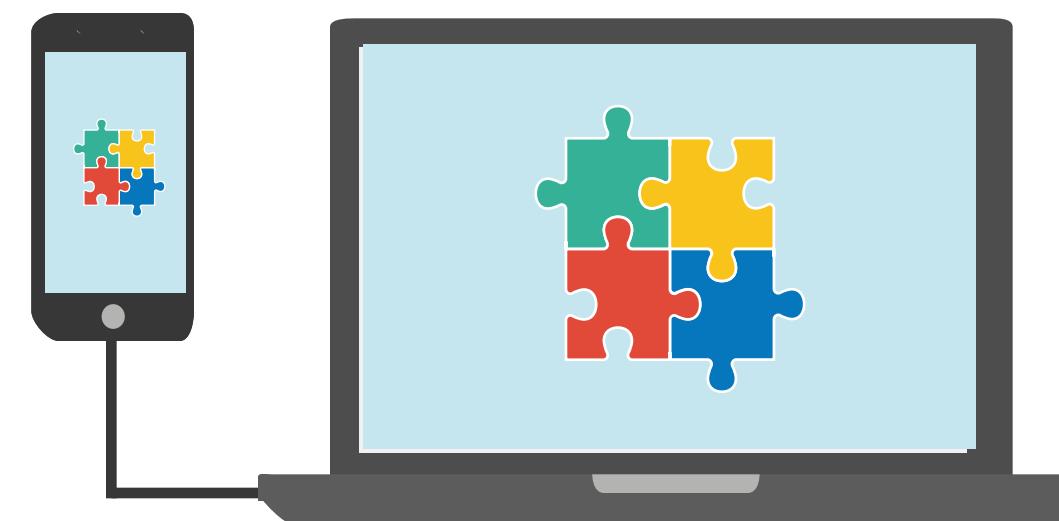
OPÇÃO 2

Não tem um dispositivo Android? Use um emulador



OPÇÃO 3

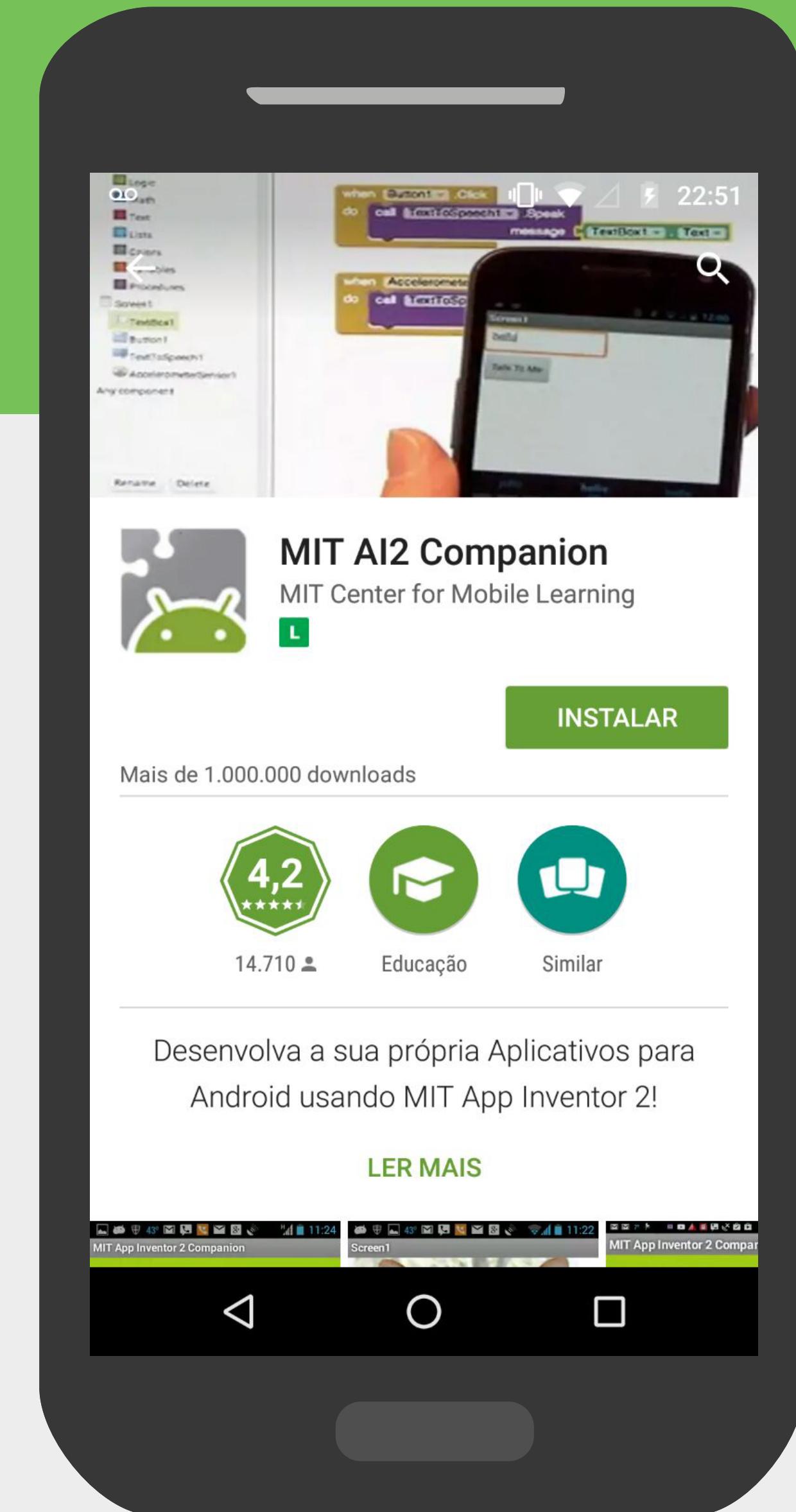
Não tem conexão wi-fi?
Costrua apps com um dispositivo Android e um cabo USB



INSTALANDO O APLICATIVO

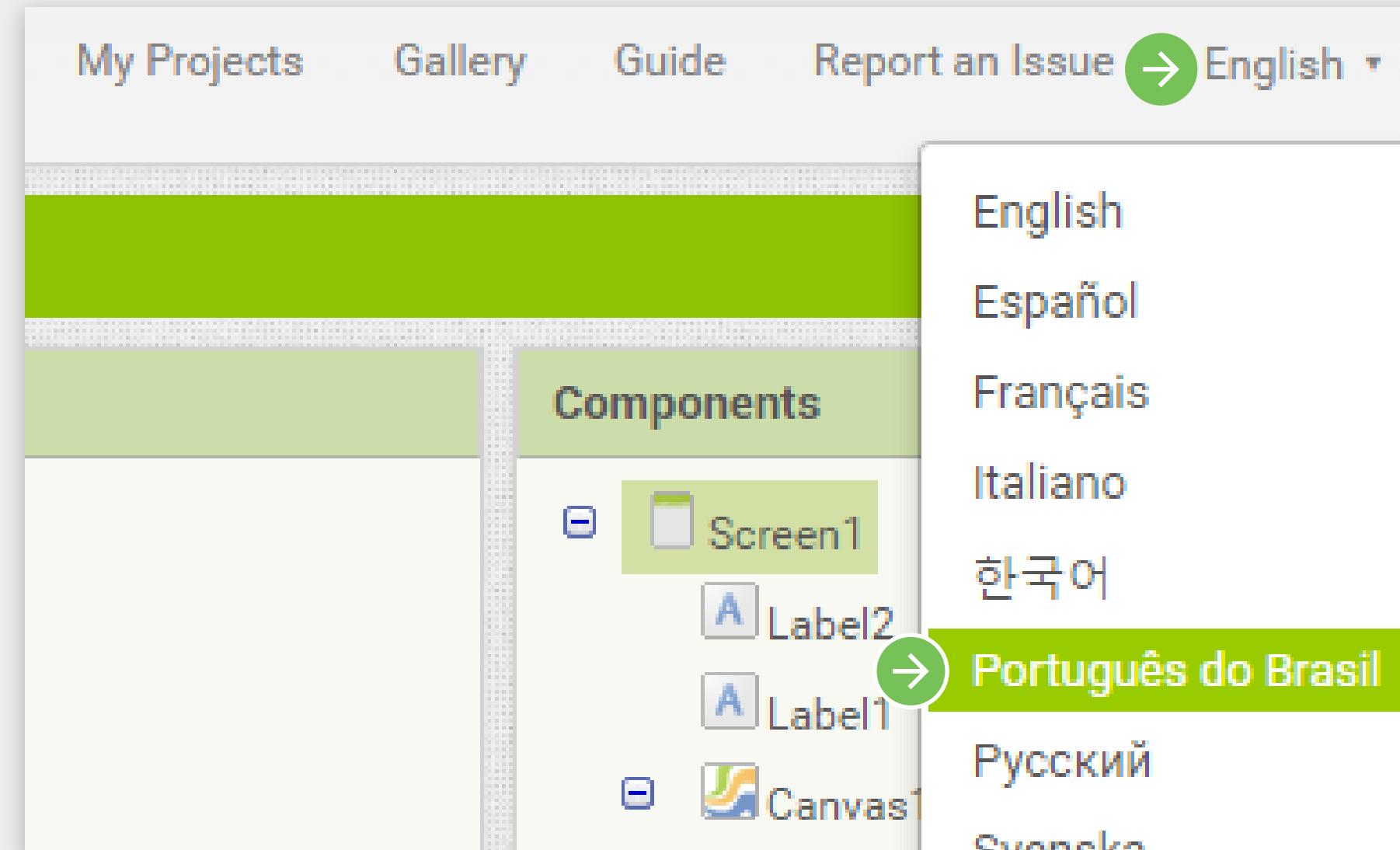
Para poder testar o seu jogo do mosquito em seu celular, primeiramente precisamos instalar o aplicativo do MIT.

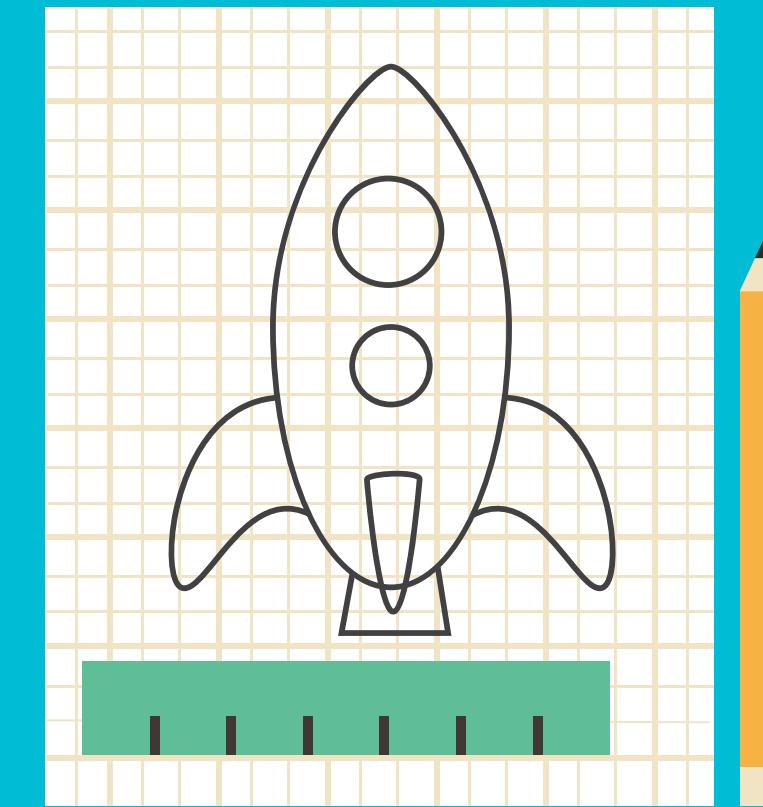
1. Vá na Google Play e procure por 'MIT AI2 Companion'.
2. Escolha o app de mesmo nome
3. Clique em 'Instalar' e aguarde o download e instalação do aplicativo



MUDANDO O IDIOMA

O idioma do AppInventor está em Inglês, e precisamos alterar para Português. No canto superior direito da sua tela do AppInventor, você verá escrito “English”. Clique ali para abrir as opções de idioma e selecione “Português do Brasil”





CRIANDO UM NOVO PROJETO

CRIANDO UM NOVO PROJETO

Vamos iniciar criando um novo projeto no AppInventor.

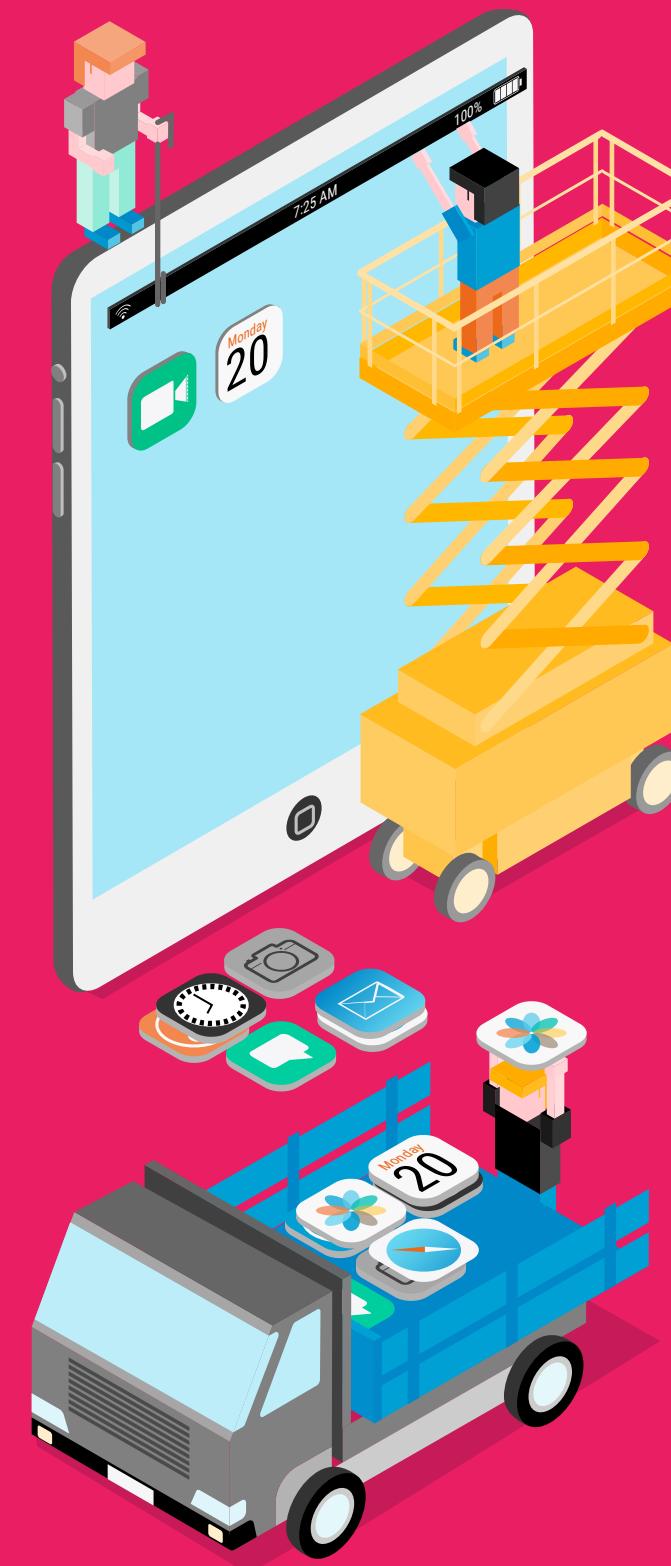
The screenshot shows the MIT App Inventor 2 Beta interface. At the top, there is a logo featuring a green Android head inside a grey puzzle piece, followed by the text "MIT App Inventor 2" and "Beta". To the right of the logo are three dropdown menus: "Projetos", "Conectar", and "Compilar". Below the header is a green navigation bar with three buttons: "Iniciar novo projeto ...", "Apagar Projeto", and "Publicar na Galeria". The main area is titled "Meus Projetos" and contains two columns: "Nome" and "Data de Criação".

CRIANDO UM NOVO PROJETO

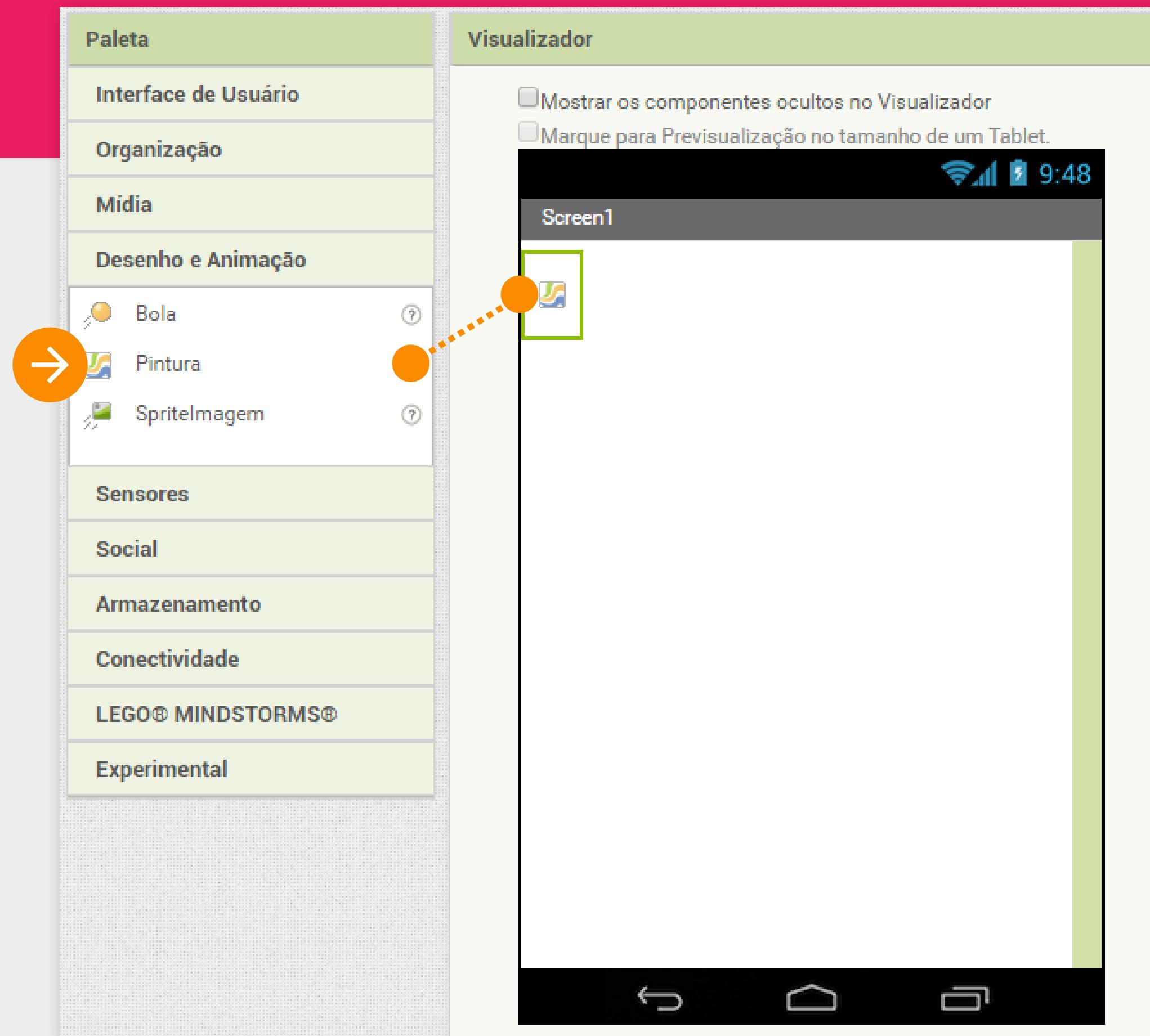
Vamos definir o nome do novo projeto como “**PegaMosquito**” e clicar em OK.



DEFININDO A INTERFACE



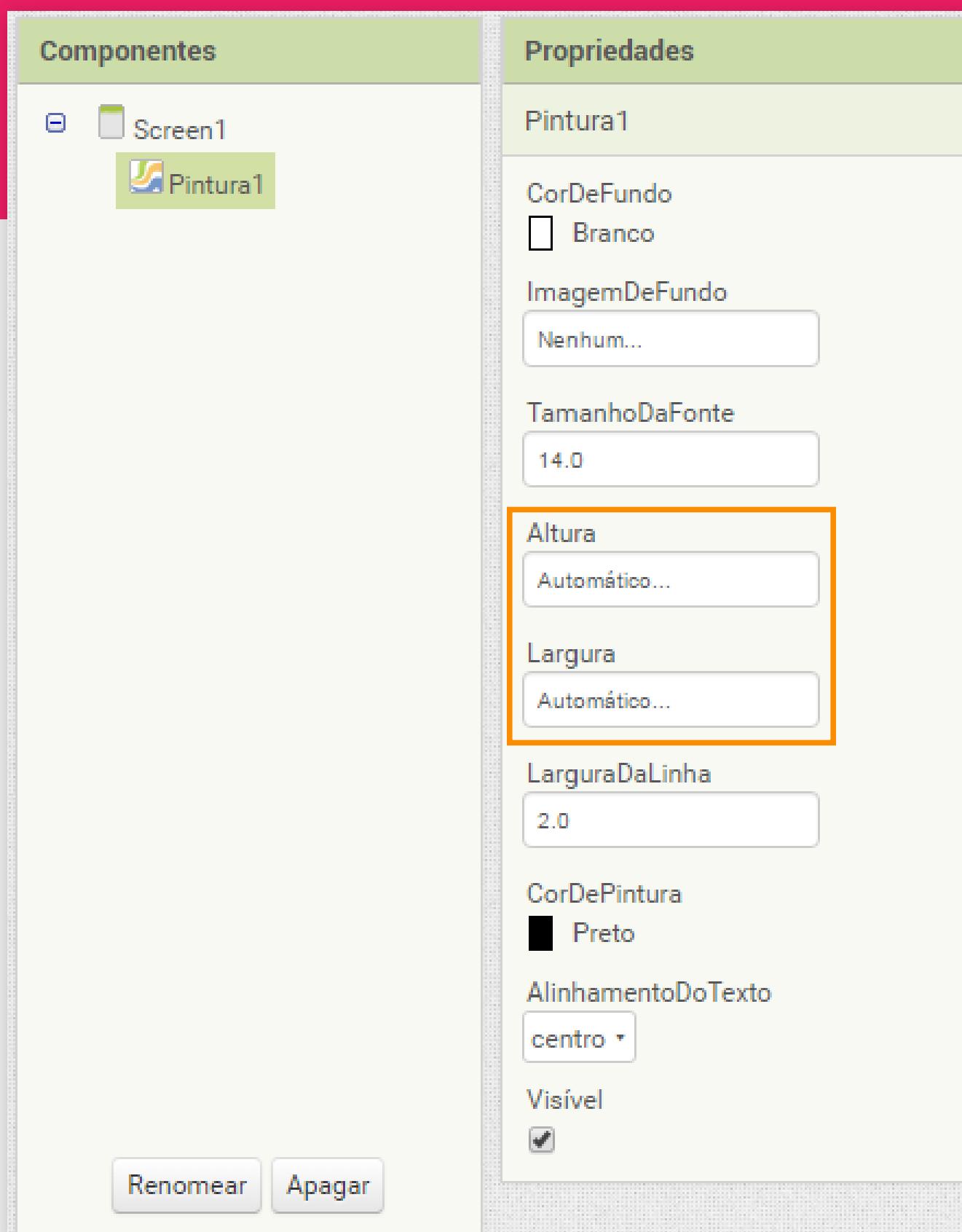
DEFININDO A INTERFACE



Vamos começar definindo a nossa interface.

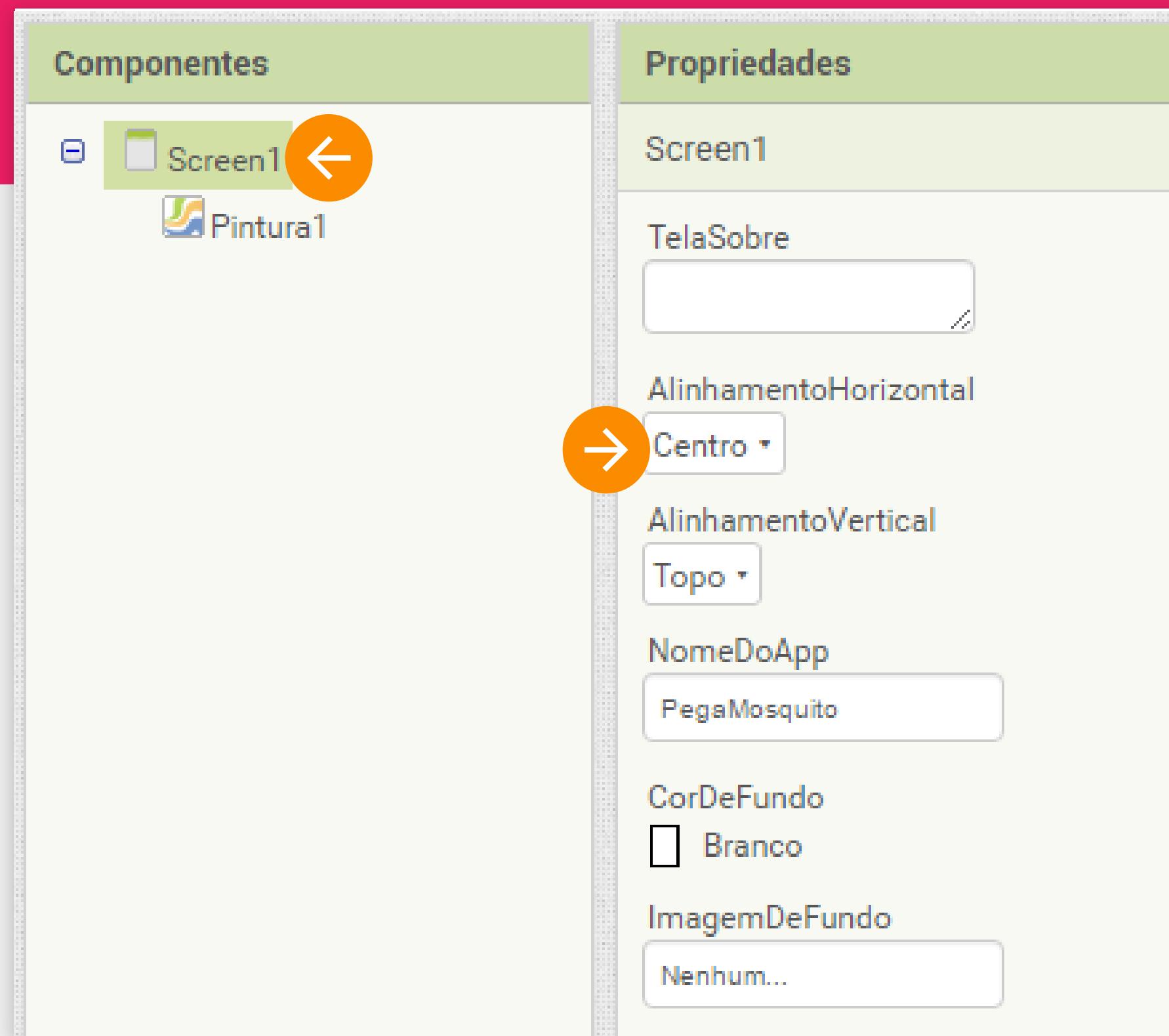
Primeiro vamos clicar na paleta “**Desenho e Animação**” e arrastar o componente “**Pintura**” para dentro da nossa tela (no visualizador).

DEFININDO A INTERFACE



Vamos alterar as propriedades da pintura para que ela possua altura de 300 pontos e largura de 300 pontos.

DEFININDO A INTERFACE



Para alinhar a pintura no centro da tela, vamos selecionar o componente “Screen1” e na propriedade “AlinhamentoHorizontal” vamos selecionar “Centro”.

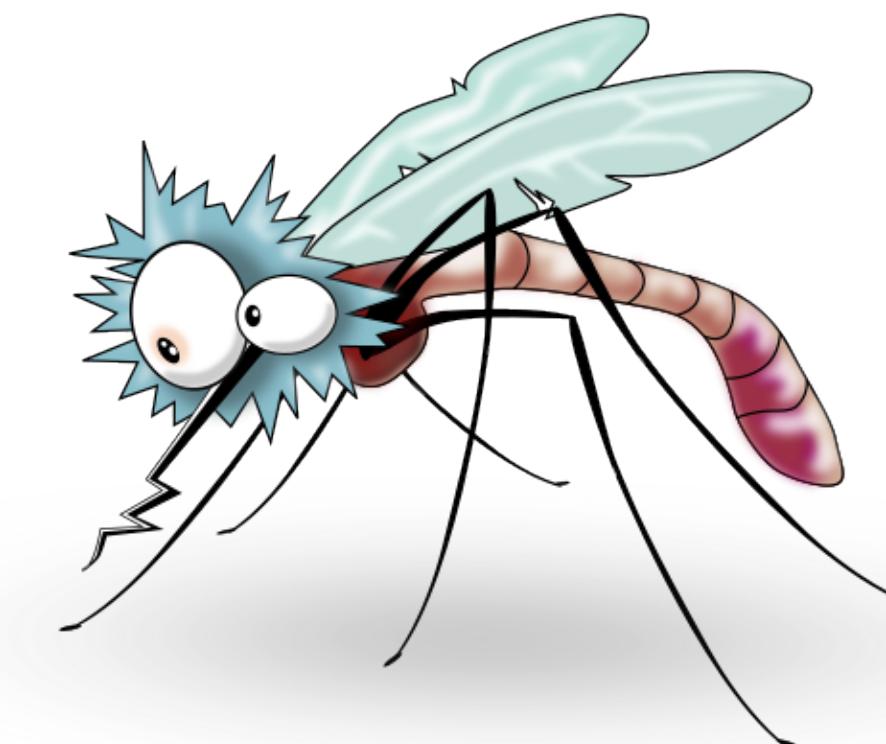


ADICIONANDO O MOSQUITO

ADICIONANDO O MOSQUITO

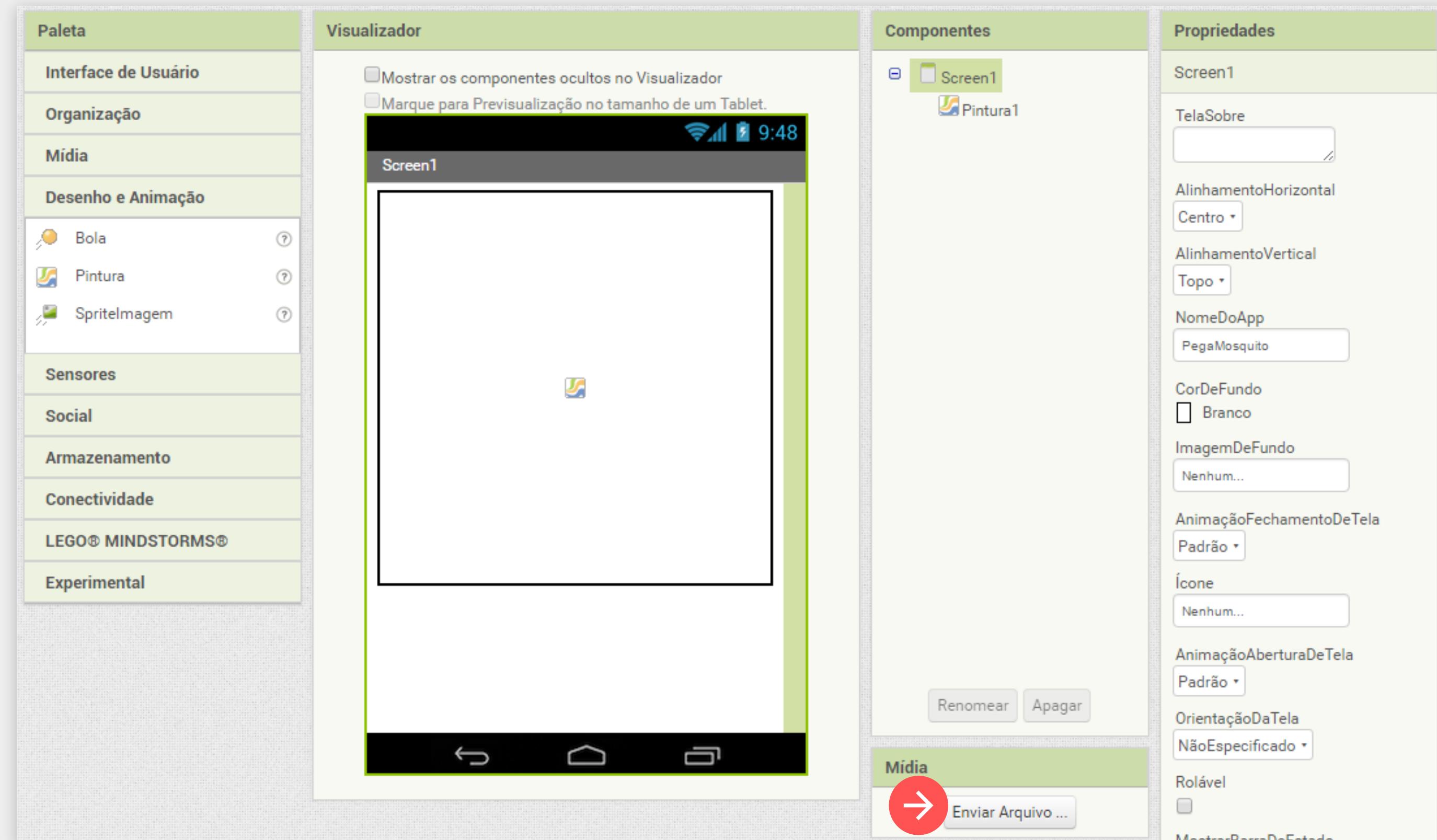
Primeiro baixe a imagem do mosquito aqui:

<http://goo.gl/XRy7jL>



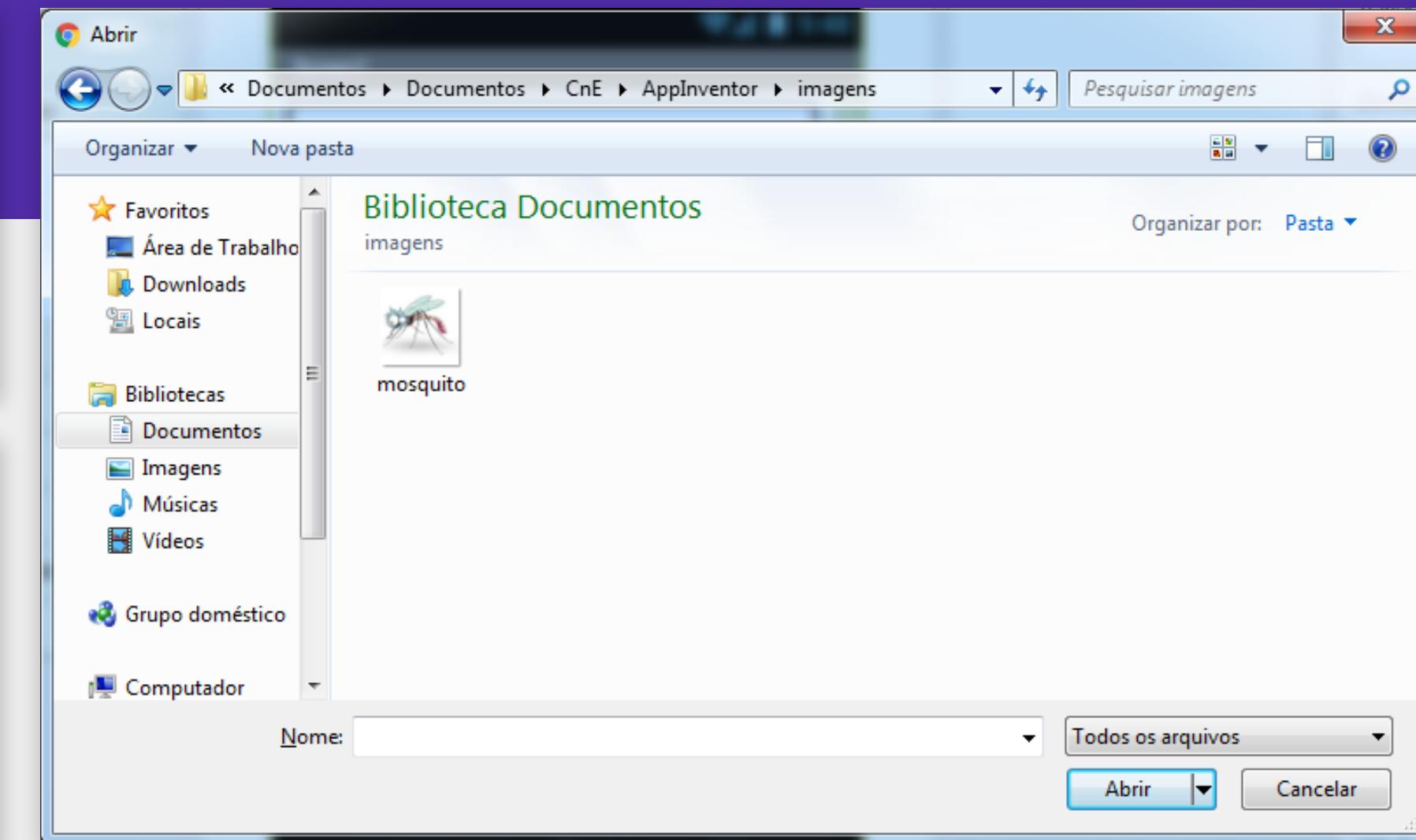
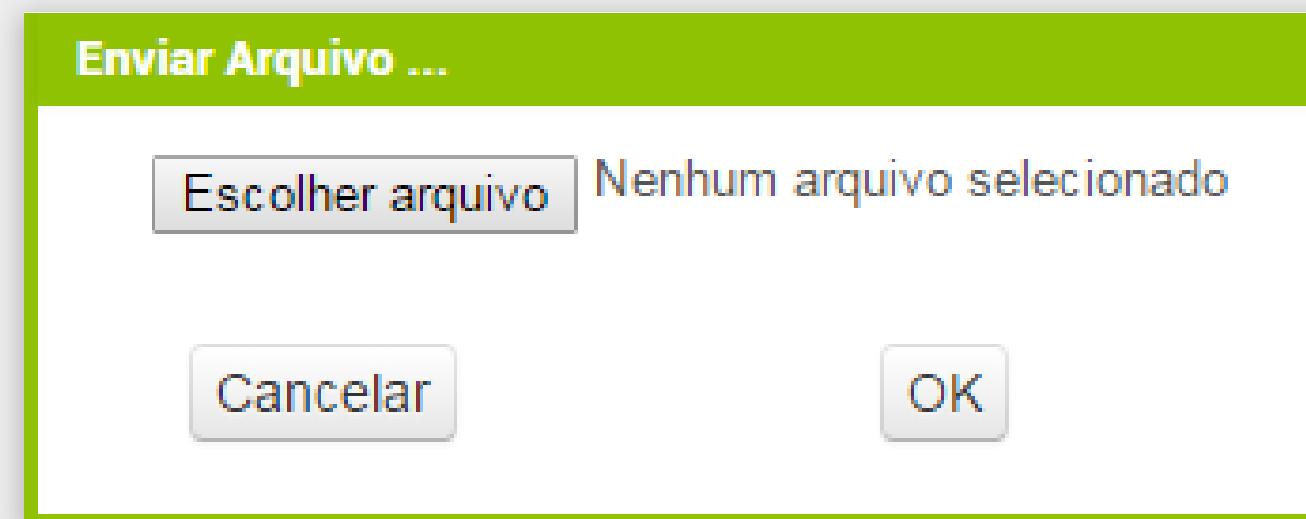
ADICIONANDO O MOSQUITO

Insira a figura do mosquito clicando no botão “Enviar arquivo...” em Mídia.



ADICIONANDO O MOSQUITO

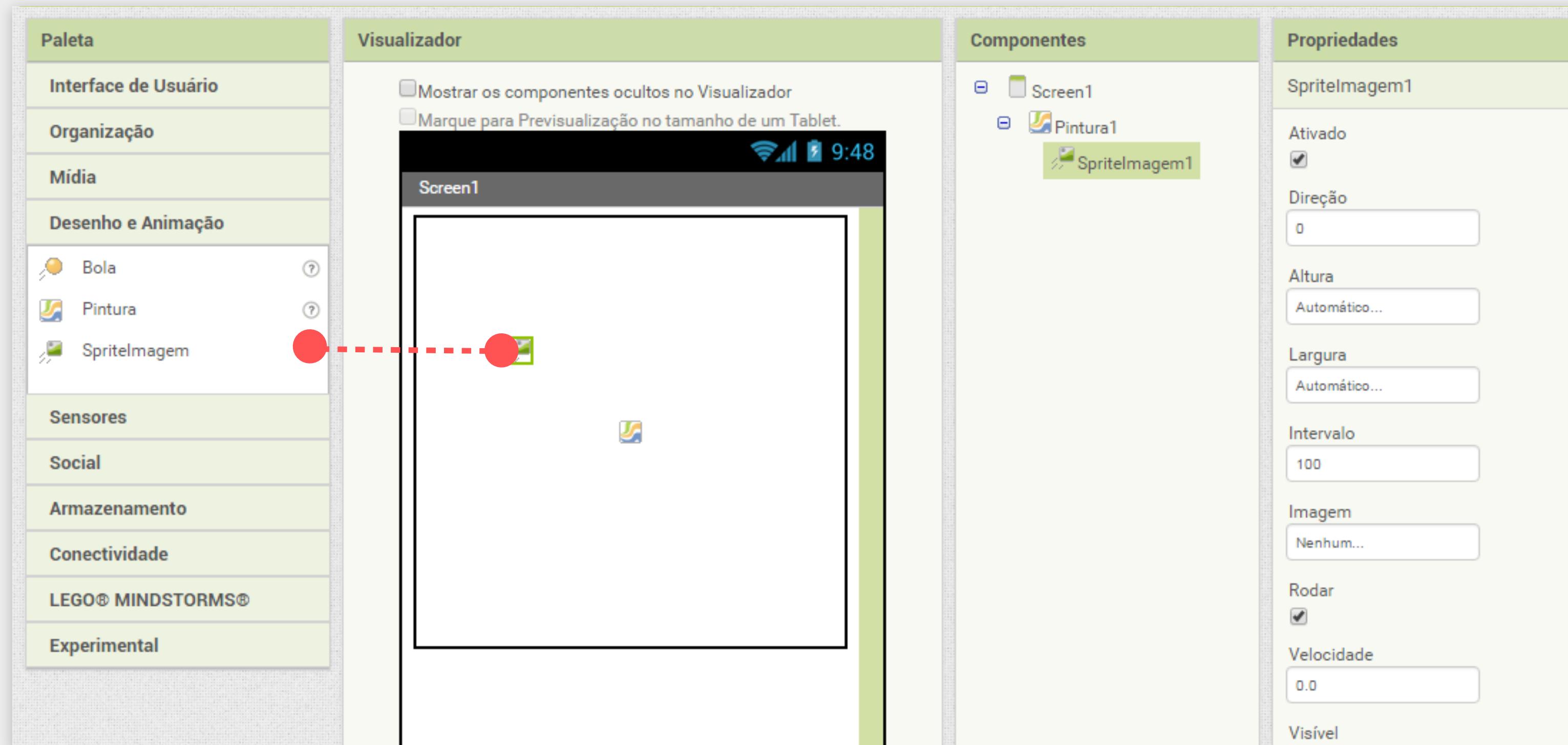
Clique no botão “**Escolher arquivo**” e selecione o arquivo com nome “**mosquito.png**” que você baixou.



ADICIONANDO O MOSQUITO

Para colocarmos o mosquito na nossa tela, precisamos usar um tipo de imagem chamada de “**Sprite**”.

Selecione o componente “**SpritelImage**” na Paleta de “**Desenho e Animação**” e arraste para a “**Pintura**”.

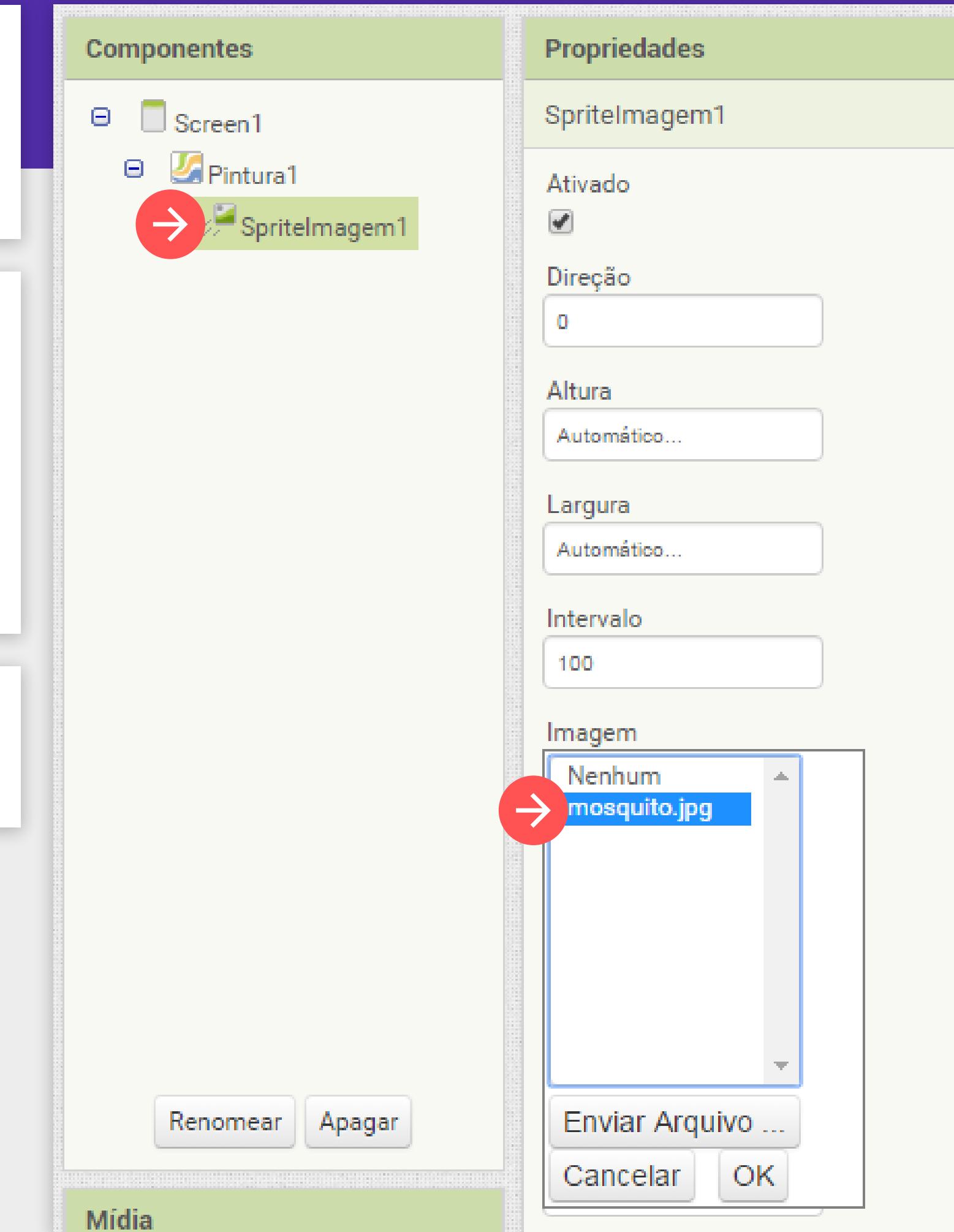


ADICIONANDO O MOSQUITO

Agora devemos definir a imagem dessa Sprite.

Para fazer isso, nas Propriedades do componente “**Spritelimage1**”, clique no campo “**Imagen**” e selecione o arquivo “**mosquito.png**”.

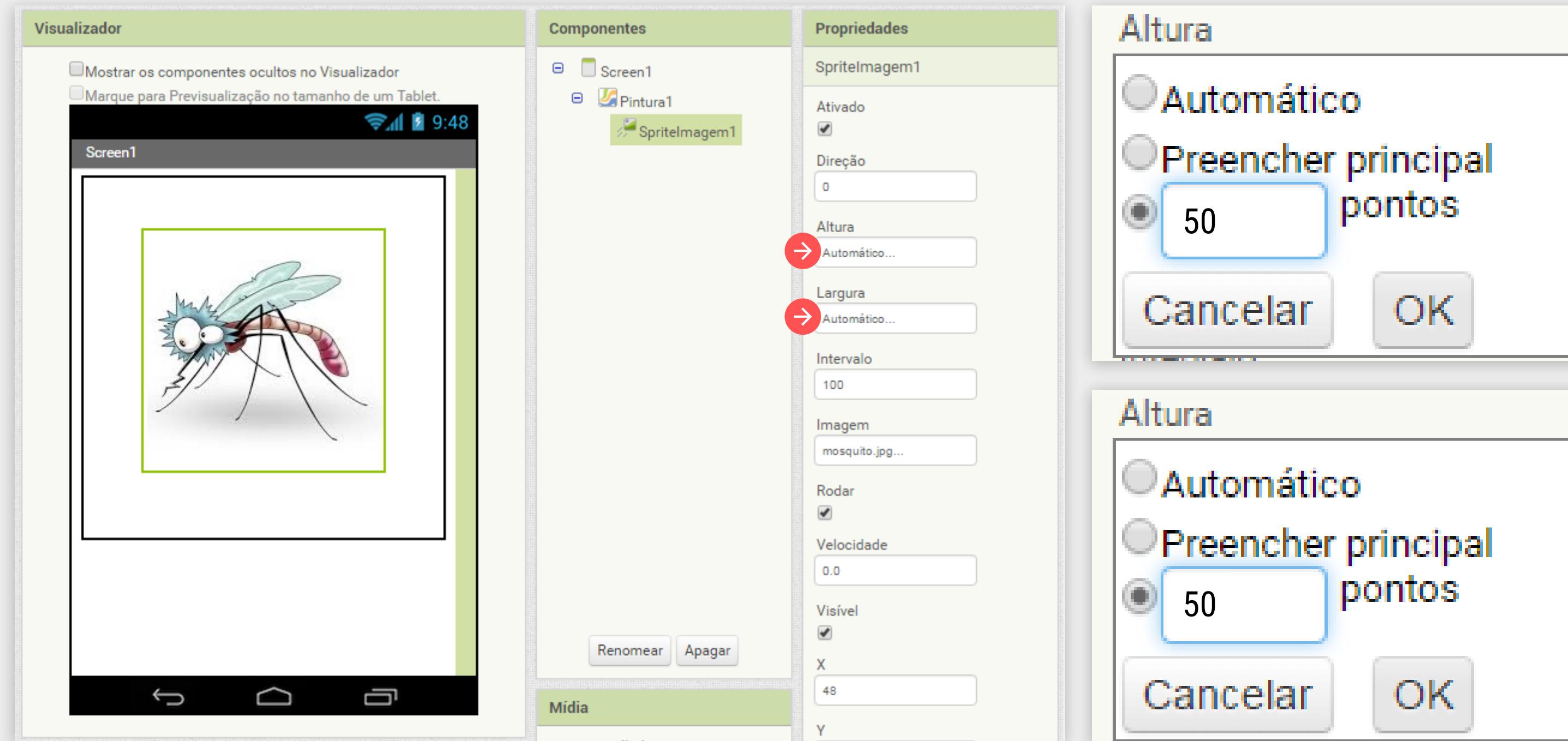
Clique em OK



ADICIONANDO O MOSQUITO

Agora vamos redimensionar a imagem para que ela fique do tamanho desejado.

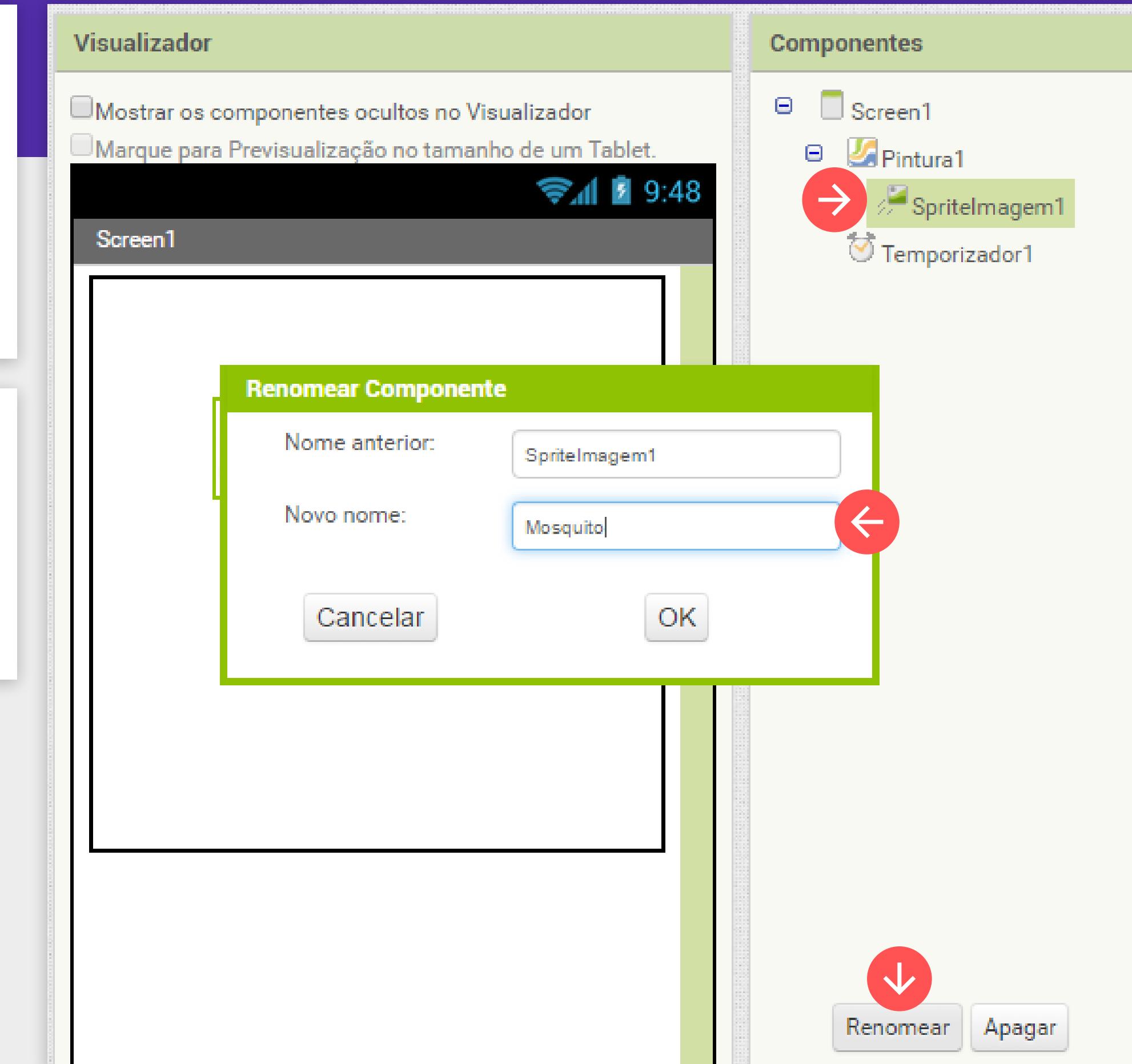
Mude a altura e largura do mosquito para 50.

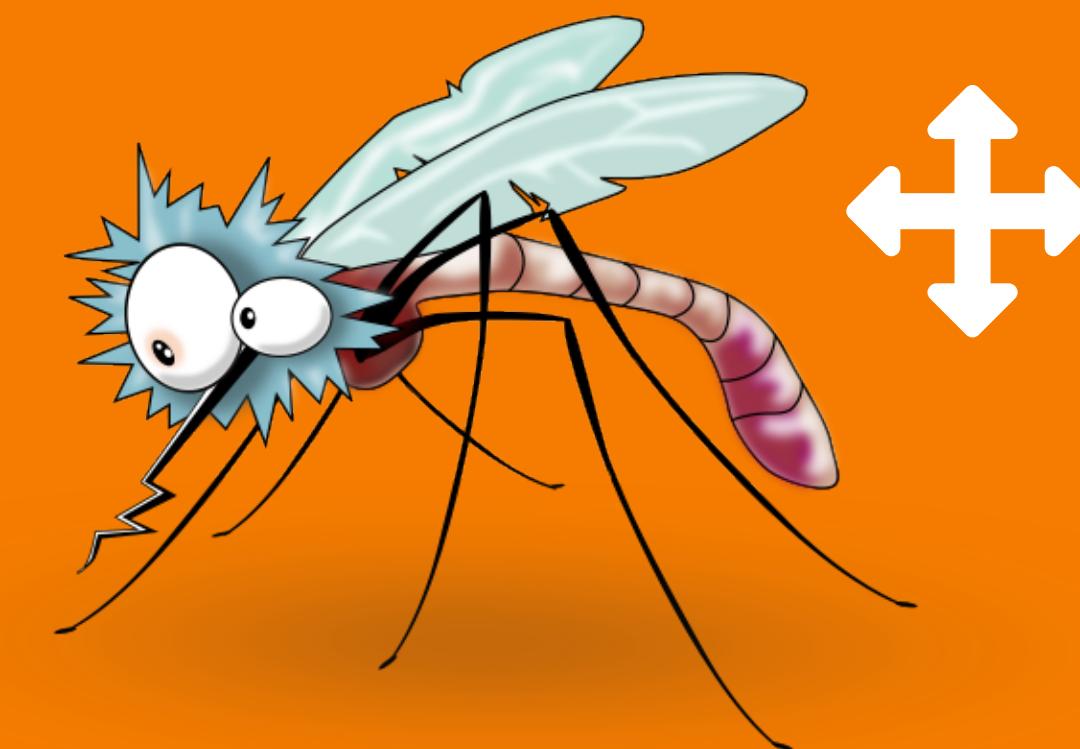


ADICIONANDO O MOSQUITO

Para ficar mais fácil reconhecer que o componente SpritelImage1 é a imagem do nosso mosquito, vamos mudar o nome dele para “**Mosquito**”.

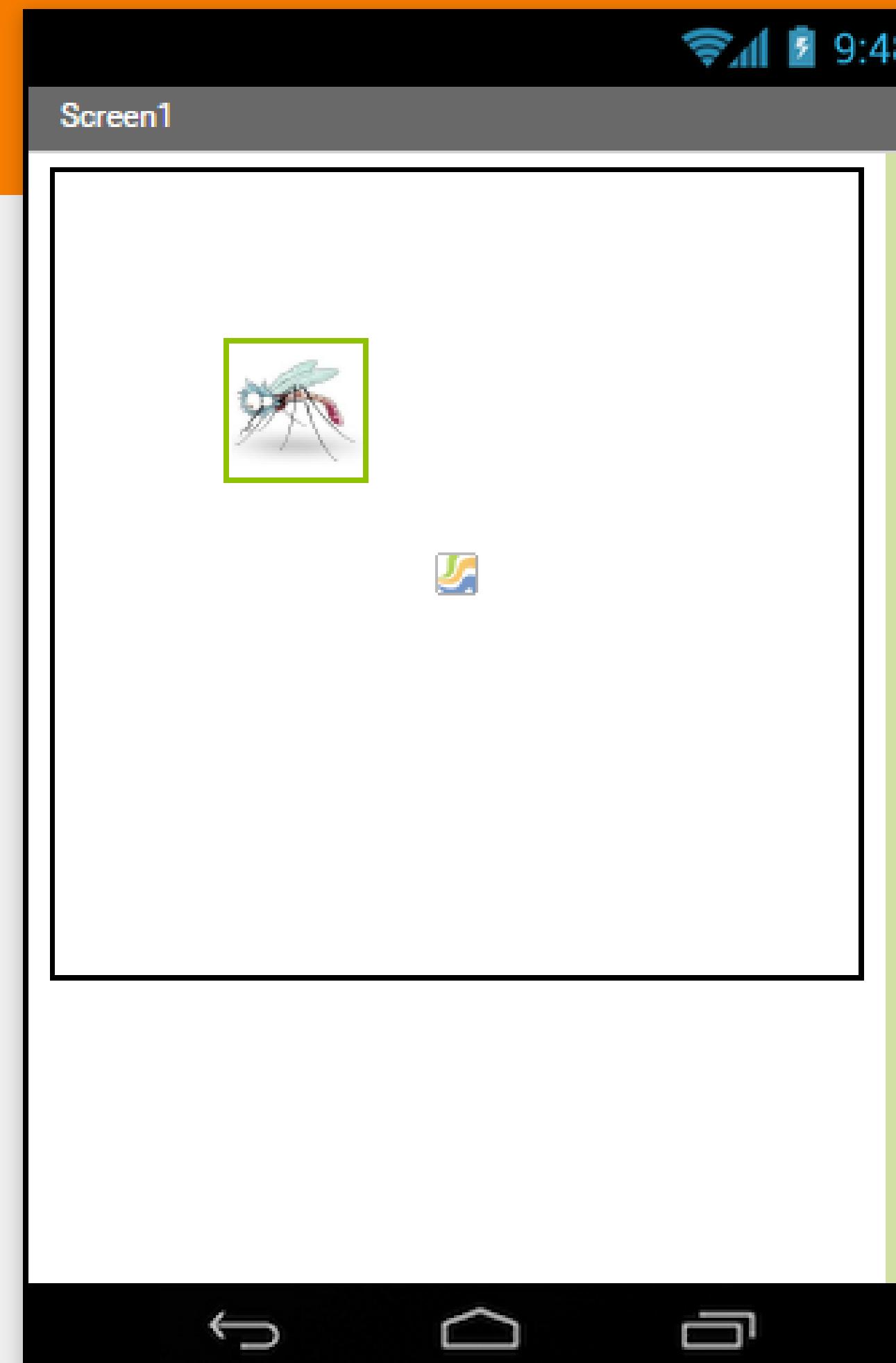
Selecione o componente e clique em Renomear. Digite “**Mosquito**” e clique em OK.





MOVER O MOSQUITO PELA TELA

MOVER O MOSQUITO PELA TELA



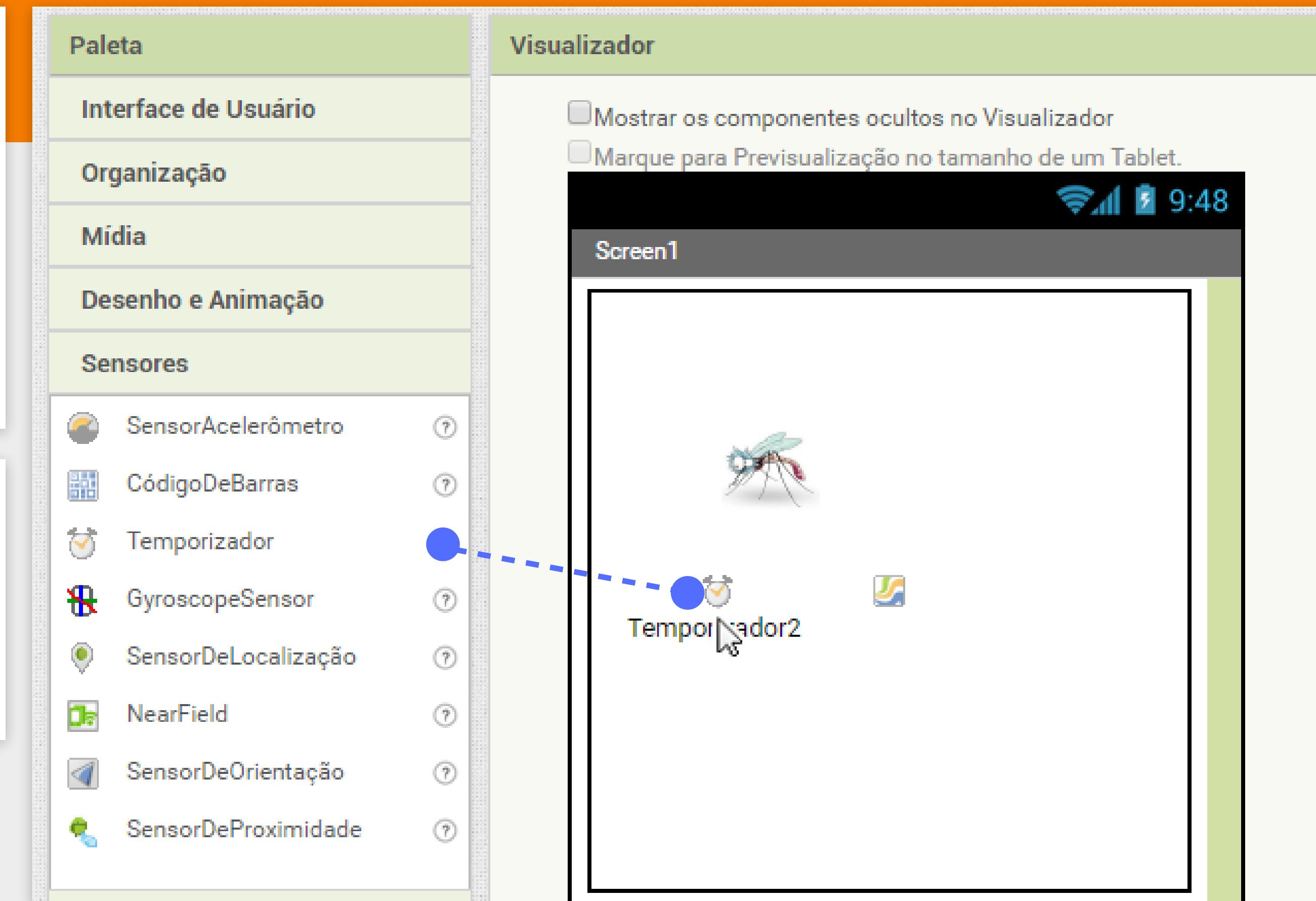
Por enquanto só temos um mosquito parado na tela.

AGORA VAMOS FAZÊ-LO SE MOVIMENTAR.

MOVER O MOSQUITO PELA TELA

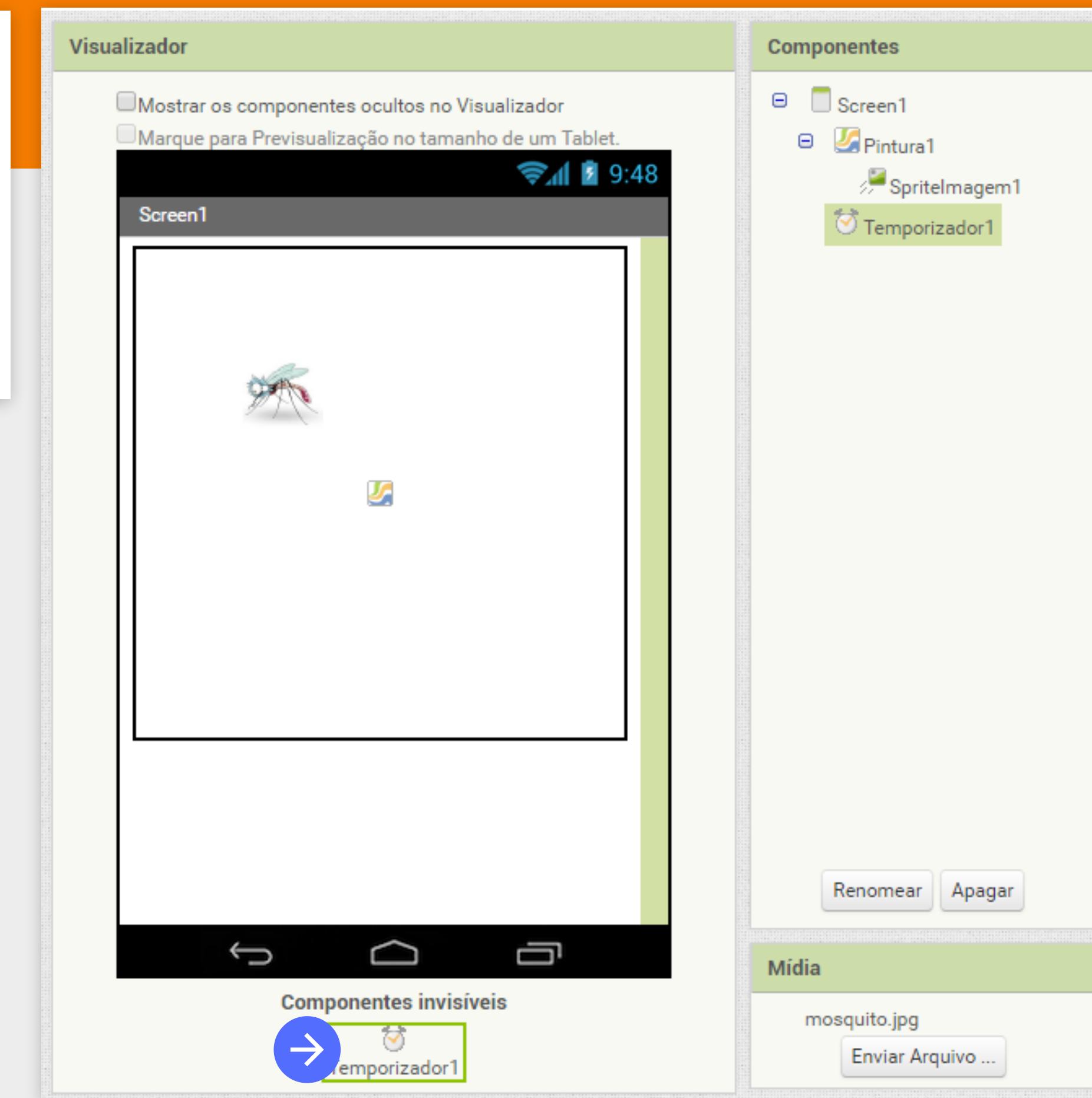
Para o mosquito se movimentar iremos utilizar o componente **Temporizador**. Este será responsável por mandar o mosquito se movimentar a cada meio segundo.

Devemos clicar em **Sensores** e arrastar o componente **Temporizador** até a tela.



MOVER O MOSQUITO PELA TELA

O componente **Temporizador** é um componente invisível na tela, ele fica apresentado na parte inferior do bloco “**Designer**”.

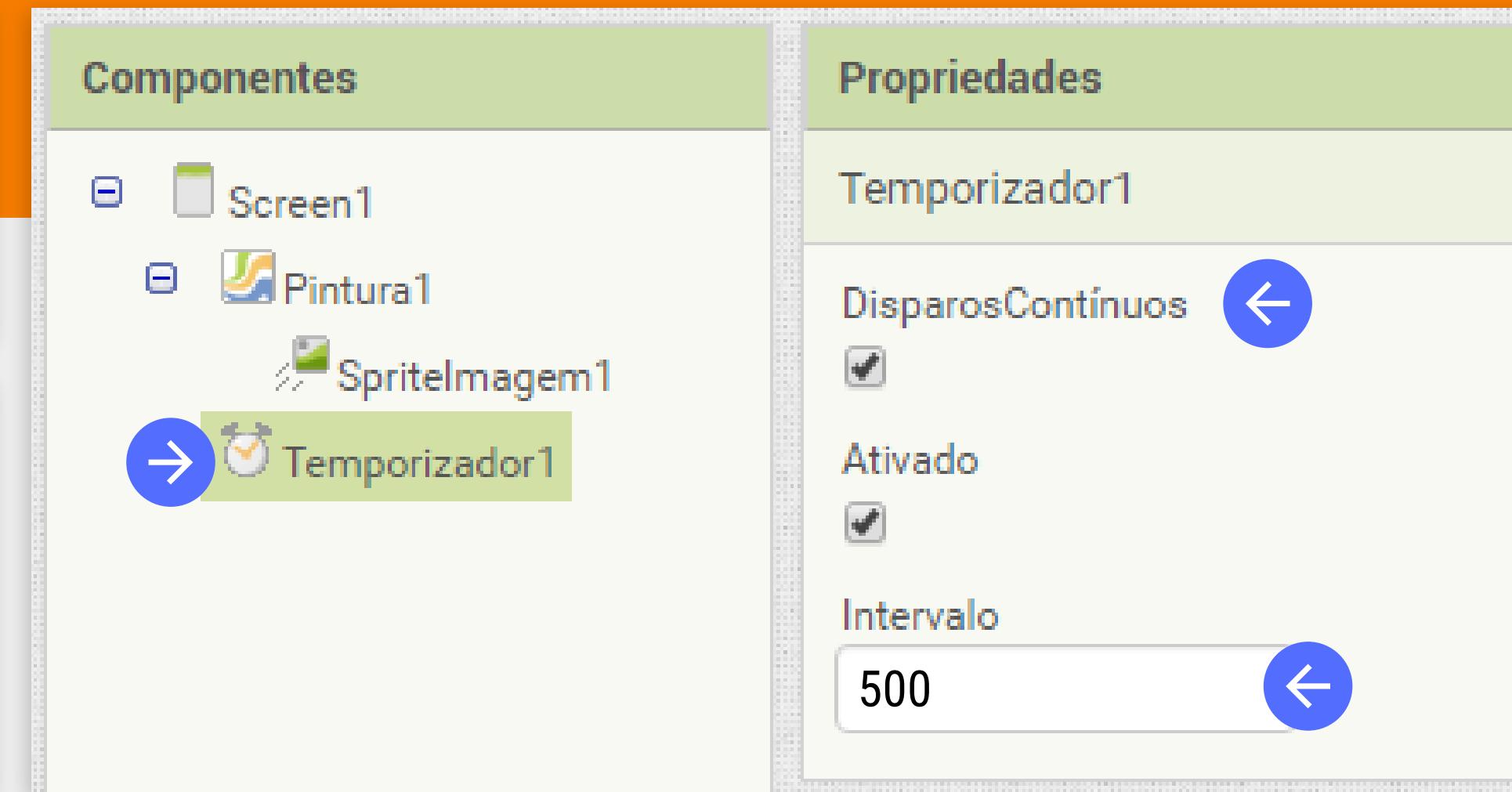


MOVER O MOSQUITO PELA TELA

Selecione o componente **Temporizador1** e defina as suas propriedades.

Com a opção “**DisparosContínuos**” o Temporizador vai ficar disparando repetidamente a cada Intervalo de tempo.

O “**Intervalo**” é o tempo em milissegundos que demora para o comando disparar, como queremos 0,5 segundo, deixamos como 500 milissegundos.



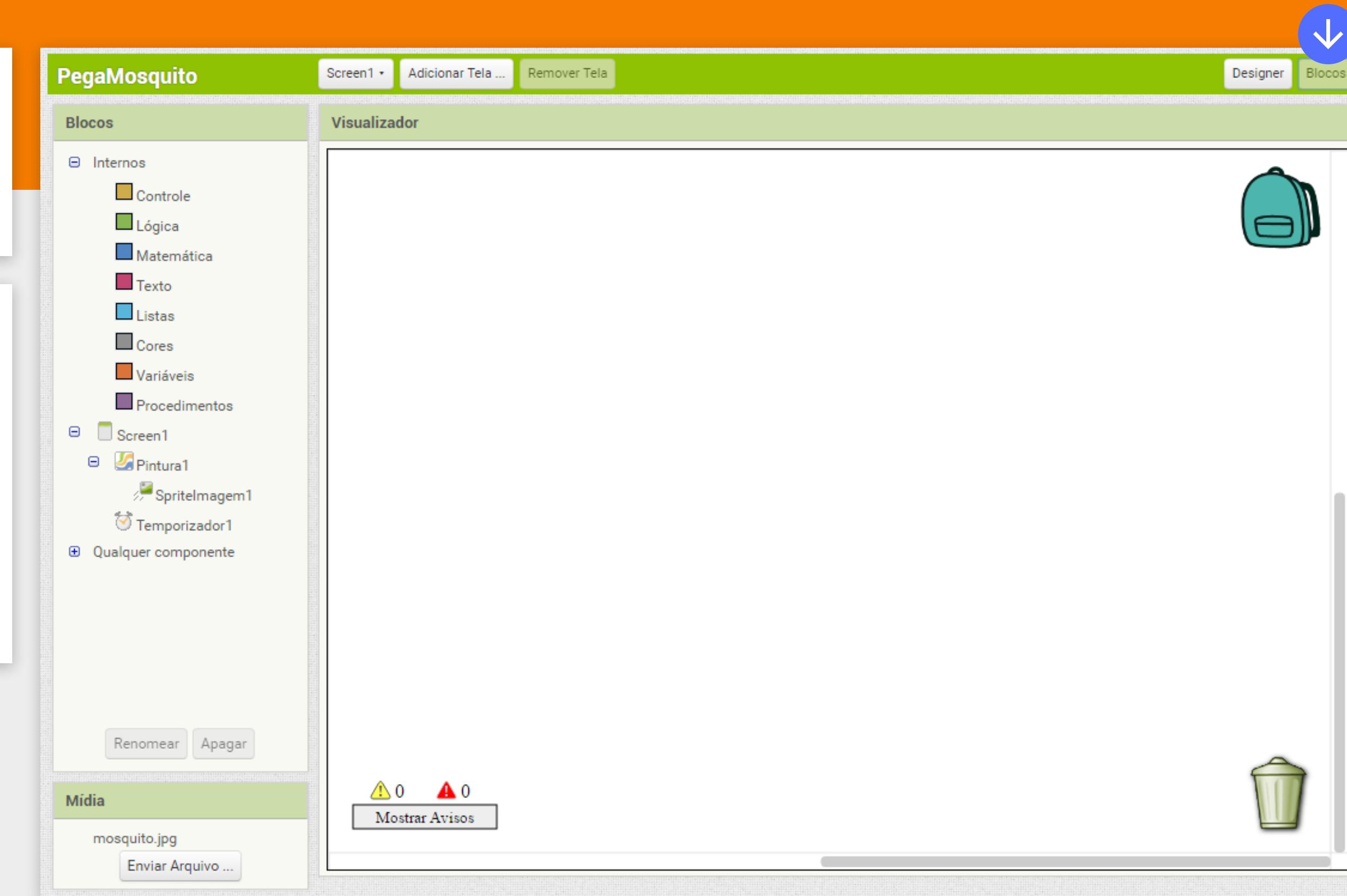
DICA

Sempre que falarmos em selecionar o componente para ver suas propriedades, significa clicar no nome do componente na coluna de “Componentes”

MOVER O MOSQUITO PELA TELA

Agora vamos programar o movimento do mosquito.

Até então estávamos trabalhando na tela de Design do nosso aplicativo, agora vamos passar para a tela de programação, ou **Blocos**.



MOVER O MOSQUITO PELA TELA

Para mover o mosquito na tela, vamos criar um **Procedimento**, ou seja, um conjunto de comandos que vamos chamar de **MoverMosquito**.

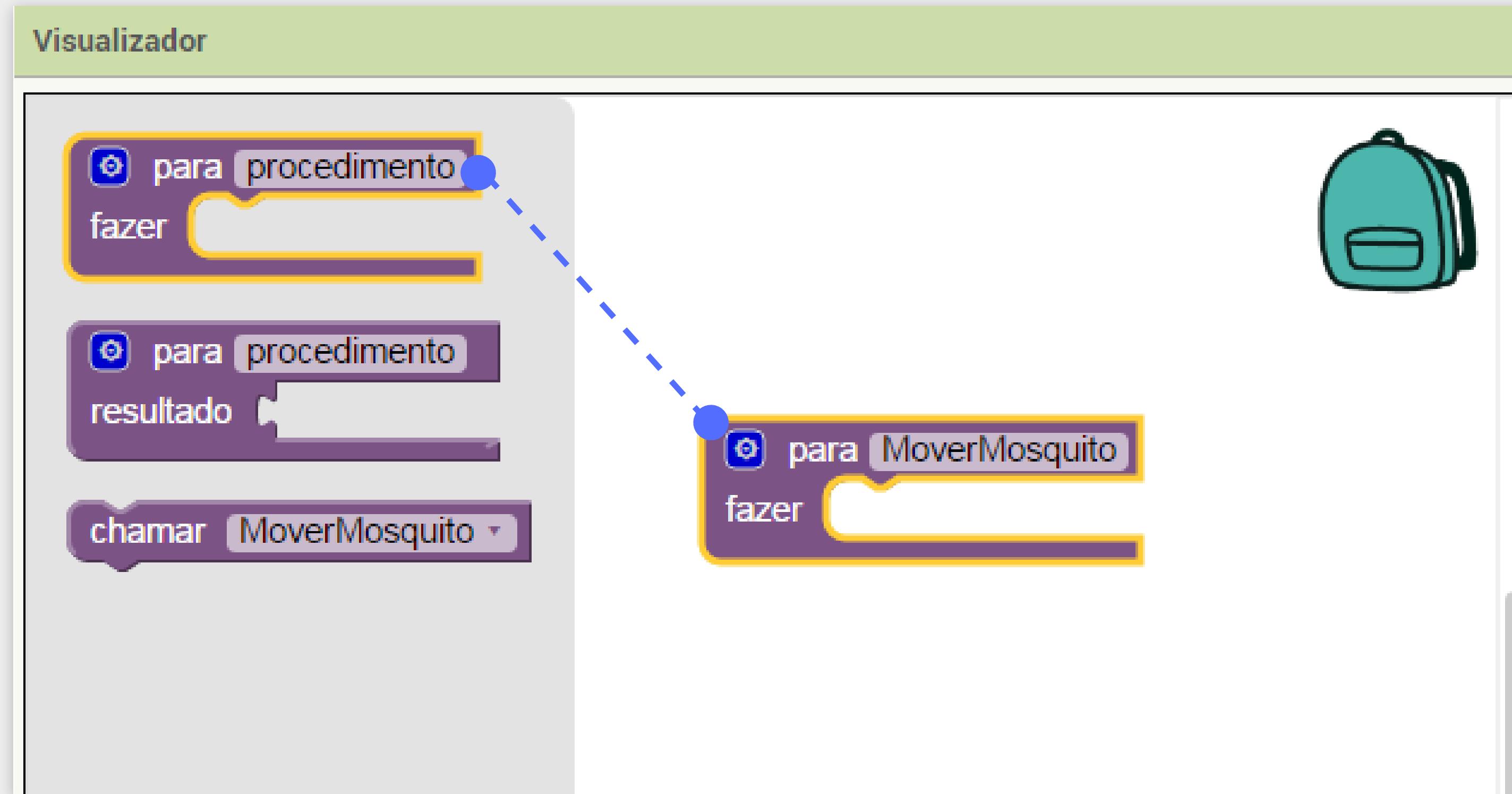
Clique em **Procedimentos** na coluna de Blocos.

The image shows the Scratch interface. On the left is the 'Blocos' (Blocks) palette, which is divided into sections: 'Internos' (Internal blocks), 'Screen1', 'Pintura1', 'Qualquer componente' (Any component), and 'Procedimentos' (Procedures). The 'Procedimentos' section is highlighted with a green background and has a blue circular arrow icon to its right. On the right is the 'Visualizador' (Stage) window, which displays two script blocks: a 'para [procedimento]' (repeat) loop and a 'resultado' (result) block attached to it.

MOVER O MOSQUITO PELA TELA

Arraste o comando “**para procedimento fazer**” até a área do Visualizador.

Clique na palavra “**procedimento**” e substitua por “**MoverMosquito**”.



MOVER O MOSQUITO PELA TELA

Esse procedimento deve trocar a posição do Mosquito toda vez que for ativado.

Para fazer isto, vamos utilizar a função de trocar a posição **X** e **Y** do componente **SpritelImage**, ou seja, do “Mosquito”.

Clique no **Mosquito** na coluna de Blocos.

The screenshot shows a Scratch script titled "PegaMosquito" with the stage background set to "Screen1". The script contains five procedures:

- "quando Mosquito .ColidiuCom outro": This procedure has a yellow "fazer" slot.
- "quando Mosquito .Arrastado xInicial yInicial xAnterior yAnterior xAtual yAtual": This procedure has a yellow "fazer" slot.
- "quando Mosquito .AlcançouBorda borda": This procedure has a yellow "fazer" slot.
- "quando Mosquito .Arremessado x y velocidade direção xvel yvel": This procedure has a yellow "fazer" slot.
- "quando Mosquito .ToqueParaBaixo x y": This procedure has a yellow "fazer" slot.

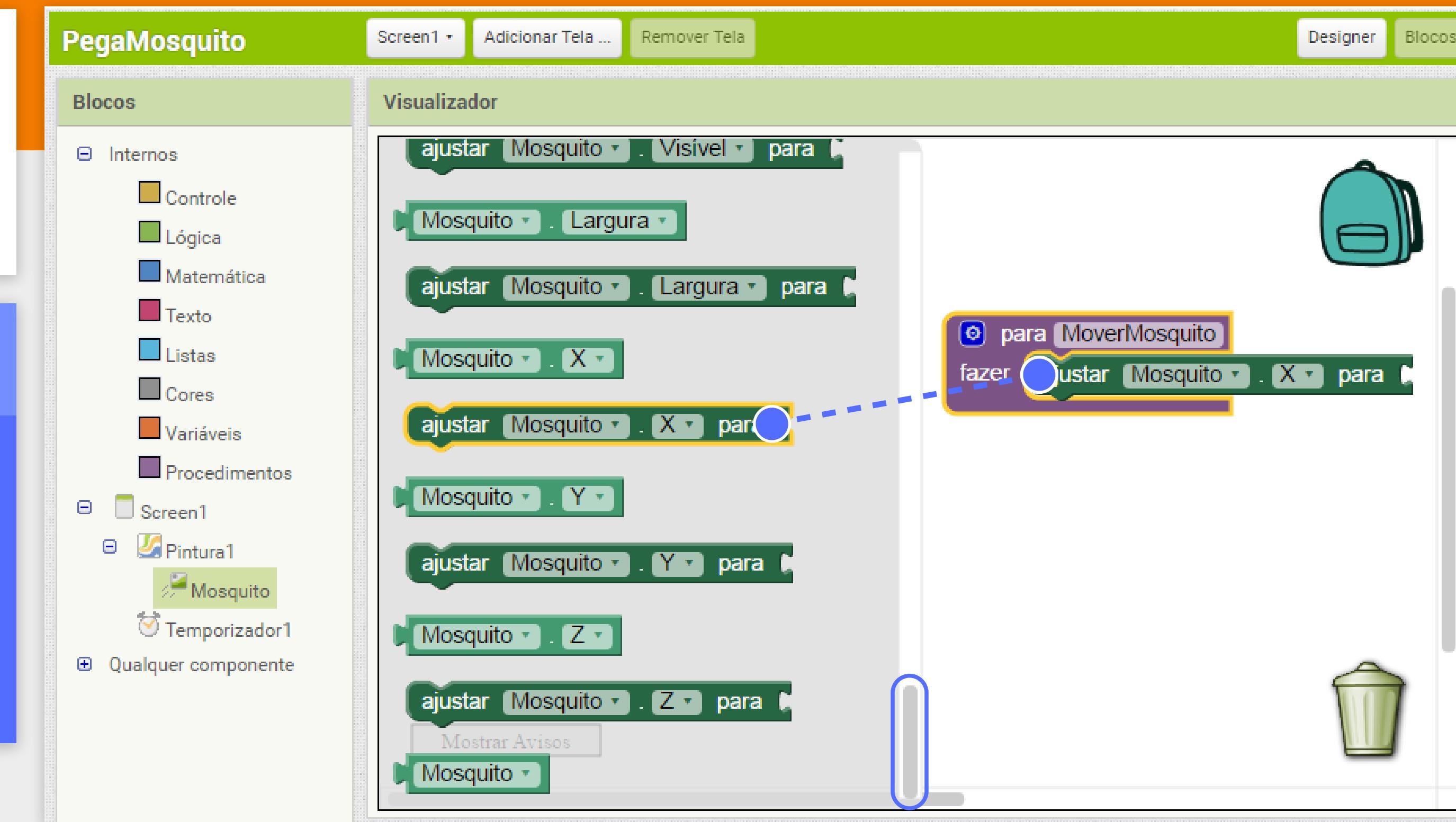
The "Blocos" palette on the left lists categories like Internos, Lógica, Matemática, Texto, Listas, Cores, Variáveis, and Procedimentos. The "Mosquito" sprite is selected in the "Screen1" backdrop. The "Mídia" palette at the bottom shows an image file named "mosquito.jpg".

MOVER O MOSQUITO PELA TELA

Arraste o comando “ajustar Mosquito.X para” para dentro do procedimento “MoverMosquito”.

DICA

Como a lista de comandos do Mosquito é muito grande, você terá que descer a barra de rolagem até o fim para achar o comando utilizado.

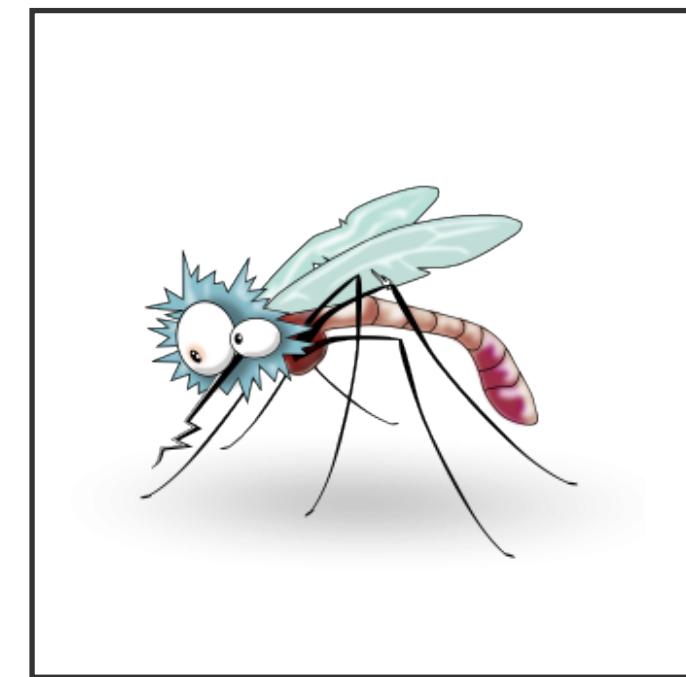


MOVER O MOSQUITO PELA TELA

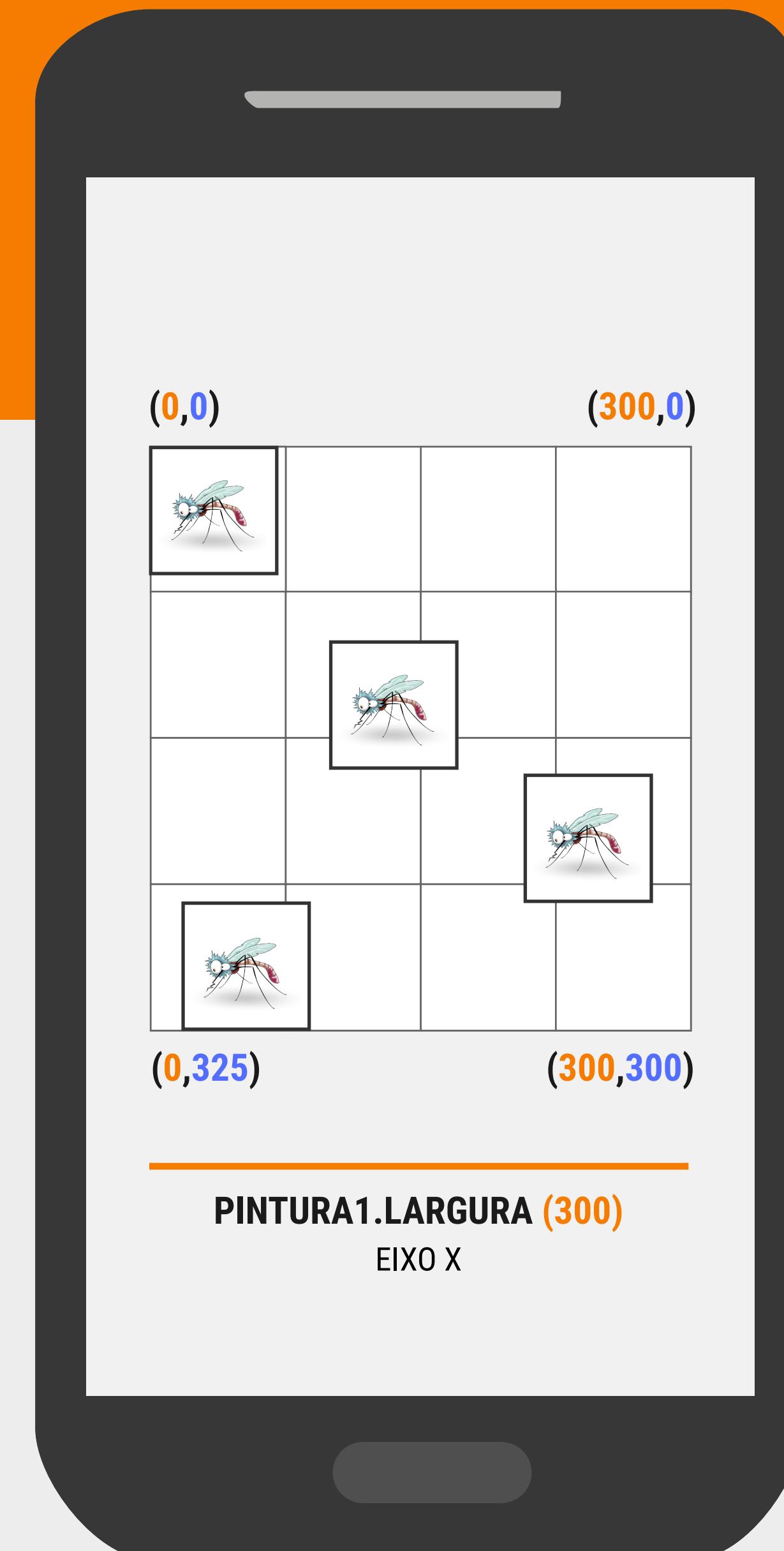
Para que posição queremos mover o mosquito?

A posição deve ficar entre **0** (zero) e **270** (tamanho da tela menos o tamanho do mosquito)

MOSQUITO.ALTURA (30)

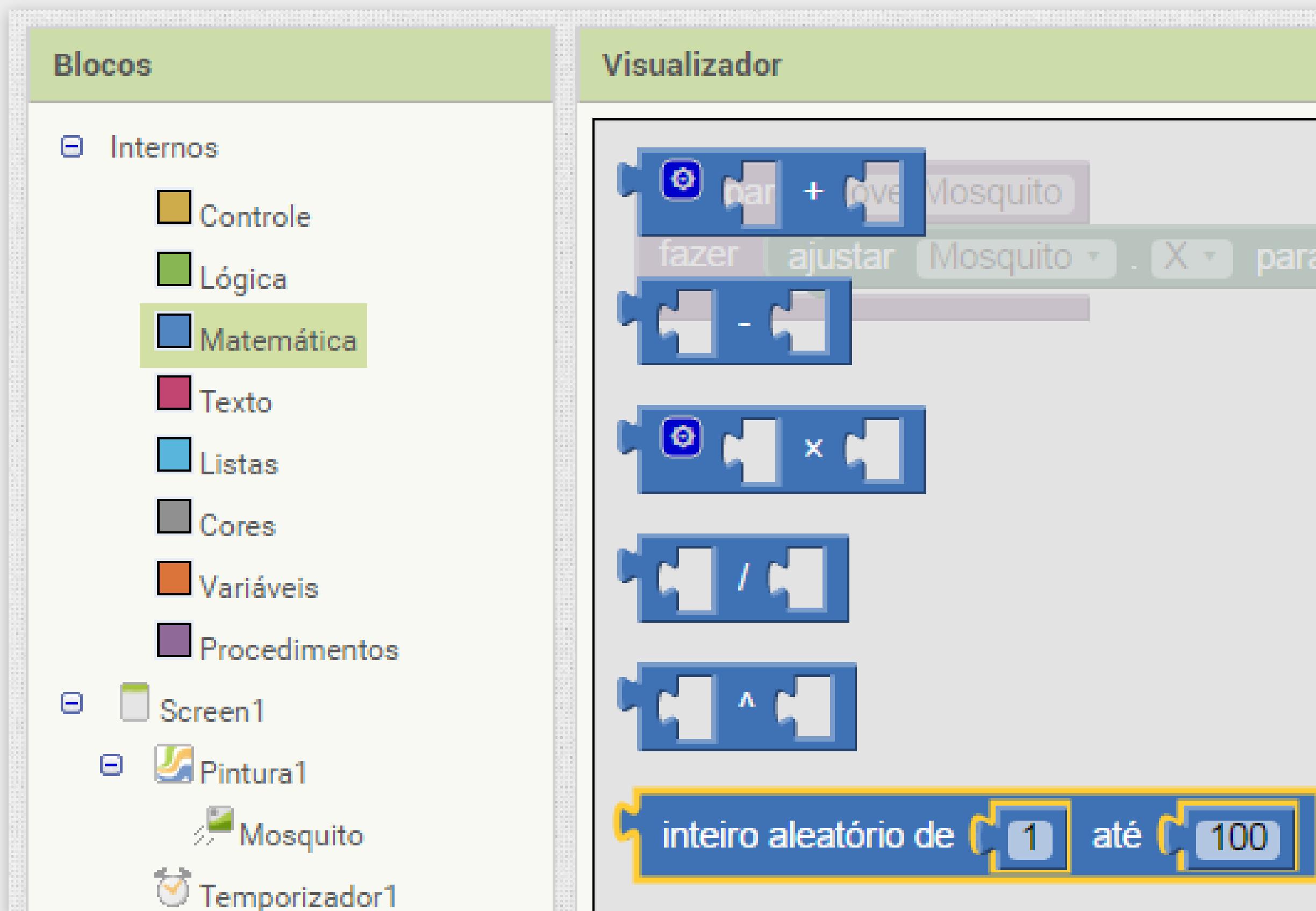


MOSQUITO.LARGURA (30)



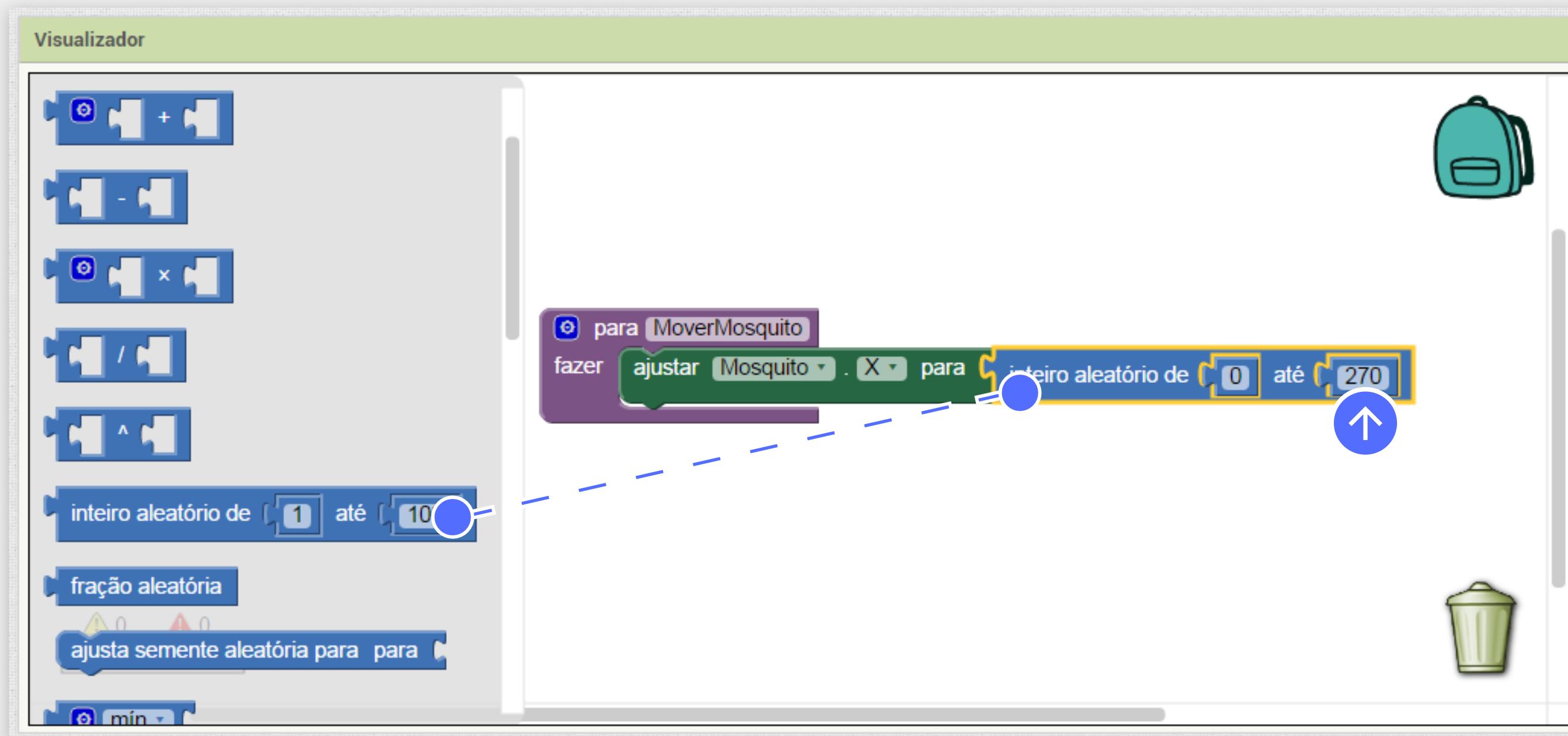
MOVER O MOSQUITO PELA TELA

Como queremos que o Mosquito se move para uma posição “sorteada”, vamos usar o bloco matemático “**inteiro aleatório de 1 até 100**”.



MOVER O MOSQUITO PELA TELA

Arraste o bloco de **inteiro aleatório** para completar o comando de **ajustar Mosquito.X** e preencha com os valores que queremos (0 e 270).



MOVER O MOSQUITO PELA TELA

Já criamos nosso procedimento para mover o mosquito.

Agora precisamos dizer **quando** ele deve se mover, isto é, quando o **Temporizador1** disparar.

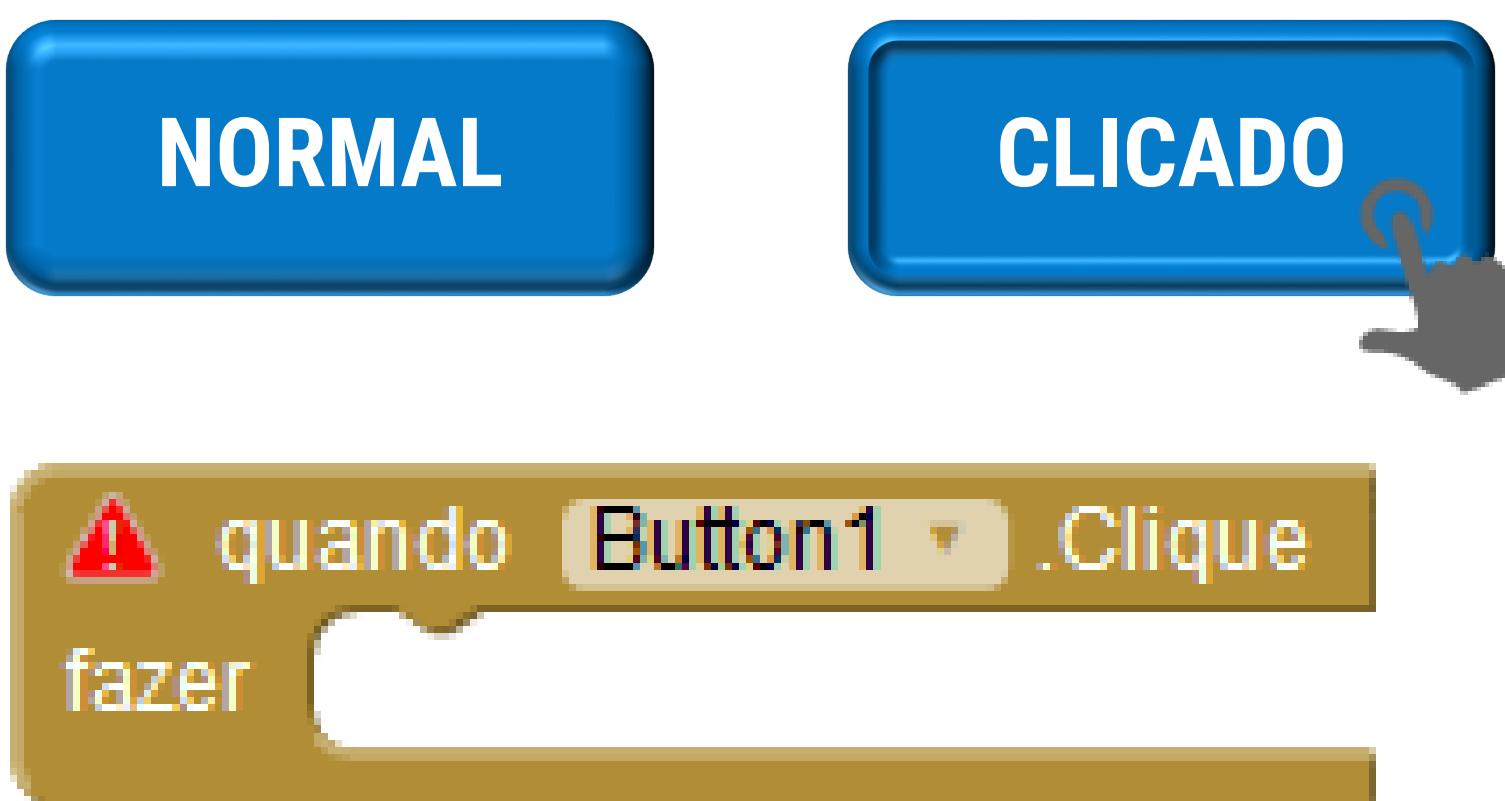
Vamos usar o bloco relacionado ao evento de Disparo do **Temporizador1**.

The image shows the Scratch script editor interface. On the left, the 'Blocos' (Blocks) palette is open, displaying categories like 'Internos' (Internal blocks), 'Screen1', and 'Qualquer componente' (Any component). In the center, the 'Visualizador' (Stage) shows a script starting with a 'when [Temporizador1] .Disparo' (when [Temporizador1] trigger) event. The script then contains four 'chamar [Temporizador1 v .Add [Days v instantе quantity]]' blocks, each with different parameters: 'Days', 'instante', and 'quantity'. Below these is another 'chamar [Temporizador1 v .Add [Duration v instantе quantity]]' block. Further down are two more 'chamar [Temporizador1 v .Add [Hours v instantе quantity]]' blocks, and finally a 'chamar [Temporizador1 v .Add [Minutes v instantе]]' block at the bottom.

MOVER O MOSQUITO PELA TELA

EVENTO

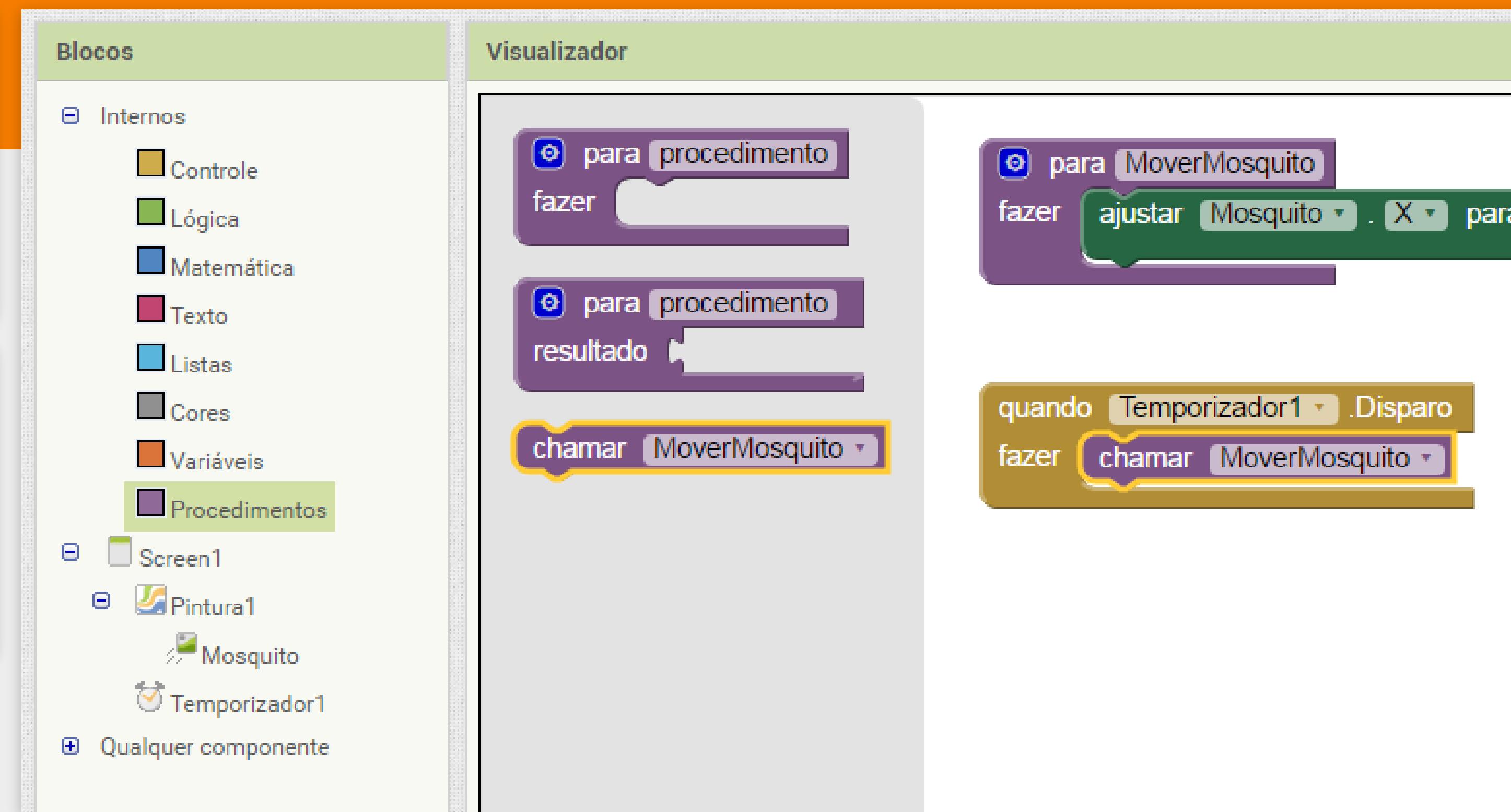
Um evento é algo que acontece, por exemplo, quando pressionamos um botão. O despertar de um alarme também pode ser considerado um evento.



MOVER O MOSQUITO PELA TELA

A cada 0,5 segundos serão executadas as instruções que estão dentro do bloco de Disparo do **Temporizador1**.

Clique em “Procedimentos”,
e arraste o bloco “**chamar
MoverMosquito**” para dentro
do bloco de Disparo.

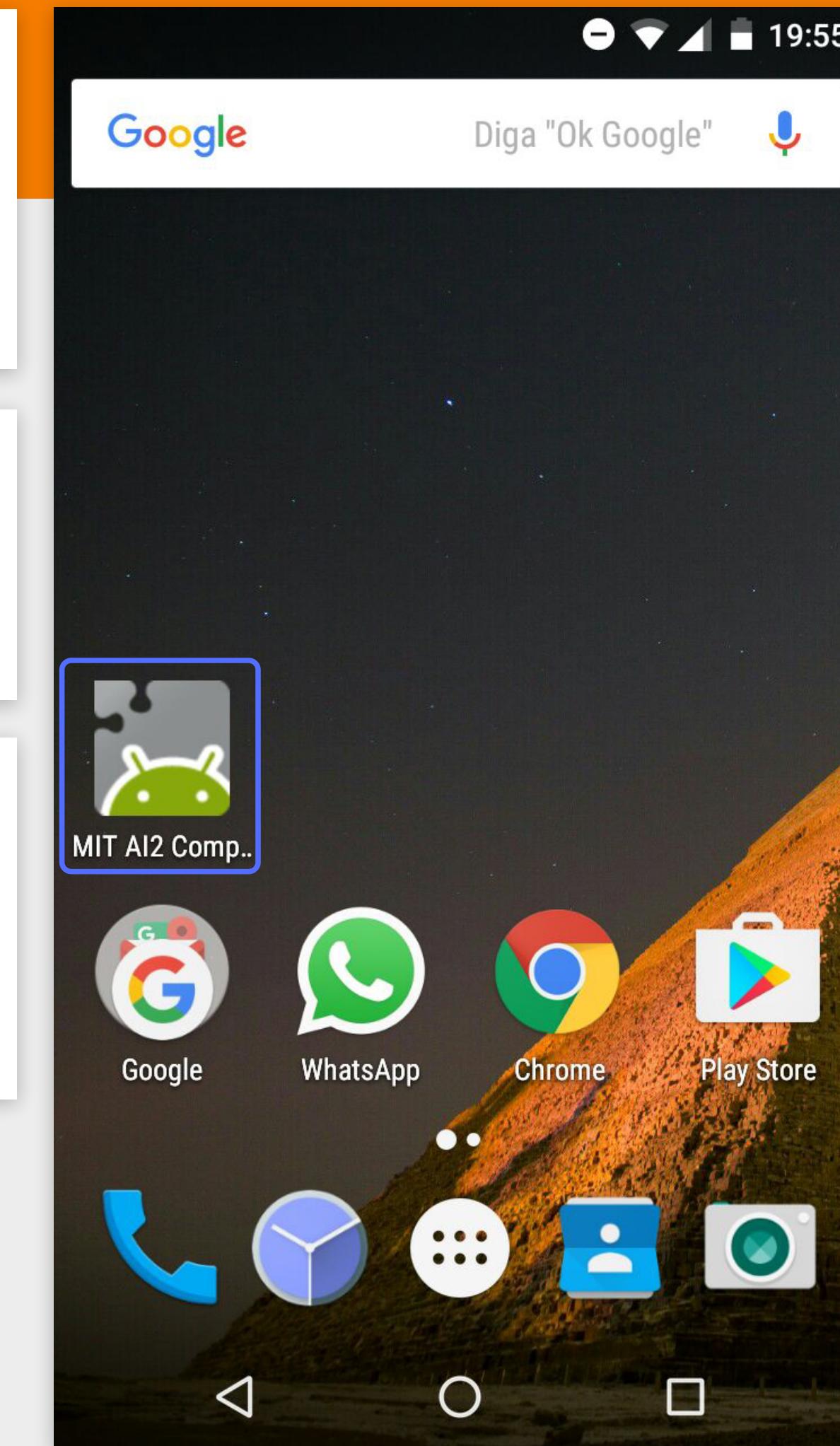


MOVER O MOSQUITO PELA TELA

Pronto! Agora nosso mosquito deve estar se movendo sozinho na tela.

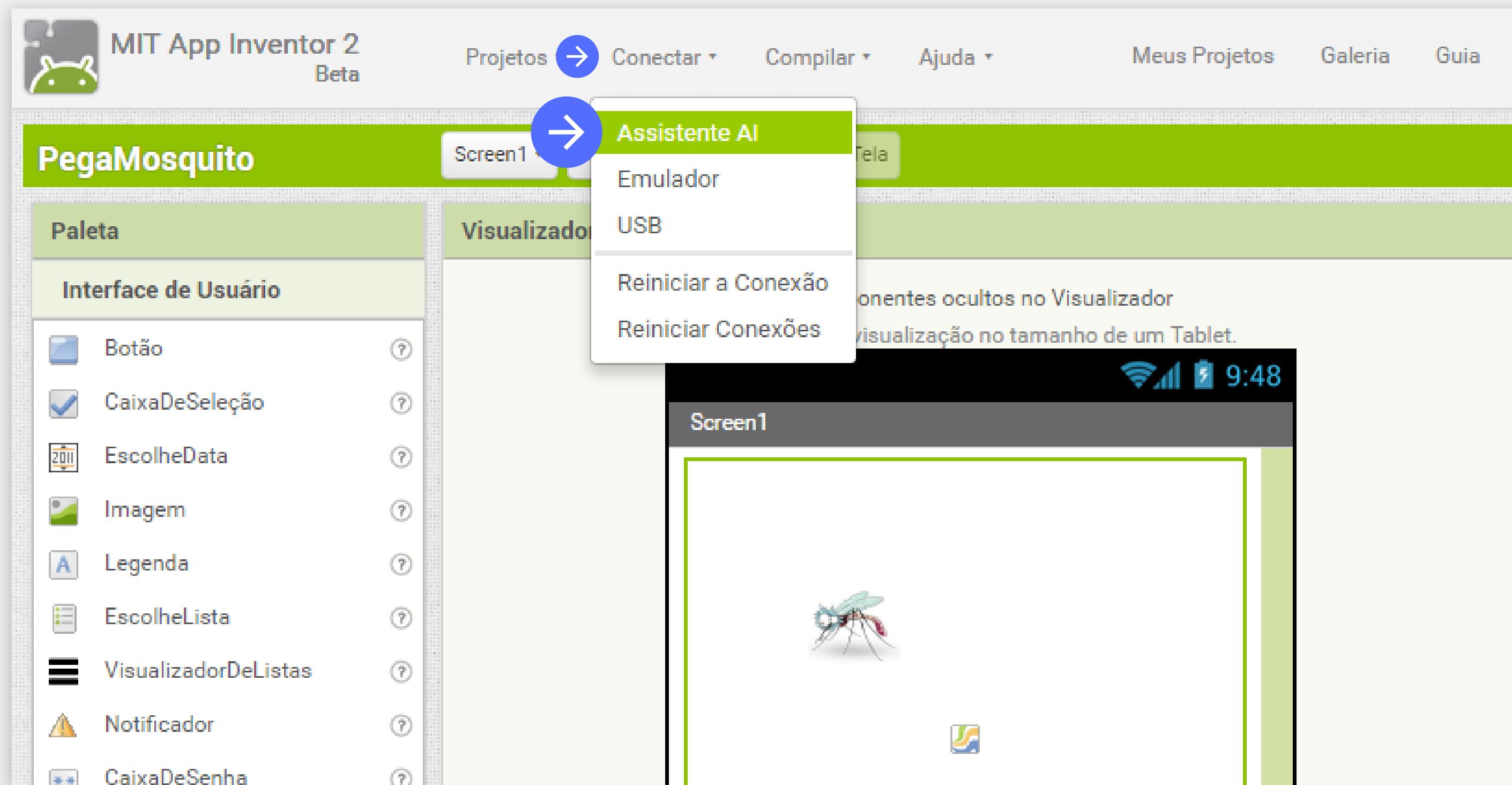
Vamos testar no celular como está ficando o jogo!

Pegue o seu celular e abra o aplicativo "[MIT AI2 Companion](#)".



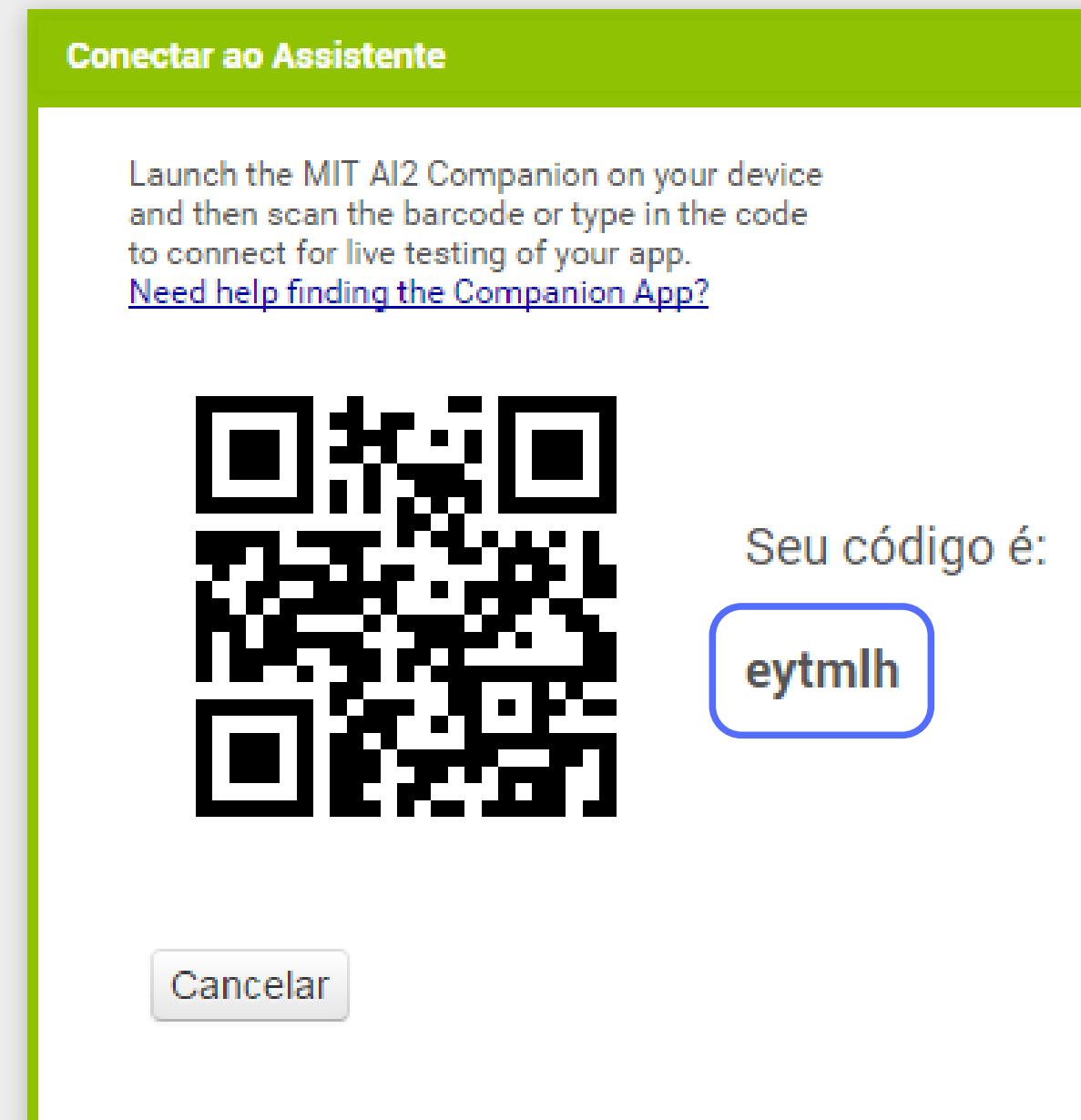
MOVER O MOSQUITO PELA TELA

No AppInventor clique em “**Conectar**” -> “**Assistente AI**”.



MOVER O MOSQUITO PELA TELA

No aplicativo do celular clique em “**scan QR code**” e aponte a câmera para o símbolo da tela do computador. Pronto, o seu celular está conectado com o programa que temos no computador.



MIT App Inventor 2

type in the 6-character code
-or-
scan the QR code

Six Character Code

connect with code

scan QR code

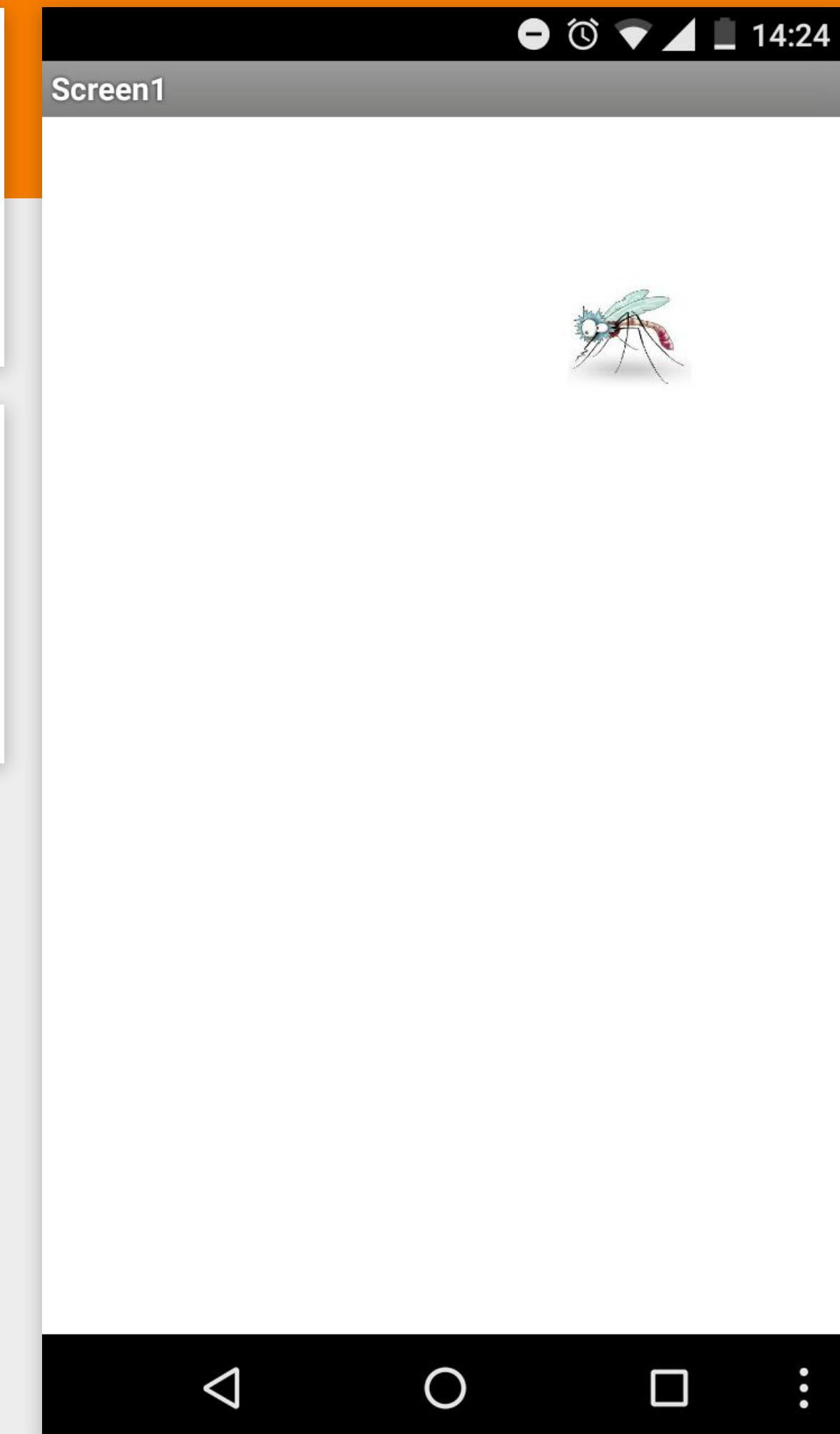
DICA

Se o seu celular não tiver câmera para usar o “scan”, você pode digitar o código indicado e clicar em “connect with code”.

MOVER O MOSQUITO PELA TELA

Você deve notar que o mosquito só está se movendo para os lados.

Temos que fazê-lo se movimentar em todas as direções, X e Y.

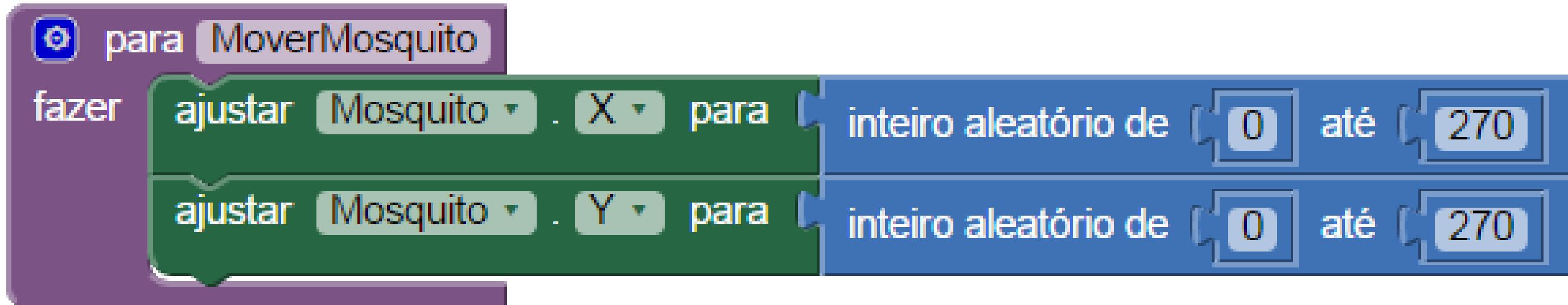


MOVER O MOSQUITO PELA TELA

Vamos seguir os mesmos passos anteriores para alterar também a coordenada Y do **Mosquito**.

O procedimento **MoverMosquito** deve ficar como a imagem ao lado.

Note como o jogo é atualizado automaticamente no celular!



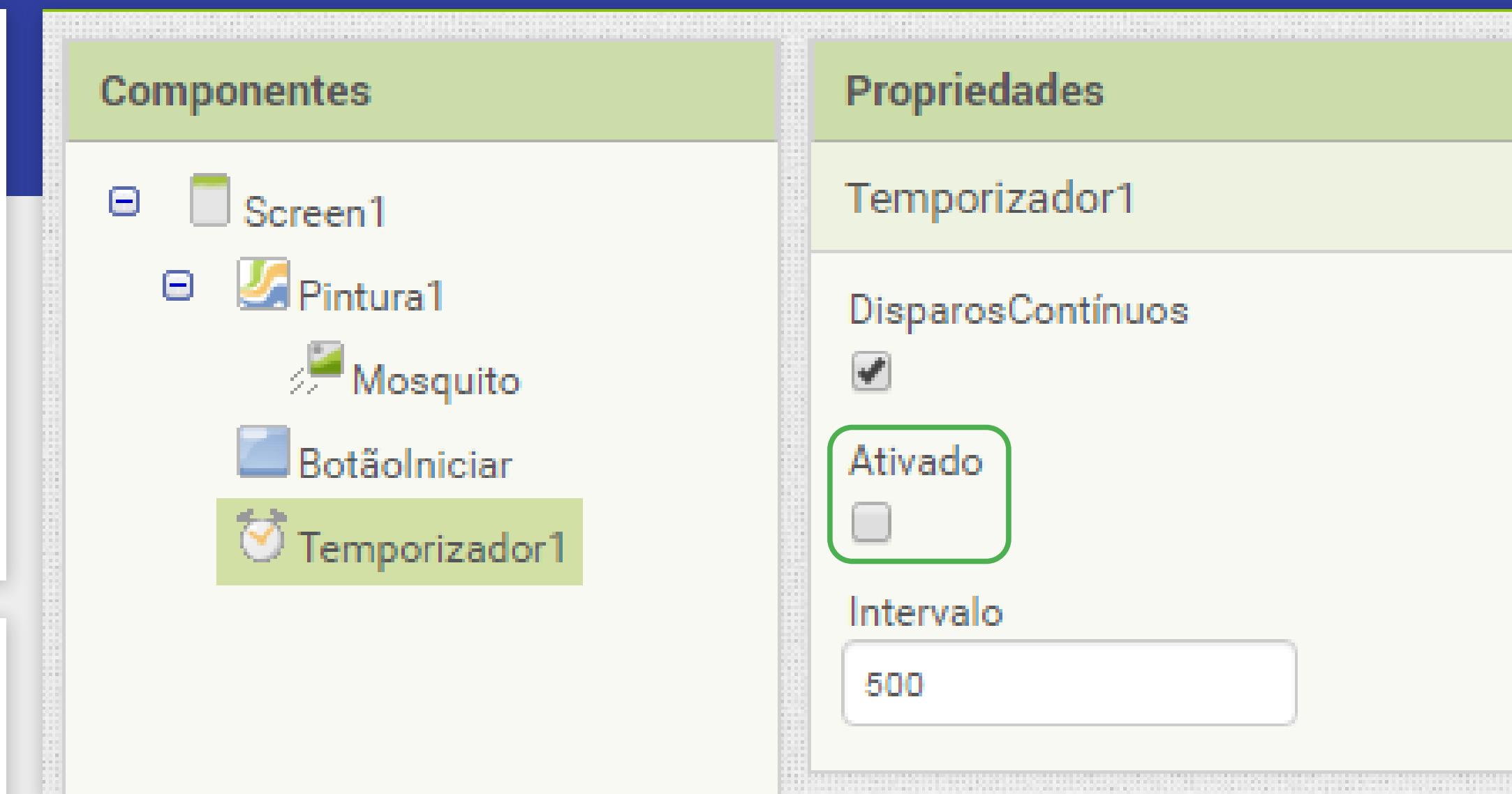
INICIAR O JOGO



INICIAR O JOGO

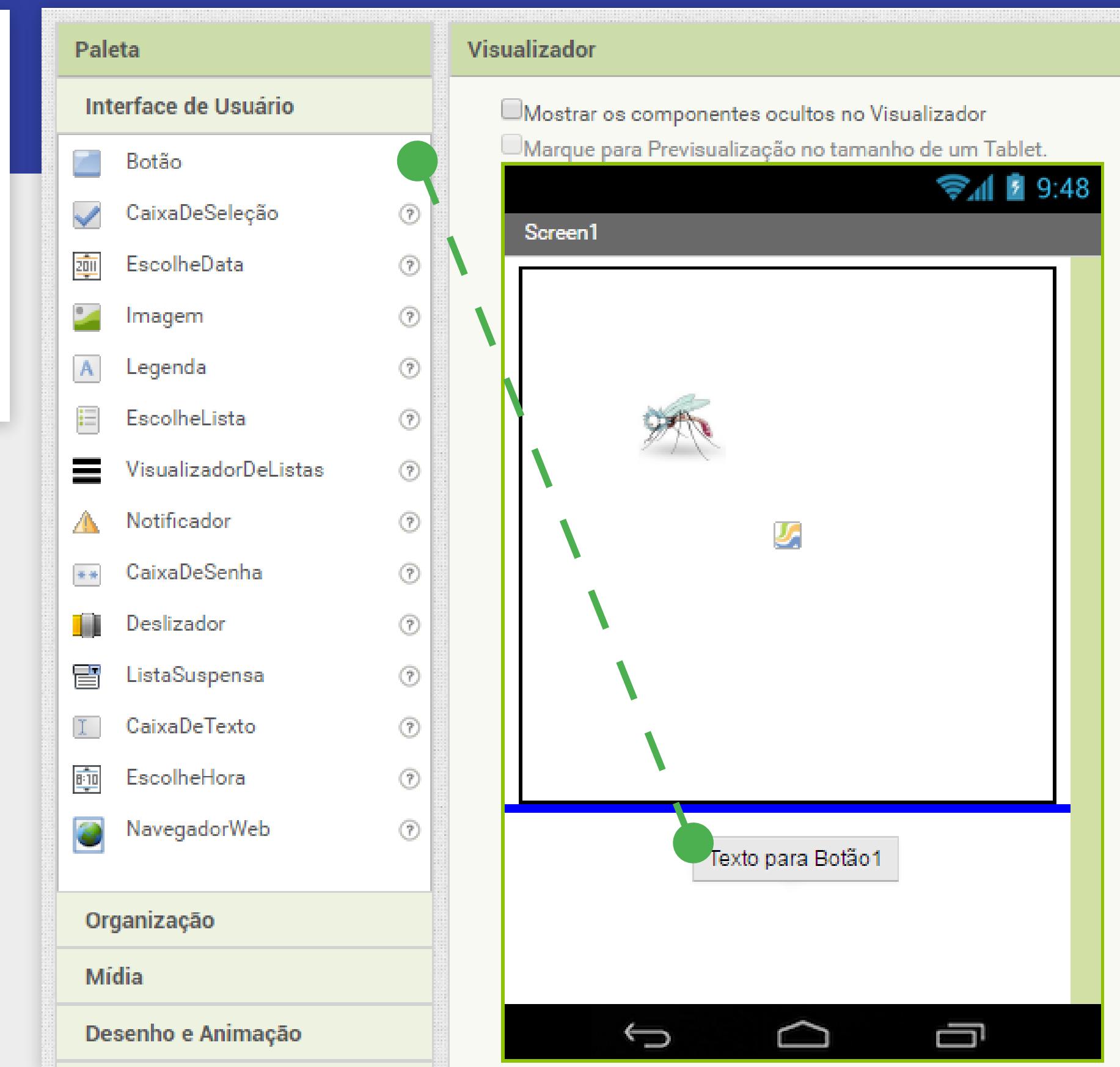
Primeiro, vamos voltar para a tela de “Designer” e desativar o **Temporizador1**. Assim, ele não vai mais disparar o movimento do mosquito até que o botão **Iniciar Jogo** seja clicado.

O mosquito está se movimentando a cada disparo do **Temporizador1**. Contudo precisamos que isso seja realizado somente quando formos jogar. Para isto, vamos inserir um botão de **Iniciar Jogo**.



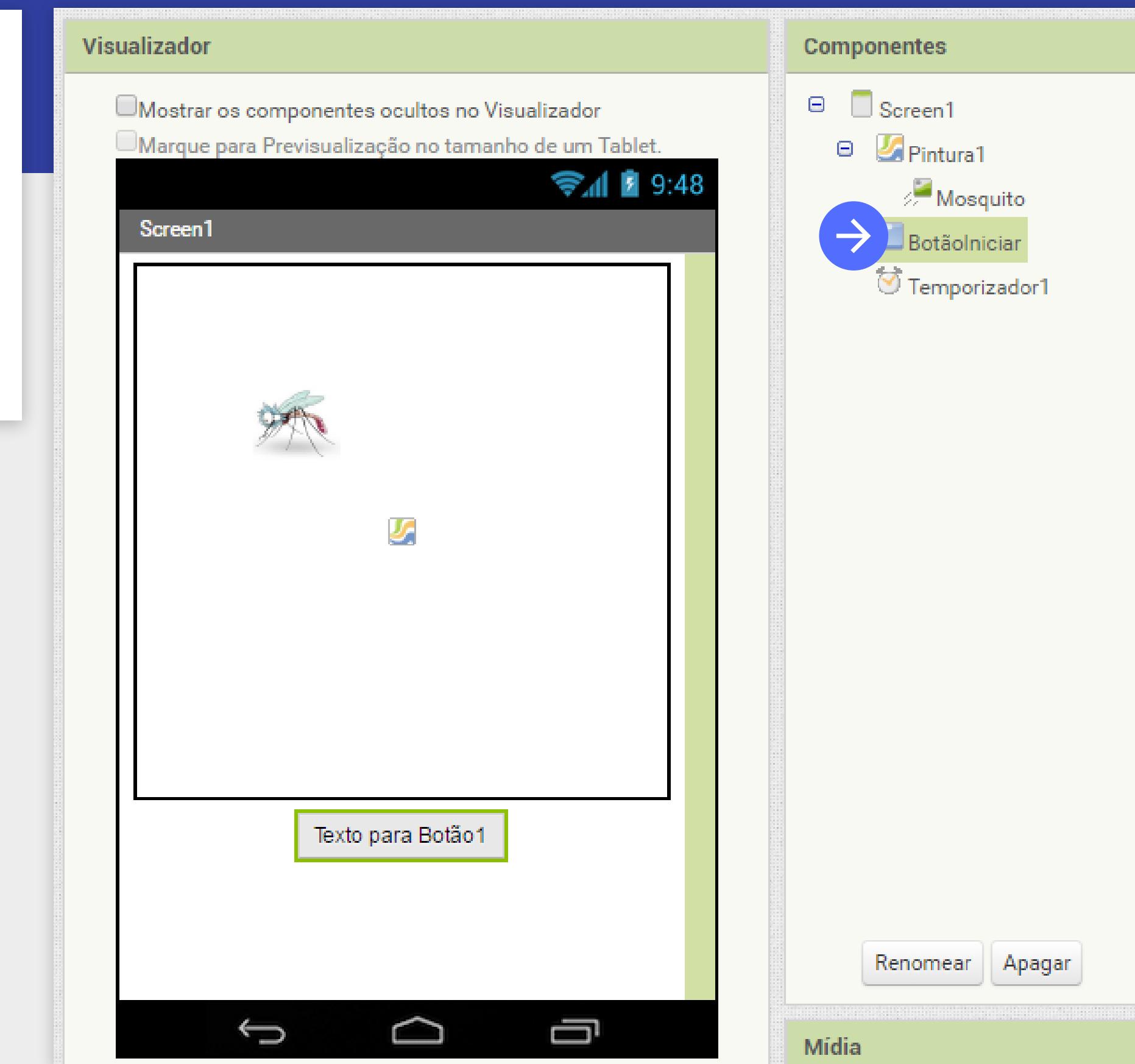
INICIAR O JOGO

Na paleta de “Interface de Usuário”, arraste o **Botão** para a tela, preferencialmente embaixo da área da **Pintura1**.



INICIAR O JOGO

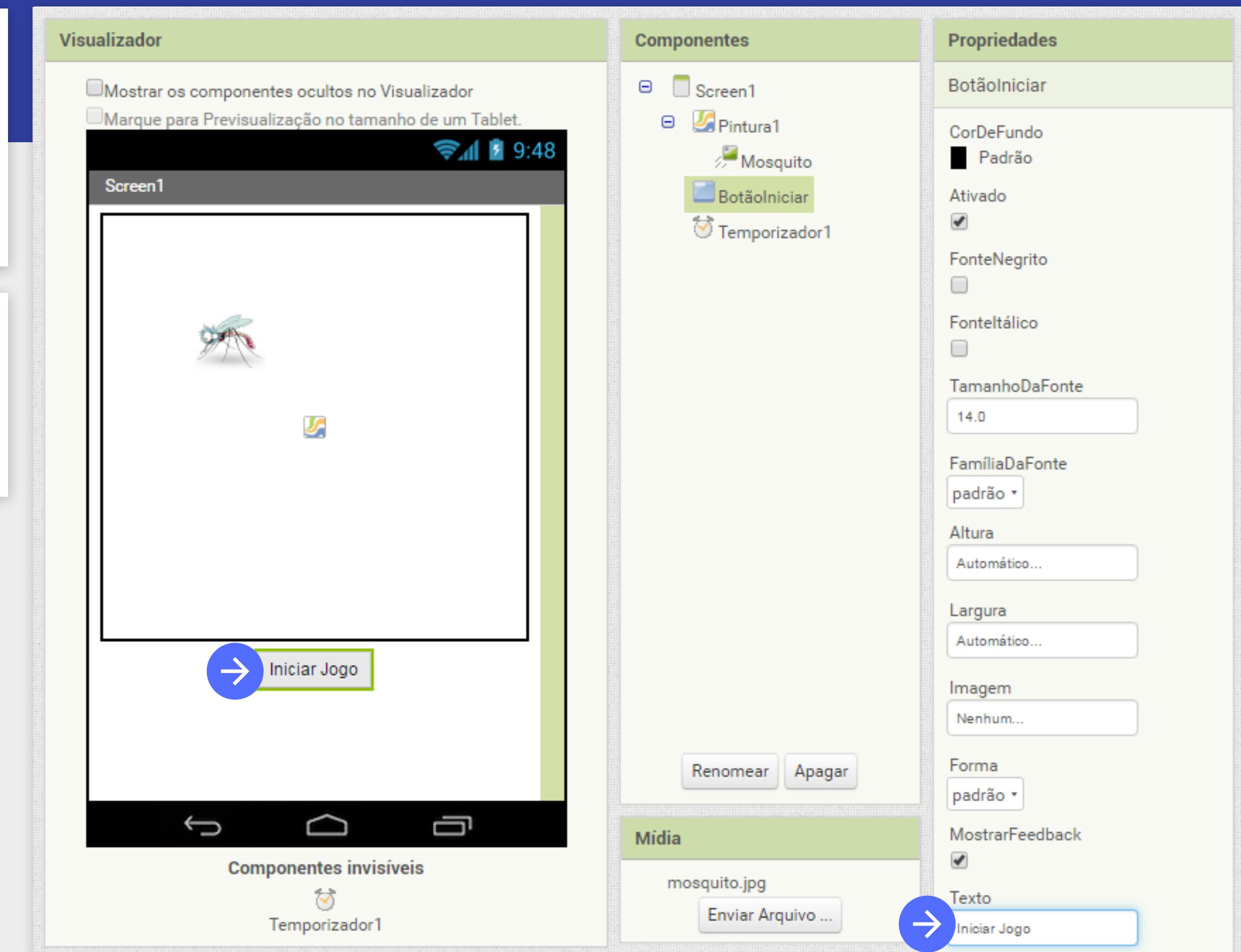
Agora, da mesma maneira que renomeamos o **Mosquito**, vamos renomear o componente -**Botao1** para **BotãoIniciar**.



INICIAR O JOGO

Também devemos trocar a propriedade do **BotãoIniciar** para ele apresentar o texto “Iniciar Jogo”.

Veja no seu celular se o botão está aparecendo corretamente!

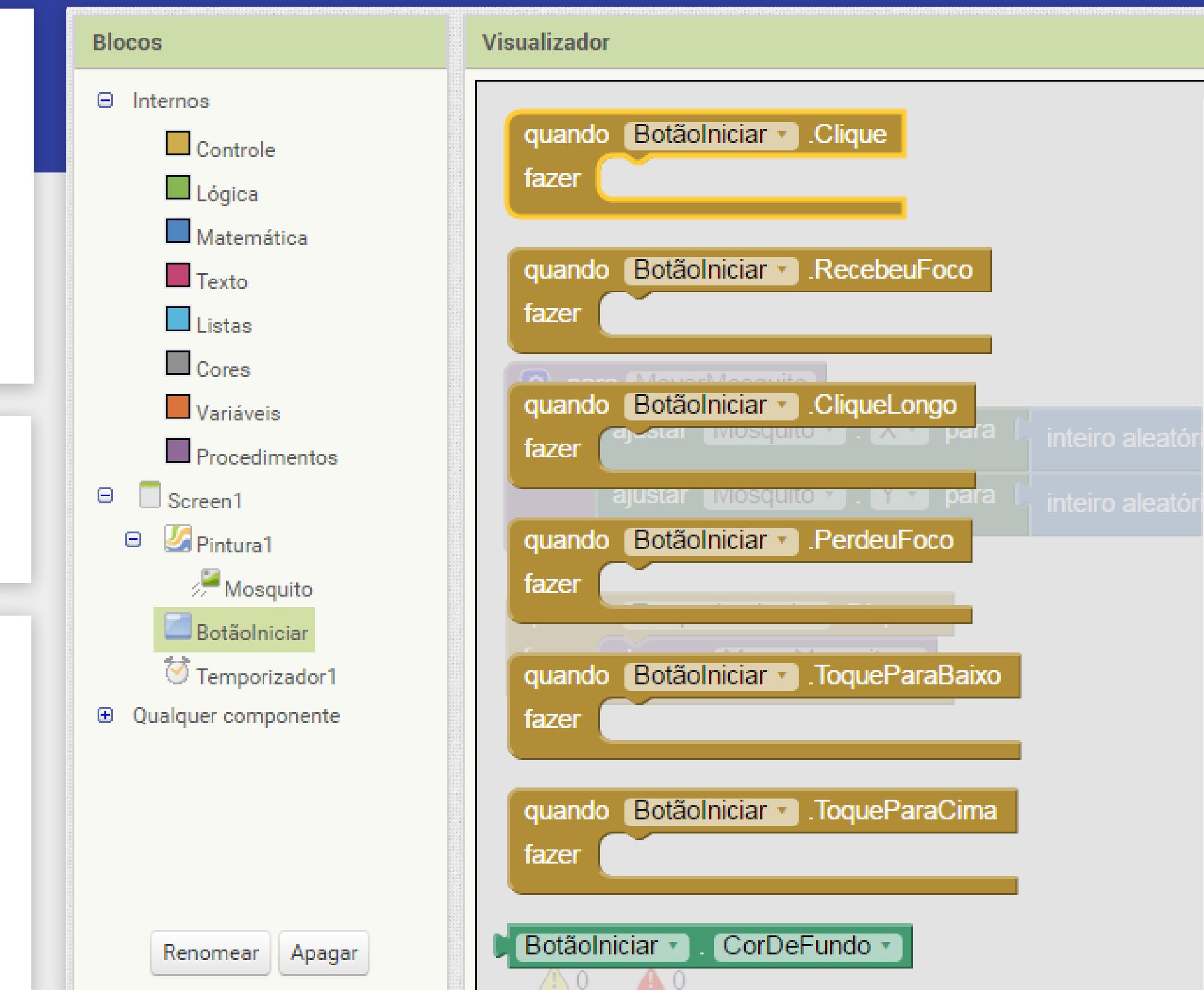


INICIAR O JOGO

Você deve ter verificado que ao clicar no botão nada está acontecendo, por isto, precisamos programar a funcionalidade de iniciar o jogo.

Volte para a tela de **Blocos**.

Vamos adicionar o bloco **quando BotãoIniciar.clique**. Este bloco se encontra nos blocos referentes ao **BotãoIniciar**.



INICIAR O JOGO

Quando o **BotãoIniciar** for clicado vamos ativar o **Temporizador1** para o jogo começar.

Clique em **Temporizador1** e escolha o bloco **ajustar Temporizador1.Ativado**, como mostrado ao lado.

The image shows the Scratch script editor interface. On the left is the 'Blocos' (Blocks) palette, which is divided into sections: 'Internos' (Internal blocks), 'Screen1' (with sub-sections for 'Pintura1', 'Mosquito', 'BotãoIniciar', and 'Temporizador1'), and 'Qualquer componente'. The 'Temporizador1' section is highlighted with a blue arrow icon. On the right is the 'Visualizador' (Stage) area where a script is displayed:

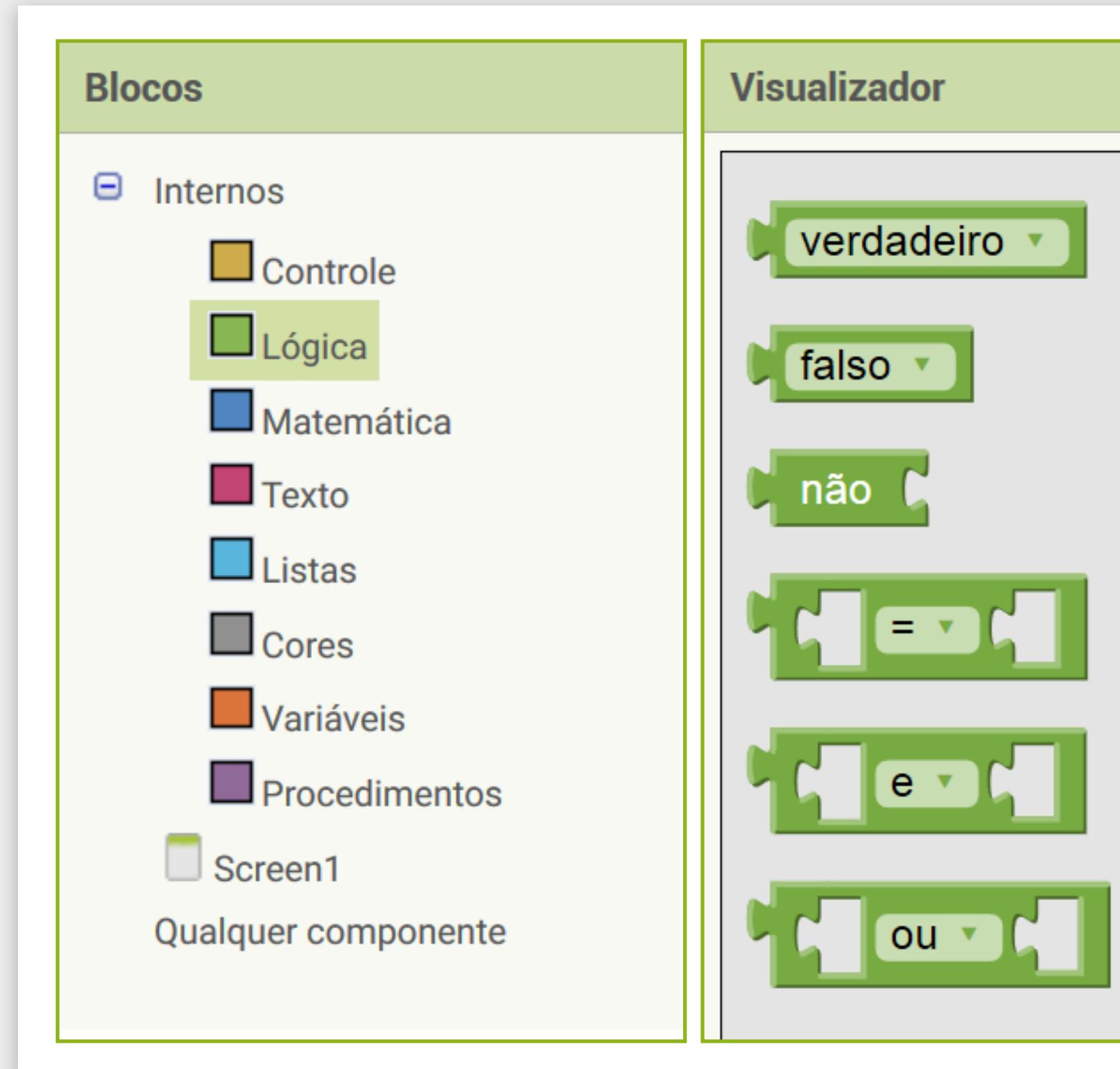
```
para [MoverMosquito]
fazer ajustar [Mosquito v] [X] para [inteiro aleatório]
ajustar [Mosquito v] [Y] para [inteiro aleatório]
quando [Temporizador1 .Disparo]
fazer chamar [MoverMosquito v]
quando [BotãoIniciar .Clique]
fazer ajustar [Temporizador1 .Ativado] [para []]

```

INICIAR O JOGO

Lembra quando desativamos o **Temporizador1** na tela de “Designer”?

A propriedade “Ativado” do **Temporizador1** pode ter dois valores, Verdadeiro, ou Falso.



ajustar Temporizador1 . Intervalo para 500

TEMPO ENTRE DISPAROS É DE 500 MILISEGUNDOS

ajustar Temporizador1 . Ativado para verdadeiro

SE A VARIÁVEL ATIVADO TIVER VALOR VERDADEIRO, VAI DISPARAR!

ajustar Temporizador1 . Ativado para falso

SE A VARIÁVEL ATIVADO TIVER VALOR FALSO, NÃO VAI DISPARAR!

INICIAR O JOGO

Para que o nosso botão ative o **Temporizador1**, selecione o bloco de **Lógica** com o valor **Verdadeiro**.

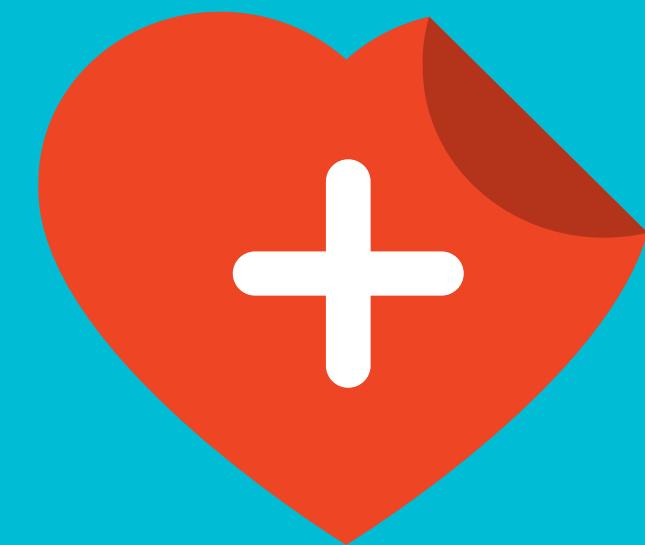
O comando do **BotãoIniciar** deve ficar assim:



```
quando [BotãoIniciar] .Clique
fazer [ajustar Temporizador1 . Ativado para verdadeiro]
```

INICIAR O JOGO

Teste o seu aplicativo e verifique que o mosquito se move após apertar no **Botão Iniciar!**



VIDAS DO MOSQUITO

Agora vamos adicionar vida ao Mosquito.
Ele vai iniciar o jogo com 3 vidas.

Cada vez que o(a) jogador(a) conseguir clicar no mosquito devemos diminuir a sua vida em 1. Ou seja, quando o(a) jogador(a) clicar 3 vezes sobre o mosquito ele(a) ganha o jogo.

VIDAS DO MOSQUITO

Precisamos criar um bloco que armazene o quanto de vida o mosquito possui. Esses blocos que armazenam valores são chamados de **variáveis**.

Vamos utilizar uma variável global que será chamada de “**vidaRestante**”. Usando o bloco “**inicializar global nome para**” e atribuir o valor “3”.

The image shows the Scratch script editor interface. On the left, the 'Blocos' (Blocks) palette is open, displaying categories like Internos, Matemática, Texto, etc., with the 'Variáveis' category highlighted. On the right, the 'Visualizador' (Stage) shows a script consisting of several blocks:

- A yellow 'Control' block: **→ inicializar global [nome] para [3]**
- An orange 'Control' block: **obter [name v.]**
- An orange 'Control' block: **ajustar [name v.] para [value]**
- Two orange 'Control' blocks: **inicializar local [nome] para [3] dentro de [script area]**
- Two orange 'Control' blocks: **inicializar local [nome] para [3] dentro de [script area]**

VIDAS DO MOSQUITO

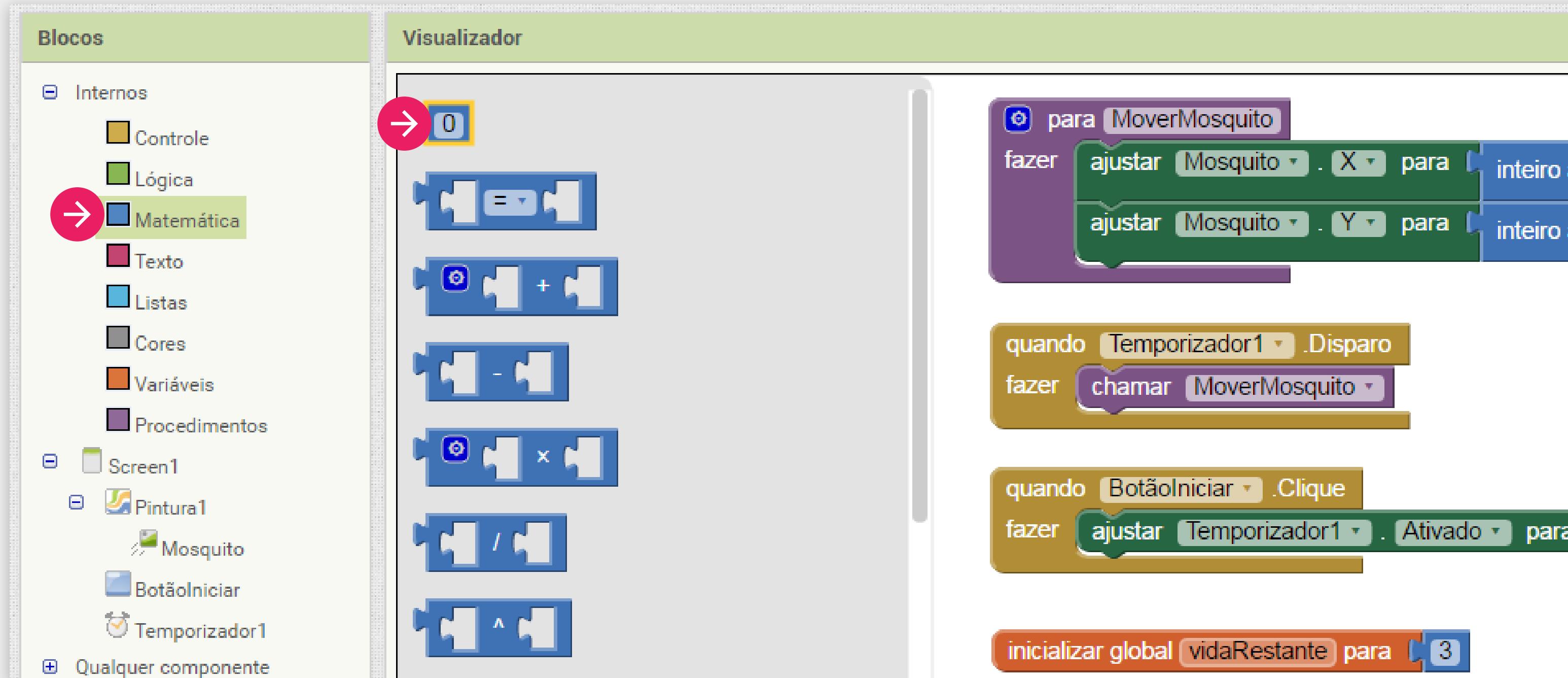
Variável é o nome dado ao local onde você pode armazenar informações e são utilizadas para lembrar de coisas como: a pontuação, o nome de um jogador ou até a velocidade do personagem.



VIDAS DO MOSQUITO

O bloco de **iniciar variável global** não precisa ser inserido dentro de outro bloco de função. Ele é executado automaticamente toda a vez que o aplicativo for aberto (inicializado).

Sendo assim, basta arrastarmos ele para a área do Visualizador e colocar o valor **Matemático “3”** junto a ele.



VIDAS DO MOSQUITO

Agora, toda vez que clicarmos no **Mosquito** devemos diminuir a variável “**vidaRestante**” em “1”. Para isto, adicionamos o bloco quando **mosquito.Tocou** pertencente aos blocos do componente **Mosquito**.

Esse bloco será executado toda vez que a imagem do **Mosquito** for tocada.

The image shows a Scratch script consisting of three blocks:

- A green **when green flag is shown** hat block.
- An orange **when [Mosquito] touches [mosquito]** sensor block with a yellow **when [mosquito] touches [mosquito]** hat block nested inside it. This nested block has an **x** and **y** input slot and a **fazer** script area.
- A purple **call [Mosquito v] [Quicar borda]** control block.

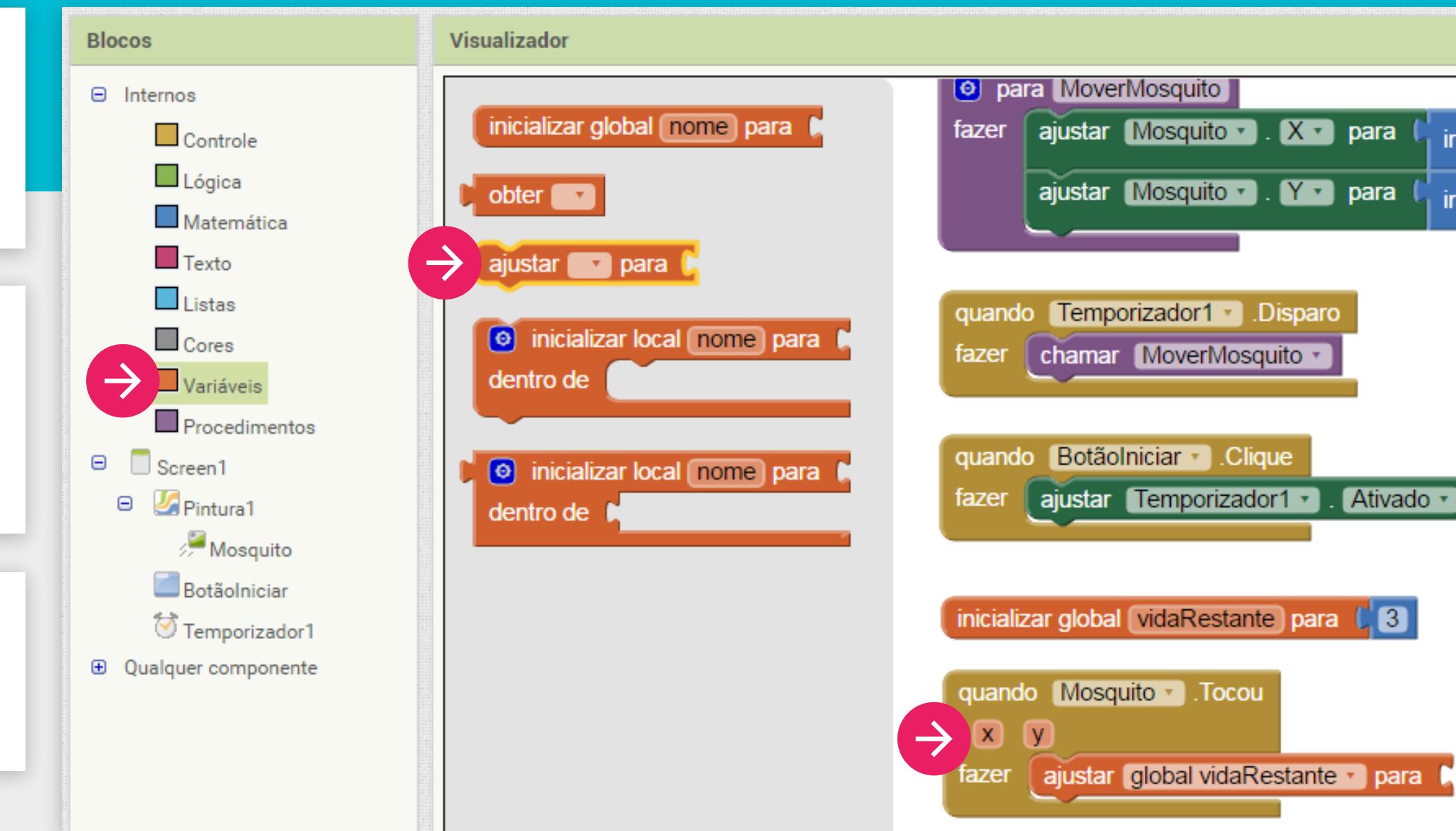
A red arrow points from the **when green flag is shown** block towards the **when [mosquito] touches [mosquito]** block.

VIDAS DO MOSQUITO

Vamos usar o bloco “ajustar ... para” que fica nos blocos de “Variáveis”.

Coloque o bloco dentro do procedimento “quando Mosquito.tocou”.

Escolha a variável “global vidaRestante” na lista ▾



VIDAS DO MOSQUITO

O novo valor vai ser o valor atual da vidaRestante - 1.

Para isso, escolha o bloco Matemático de Subtração.

The image shows a Scratch script editor window. On the left, the 'Blocos' (Blocks) palette is open, displaying categories: Internos (Internal), Controle (Control), Lógica (Logic), **Matemática** (Math), Texto (Text), Listas (Lists), Cores (Colors), Variáveis (Variables), and Procedimentos (Procedures). The 'Matemática' category is highlighted with a green background. On the right, the 'Visualizador' (Stage) shows a script consisting of four blocks stacked vertically: a blue '0' block, a blue 'operator' block with a minus sign, a blue 'operator' block with a plus sign, and a yellow 'operator' block with a minus sign.

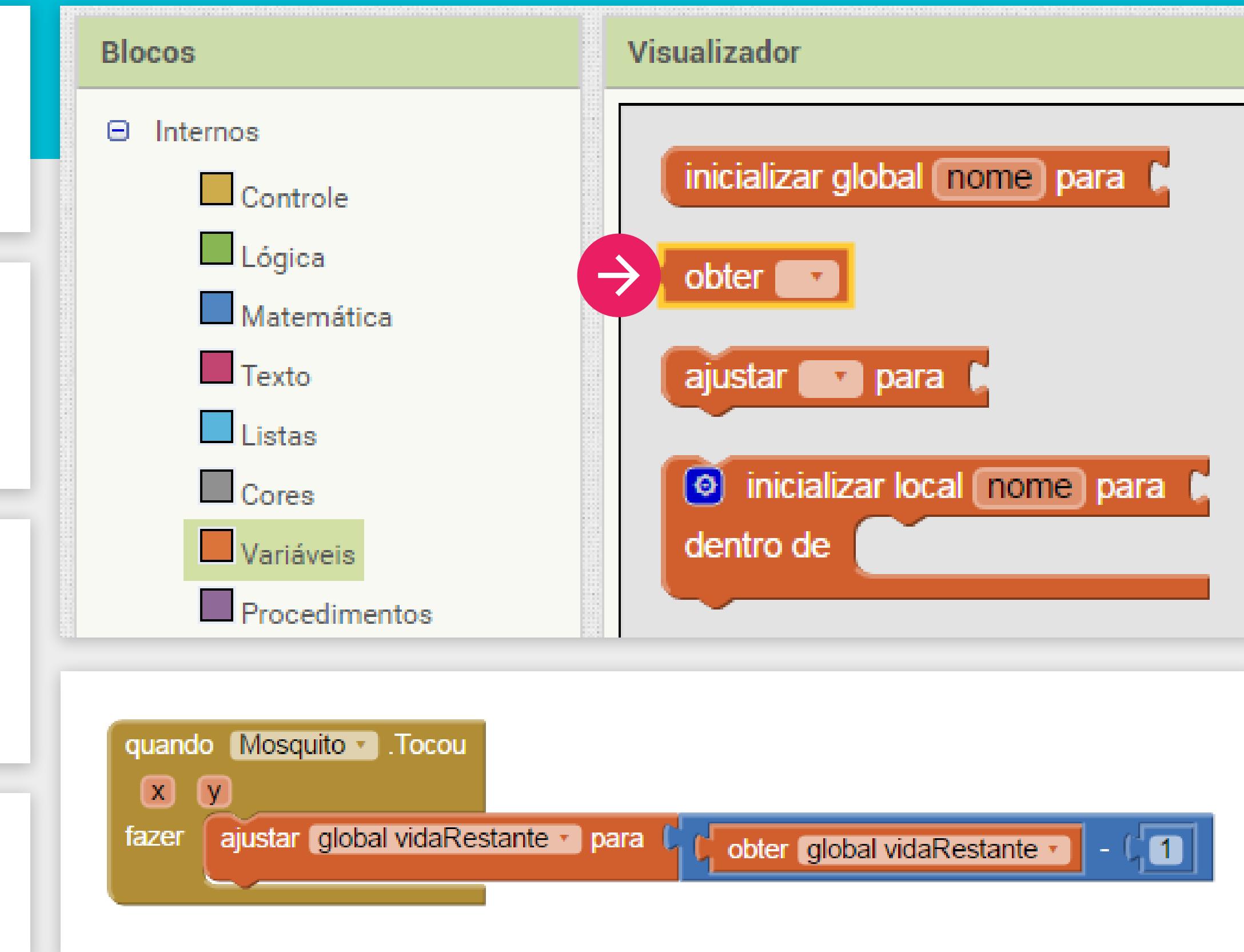
VIDAS DO MOSQUITO

O primeiro valor da subtração é o valor da variável vidaRestante.

Selecione o bloco “**obter ...**” e escolha “global vidaRestante” na lista.

O segundo valor é apenas o bloco Matemático de número (mude de 0 para 1).

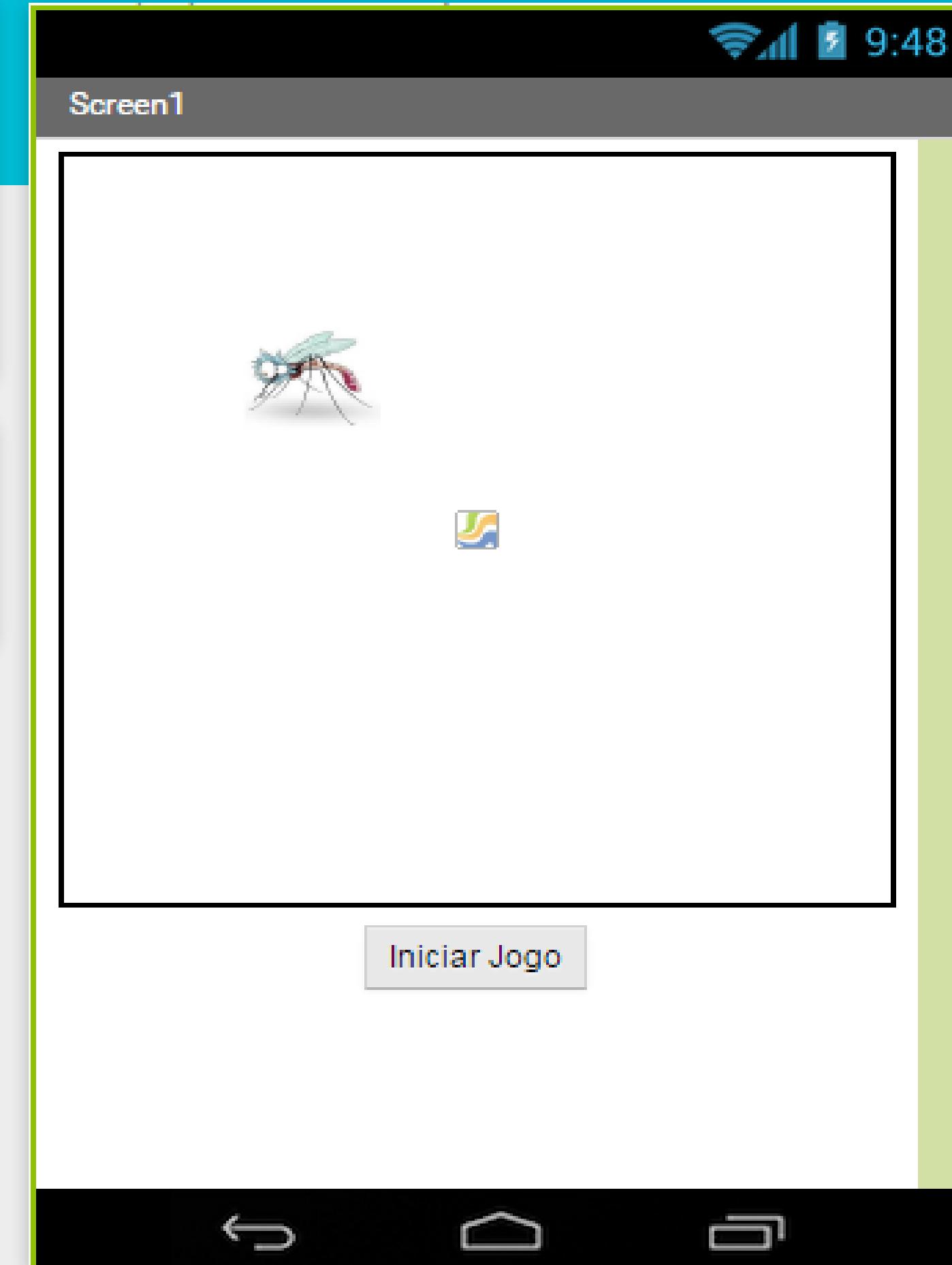
O procedimento vai ficar assim:



VIDAS DO MOSQUITO

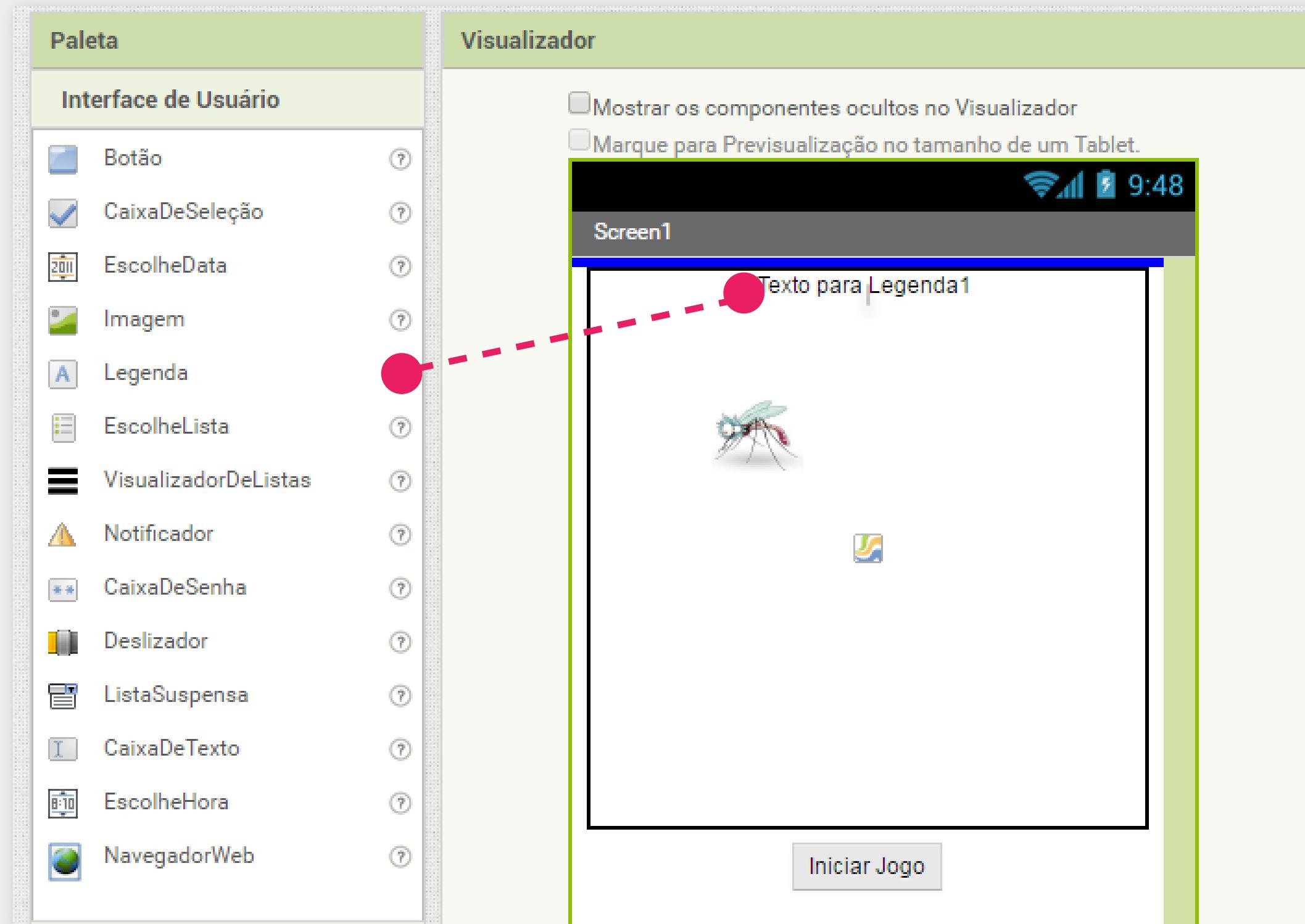
O nosso código de vida já está funcionando. Mas note que na tela do jogo não existe nada indicando para o jogador quanto é a vida do mosquito.

Precisamos mostrar o valor da vida na tela.



VIDAS DO MOSQUITO

Na tela de “Designer”, arraste o componente **Legenda** da Paleta para a Tela, preferencialmente acima da **Pintura1**.



VIDAS DO MOSQUITO

Teste o seu aplicativo, veja se está aparecendo corretamente o número de vidas do mosquito! Você percebeu que ao tocar no mosquito o valor da vida dele não está sendo alterado? Precisamos arrumar isto!

The screenshot shows the Construct 3 IDE interface. On the left is the 'Components' panel, which lists the following objects:

- Screen1
- Legend1 (highlighted with a red arrow)
- Pintura1
- Mosquito (highlighted with a red arrow)
- BotãoIniciar
- Temporizador1

On the right is the 'Properties' panel, showing properties for the selected 'Legend1' object:

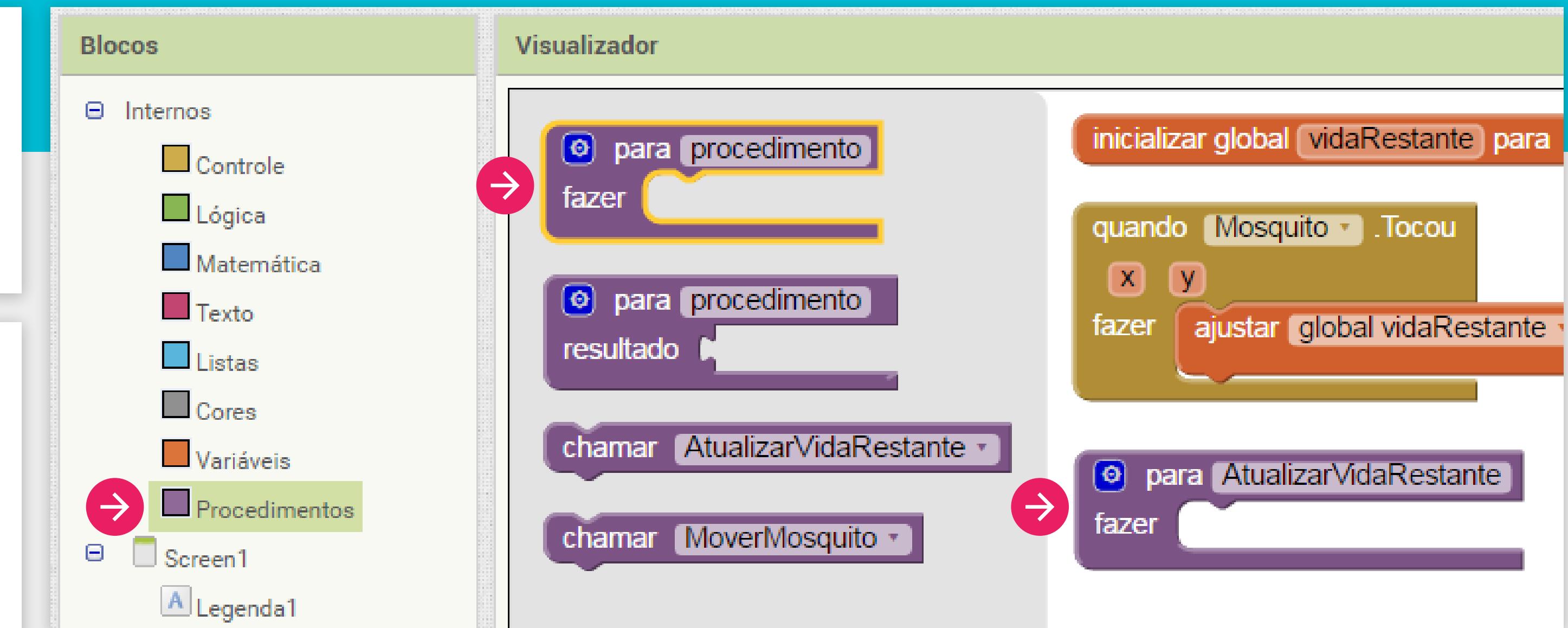
- CorDeFundo: Nenhum
- FonteNegrito: (checkbox)
- Fonteltálico: (checkbox)
- TamanhoDaFonte: 14.0
- FamíliaDaFonte: padrão
- TemMargens: (checkbox checked)
- Altura: Automático...
- Largura: Automático...
- Texto: Vidas: 3 (highlighted with a red arrow)
- AlinhamentoDoTexto: esquerda

At the bottom of the Properties panel are two buttons: 'Renomear' and 'Apagar'.

VIDAS DO MOSQUITO

Quando o mosquito for tocado, o valor da variável “vidaRestante” é diminuído, mas também devemos atualizar o texto contido na **Legenda1**.

Vamos criar um procedimento chamado de “AtualizarVidaRestante” que será responsável por atualizar o texto da **Legenda1** baseado no valor que temos na variável “vidaRestante”.



VIDAS DO MOSQUITO

Para alterar o valor do texto escrito no componente **Legenda1** utilizamos o bloco “**ajustar Legenda1.Texto**”.

The image shows a Scratch script in the 'Visualizador' (Scratch Stage) window. The script consists of the following blocks:

- ajustar [Legenda1 v] . TemMargens v para [C v]
- Legenda1 v . Altura v
- ajustar [Legenda1 v] . Altura v para [C v]
- ajustar [Legenda1 v] . PercentualDeAltura v para [C v]
- Legenda1 v . Texto v
- ajustar [Legenda1 v] . Texto v para [C v] (highlighted with a red arrow)
- Legenda1 v . CorDeTexto v
- ajustar [Legenda1 v] . CorDeTexto v para [C v]
- Legenda1 v . Visivel v

The 'Blocos' (Scratch blocks library) window is also visible on the left, showing categories like Internos, Screen1, and Qualquer componente. The 'Legenda1' component is highlighted with a red arrow.

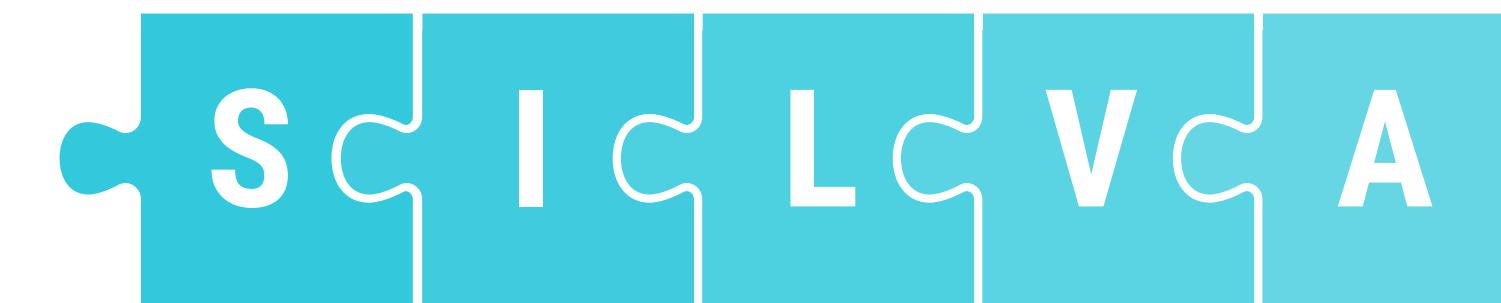
STRINGS

Um string é uma sequência de caracteres. O texto da Legenda1 é considerado uma String.

STRING NOME



STRING SOBRENOME



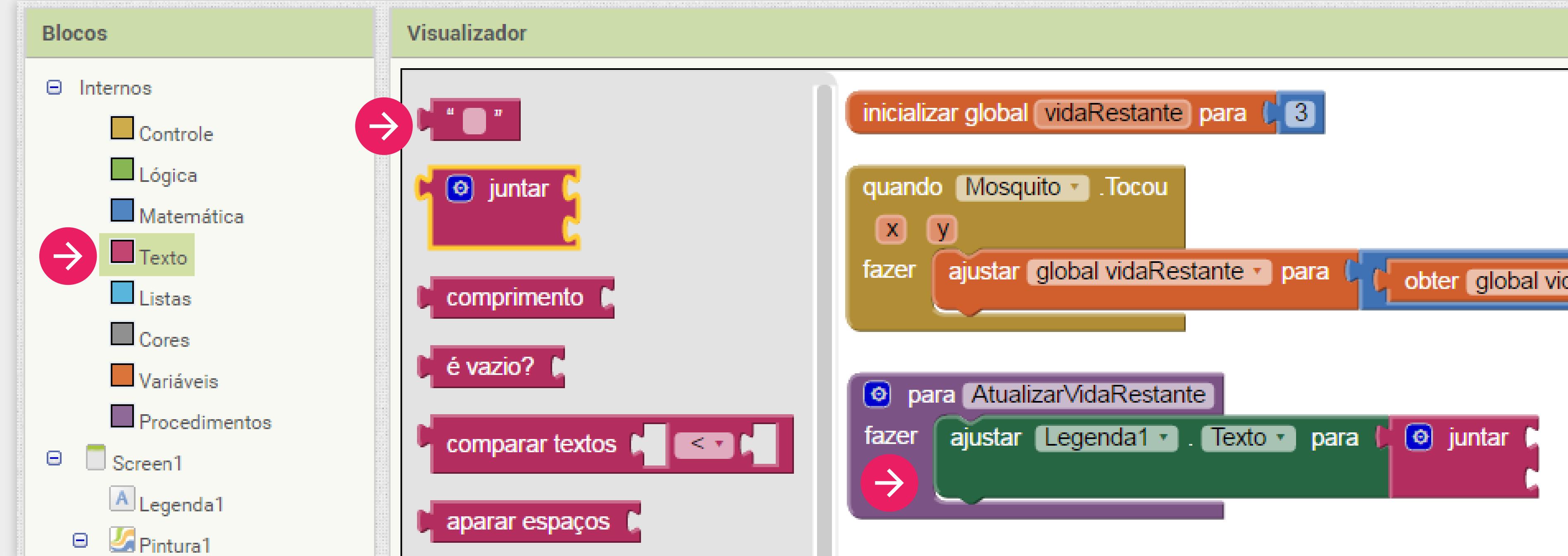
JUNTAR



VIDAS DO MOSQUITO

O texto da **Legenda1** sempre terá a palavra "Vida: " mais o valor da variável **vidaRestante**.

Por isso, precisamos juntar esses dois textos. Vamos utilizar o bloco de manipulação de texto **Juntar**.



VIDAS DO MOSQUITO

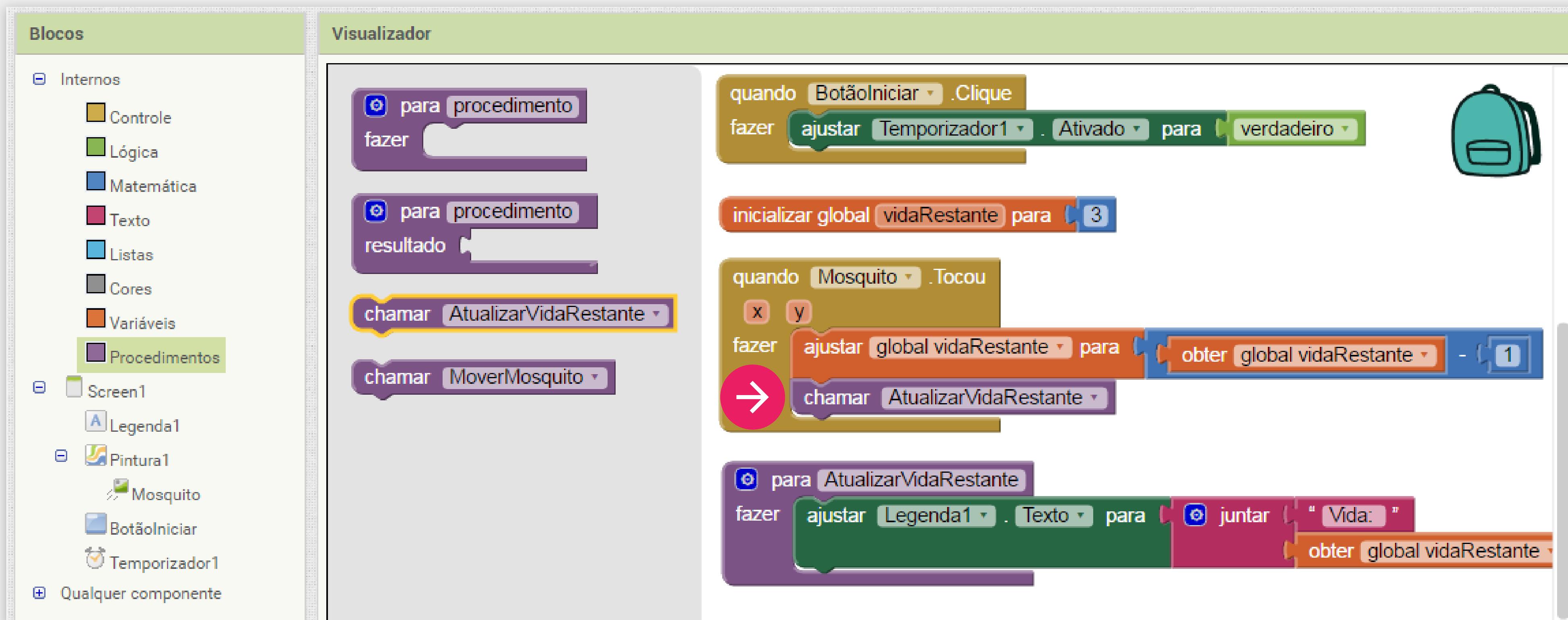
Encaixe no primeiro espaço o bloco de texto e escreva o valor “Vida: ”.

No segundo espaço encaixe o valor da variável “vidaRestante” (bloco **obter ...**).



VIDAS DO MOSQUITO

Não podemos esquecer de chamar esse procedimento para atualizar o texto na tela quando o mosquito for tocado.



VIDAS DO MOSQUITO

Teste o seu jogo e veja que agora as vidas estão diminuindo quando você toca no mosquito.

Porém, ainda está difícil saber quando você acertou o toque no mosquito ou não. Para ficar mais claro para o jogador, vamos vibrar o celular quando o mosquito for tocado.

DICA

Não se preocupe se o número de vidas ficar negativo! Isso acontece porque ainda não criamos o código para terminar o jogo. Logo faremos isso!

VIDAS DO MOSQUITO

Volte para a tela de “Designer”.

Para vibrar o celular, vamos inserir o componente **Som** da Paleta de “Mídia”.

O componente Som é invisível e vai ficar abaixo da tela, junto com o Temporizador.

Componentes invisíveis



Temporizador1



Som1

The screenshot shows the MIT App Inventor Designer environment. On the left is the Paleta (Palette) with categories like Interface de Usuário, Organização, Mídia, Desenho e Animação, Sensores, Social, Armazenamento, Conectividade, LEGO® MINDSTORMS®, and Experimental. The Mídia category is currently selected, showing components like CâmeraDeVídeo, Câmera, Escolhemagem, Tocador, Som, Gravador, ReconhecedorDeVoz, TextoParaFalar, ReprodutorDeVideo, and TradutorYandex. On the right is the Visualizador (Visualizer), which displays a mobile phone screen titled "Screen1". The screen shows a mosquito icon at the top with the text "Vidas: 3". Below it is a large empty rectangular area containing a small speaker icon labeled "Som1". At the bottom of the screen is a button labeled "Iniciar Jogo". The status bar at the top of the visualizer shows signal strength, battery level, and the time 9:48. The bottom of the visualizer also shows the MIT App Inventor navigation bar with icons for back, home, and search.

VIDAS DO MOSQUITO

Agora vamos voltar para a tela de “Blocos”

Temos que chamar o componente **Som1** para vibrar o telefone dentro do procedimento **mosquito.Tocou.**

O bloco que devemos utilizar
é chamar **Som1.Vibrar**.

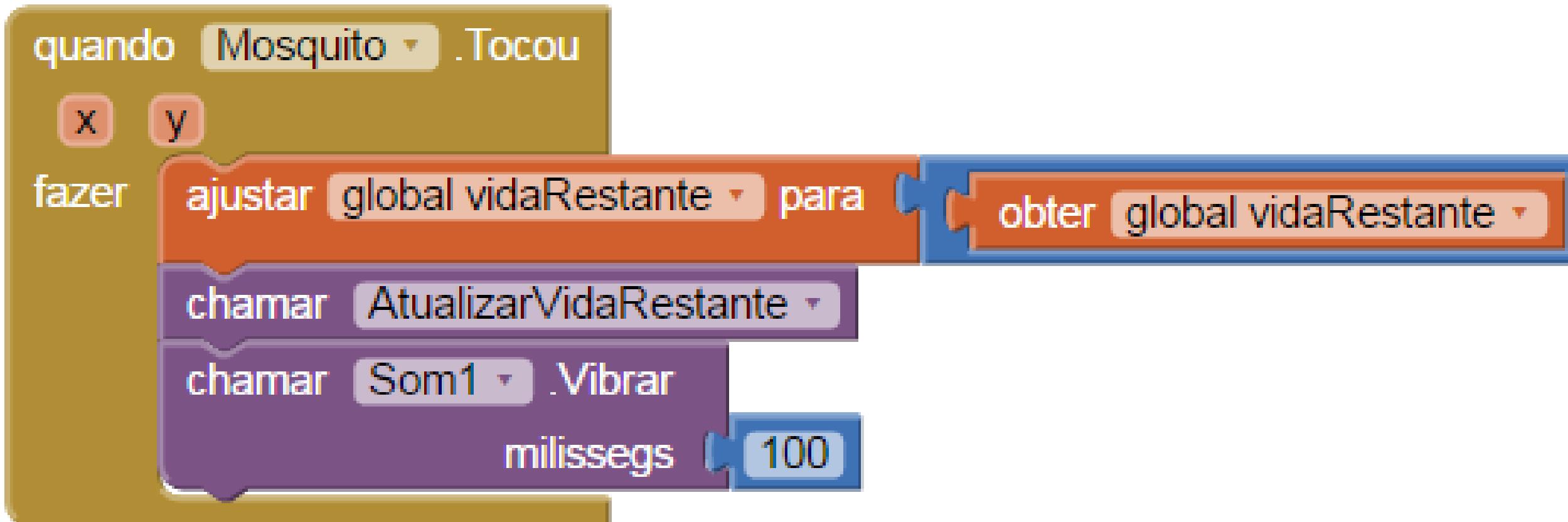
The image shows a Scratch script consisting of the following blocks:

- when [Som1 v] undefined
- say [mensagem v]
- do until [fazer v]
- call [Som1 v].Pausar
- call [Som1 v].Tocar
- call [Som1 v].Retomar
- call [Som1 v].Parar
- call [Som1 v].Vibrar [milisegs v]
- set [Som1 v] to [IntervaloMínimo v]
- adjust [Som1 v] [IntervaloMínimo v] by [para v]

A red arrow points from the "Som1" component in the "Qualquer componente" category of the blocks palette to the "Som1" component in the first "call" block of the script.

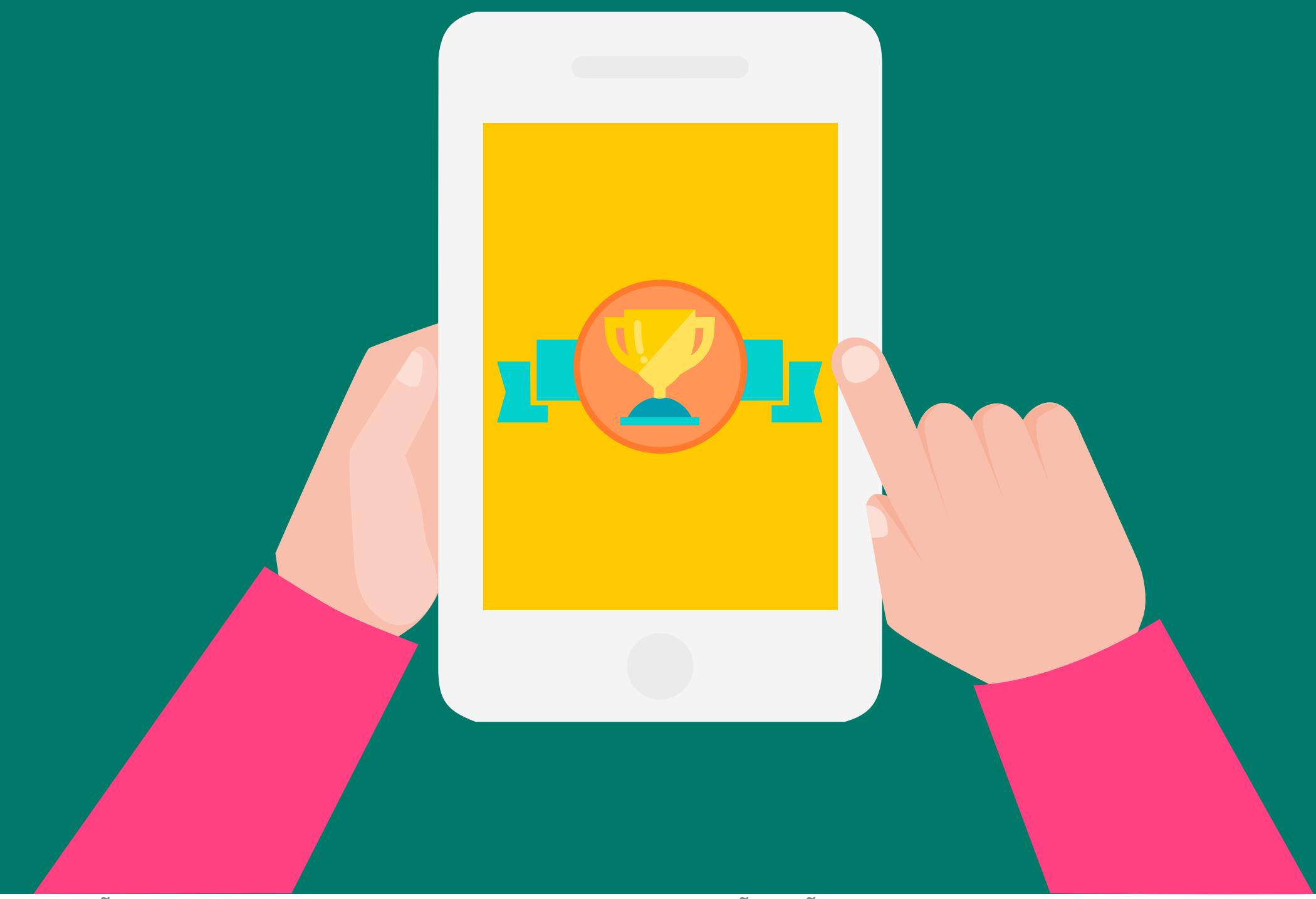
VIDAS DO MOSQUITO

Vamos definir o tempo que desejamos que ele vibre como “100” milissegundos.



Agora você pode testar seu jogo novamente. Clique sobre o mosquito e veja se o seu celular está vibrando!

FINALIZAR O JOGO



Como você deve ter percebido o nosso mosquito ainda não está morrendo, ele está ficando com a vida negativa. Para resolver este problema precisamos saber quando terminar o jogo, ou seja, quando o mosquito ficar com a vida igual a “0”.

Para isto, depois de vibrar o telefone quando o Mosquito foi tocado, devemos verificar se a sua vida restante é maior ou igual a zero para decidir se o jogo continua ou não.

TOMAR DECISÕES

VIDA DO MOSQUITO = 0



FALSO



VERDADEIRO

CONTINUAR O JOGO

TERMINAR O JOGO

A Scratch script consisting of a yellow "se" hat block with a blue gear icon, followed by a yellow "então" slot block containing the text "VALOR DA VARÍAVEL DA VIDA = 0" and another yellow "então" slot block containing the text "Chame procedimento para terminar jogo".

Para fazer esta verificação vamos usar o Bloco de Controle “Se, então”.

EXPRESSÕES BOOLEANAS

Computadores tomam suas decisões baseados em perguntas, cujas respostas, são duas: verdadeiro ou falso.

[=]

**COMPARA SE DOIS
VALORES SÃO IGUAIS**

5=5

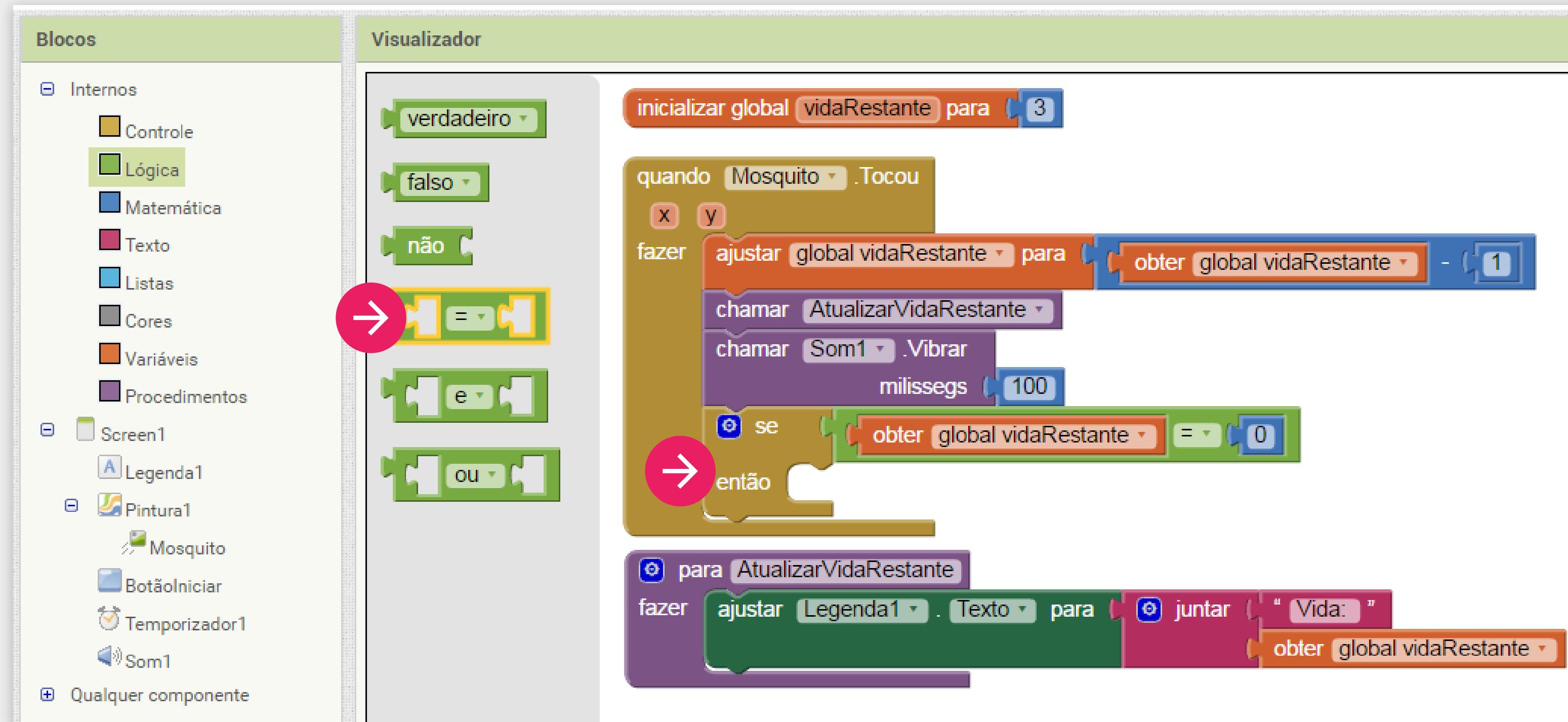
VERDADEIRO

5=2

FALSO

FINALIZAR O JOGO

Para a comparação, utilizamos a operação Lógica “vidaRestante = 0”.



FINALIZAR O JOGO

Se o valor de “vidaRestante” for igual a zero devemos chamar um procedimento responsável por terminar o jogo.

Vamos criar um novo procedimento “FinalizarJogo” através dos Blocos de Procedimento.



FINALIZAR O JOGO

O QUE DEVE SER FEITO AO FINALIZAR O JOGO?

1. Precisamos fazer o mosquito parar de se movimentar.
2. Precisamos informar na tela que o mosquito foi eliminado.
3. Precisamos parar de diminuir o numero de vidas do mosquito e atualizar a tela ao clicar no mosquito.

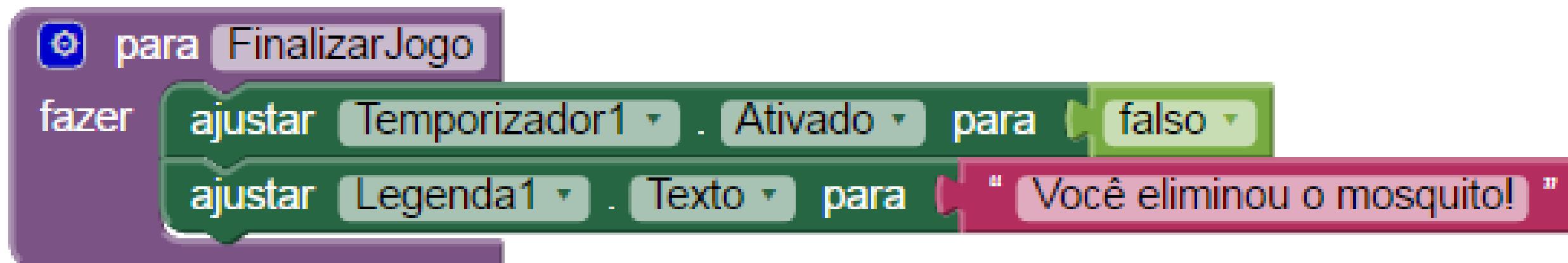
FINALIZAR O JOGO

1. Precisamos fazer o mosquito parar de se movimentar.
 - a) Para solucionar o primeiro problema, vamos desativar o Temporizador.
 - b) Adicione o bloco “ajustar Temporizado1.Ativado” informando o valor “falso”.



FINALIZAR O JOGO

2. Precisamos informar na tela que o mosquito foi eliminado.
 - a) Vamos mudar o texto da Legenda1 para informar “Você eliminou o mosquito!”.
 - b) Use o bloco “ajustar Legenda1.Texto” e escreva o texto acima.

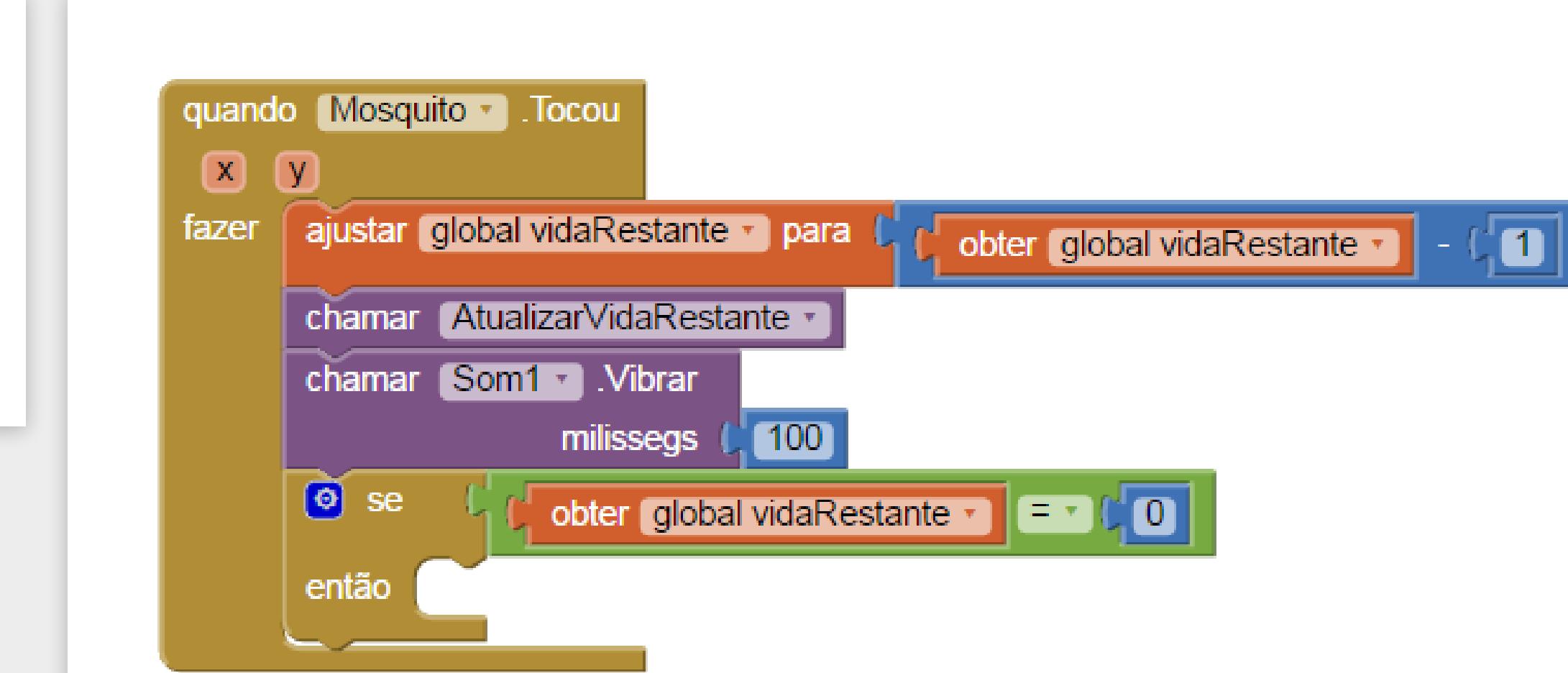


FINALIZAR O JOGO

3. Precisamos parar de diminuir o numero de vidas do mosquito

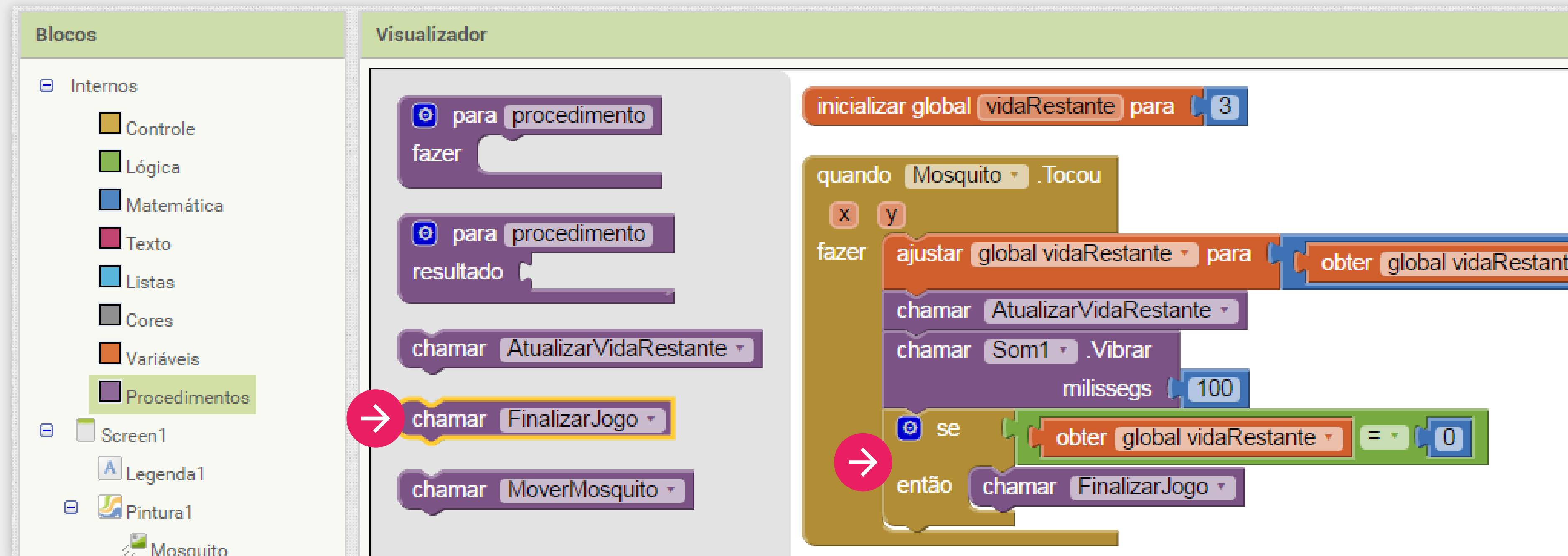
a) Lembra que toda a lógica de verificar e diminuir vida estava no procedimento “quando Mosquito.tocou” ?

b) Para parar de diminuir as vidas é só desativar o mosquito, assim ele não vai mais fazer ação nenhuma quando for tocado.



FINALIZAR O JOGO

Agora que nosso procedimento para **FinalizarJogo** está pronto só precisamos chamá-lo quando a **vidaRestante** for igual a zero, ou seja, dentro do bloco “então”.

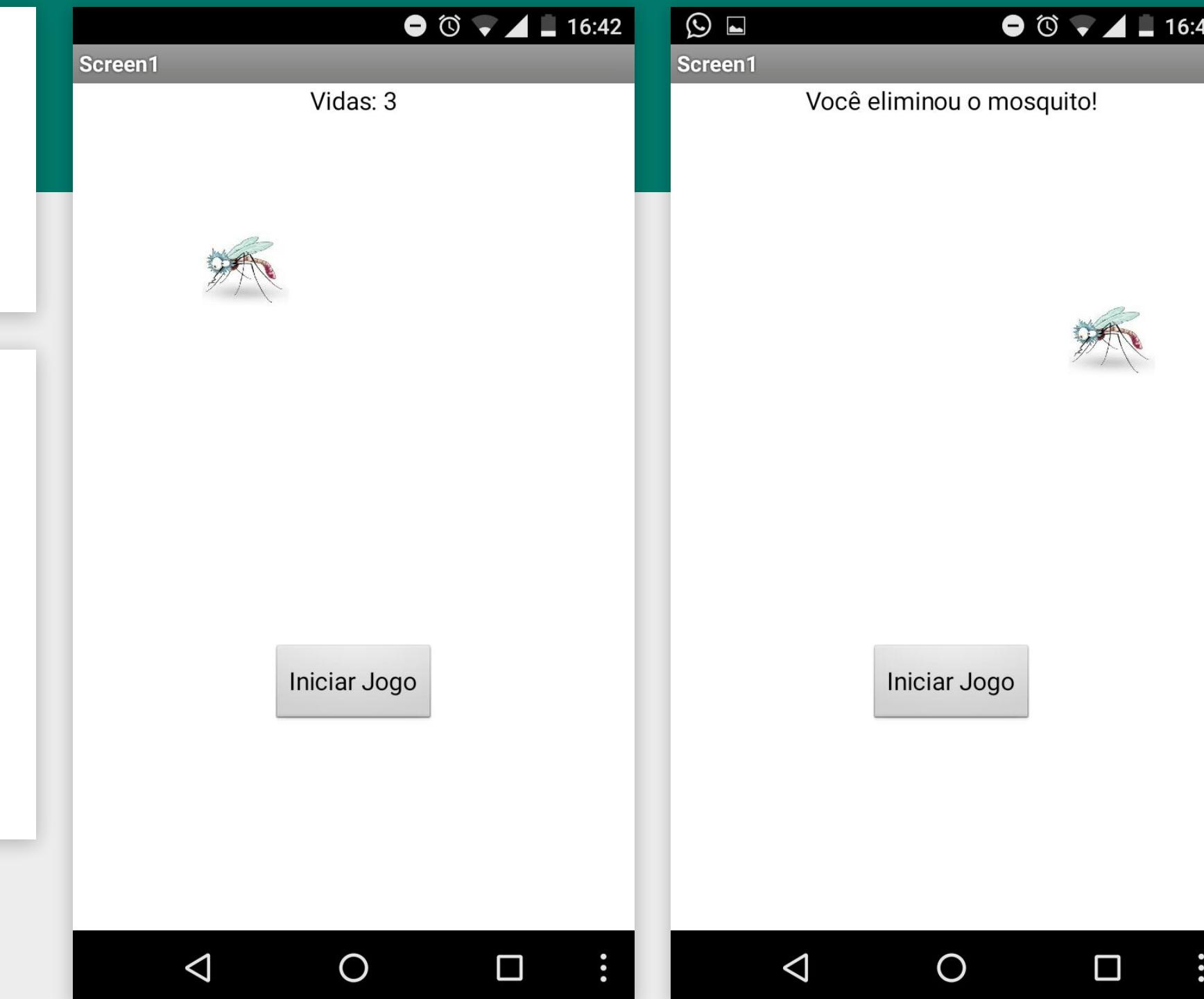


FINALIZAR O JOGO

AGORA VAMOS TESTAR!

Veja no seu celular se o jogo está finalizando corretamente.

Você verificou que ao finalizar o jogo e depois recomeçar clicando no botão Iniciar Jogo o texto da **Legenda1** ainda está mostrando que o mosquito foi eliminado mesmo ele estando vivo? O mosquito não está mais perdendo vidas? Temos bastante coisas para consertar!



REINICIAR O JOGO



Como no procedimento FinalizarJogo estamos trocando o texto da Legenda1, devemos trocá-lo novamente ao Iniciar o Jogo.

Como o nosso jogo é iniciado com o clique do “BotãoIniciar” vamos alterar o seu conteúdo e fazer com que ele:

1. Inicie o valor da variável “vidaRestante” para “3” novamente.
2. Atualize o texto da “Legenda1” para o texto “Vida Restante: 3”.
3. E ativar o mosquito para ele perder vida ao ser tocado.

REINICIANDO O JOGO

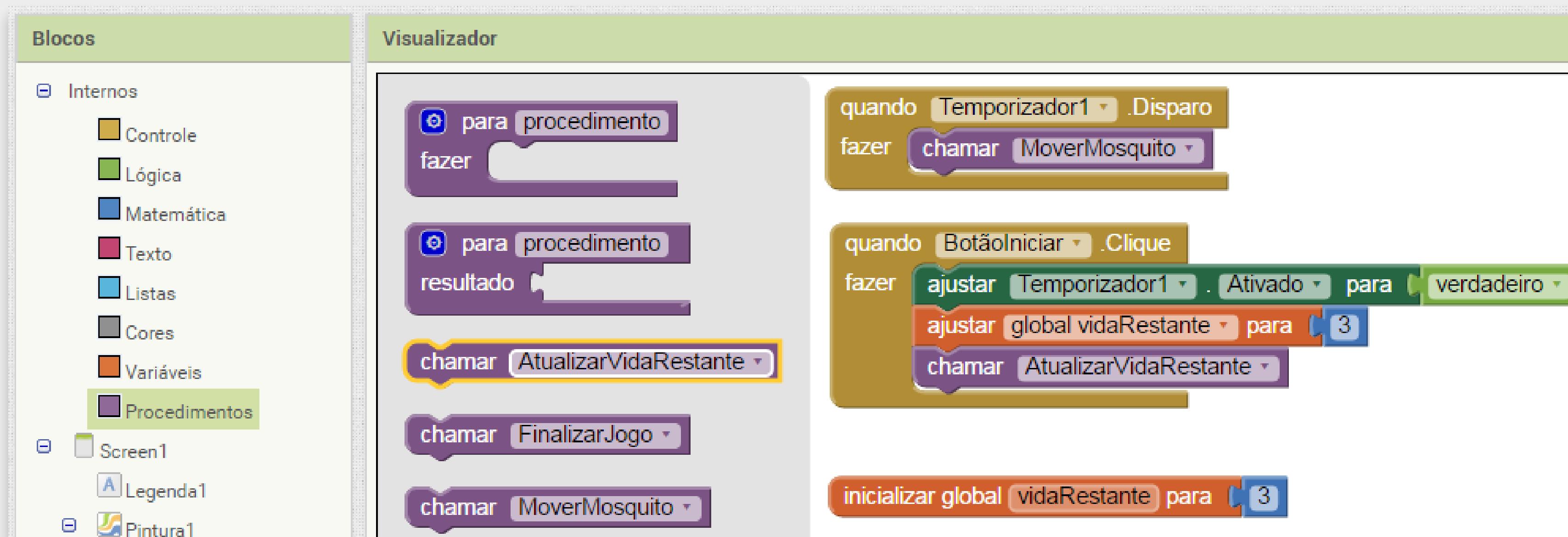
1. Inicie o valor da variável “vidaRestante” para “3” novamente.

a) Para alterar o valor da variável “vidaRestante” vamos utilizar o bloco “ajustar vidaRestante para” e informar o valor “3”.



REINICIANDO O JOGO

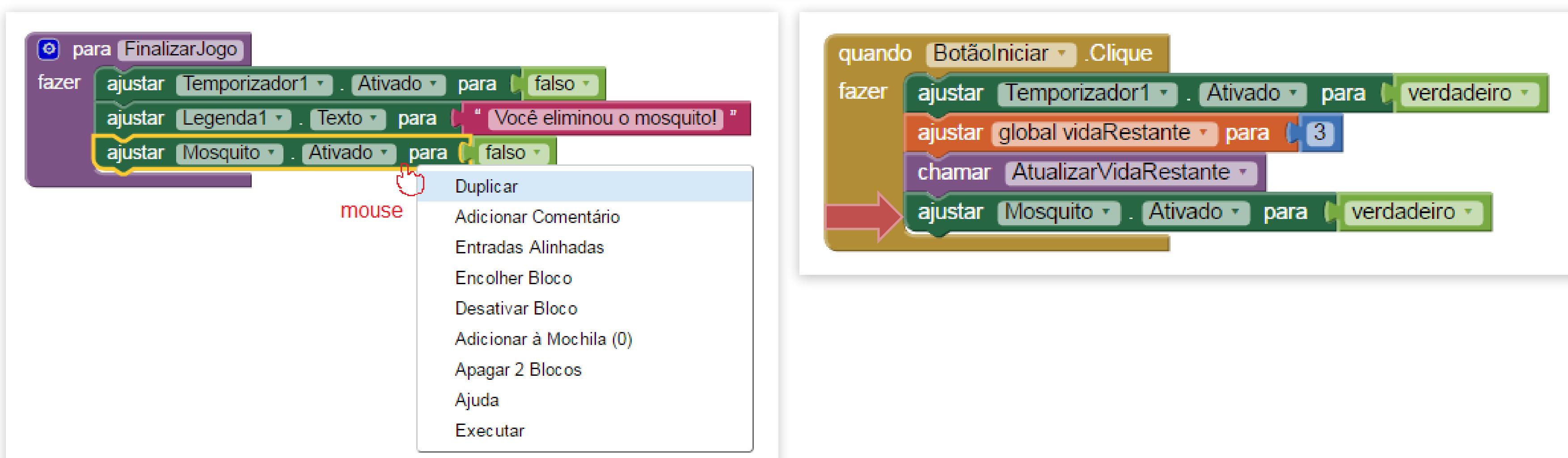
2. Atualize o texto da “Legenda1” para o texto “Vida Restante: 3”
Para isso só precisamos chamar o procedimento “AtualizarVidaRestante”.



REINICIANDO O JOGO

3. Reativar o mosquito para ele perder vida ao ser tocado

- Para ativar o mosquito, podemos usar o bloco “ajustar Mosquito.Ativado” com o valor “Verdadeiro”.
- A maneira mais fácil de fazer isso é clicar com o botão direito no bloco “ajustar Mosquito.Ativado” que já está dentro do procedimento “FinalizarJogo”, e escolher “Duplicar”.
- Depois disso é só mudar o valor “falso” para “verdadeiro”. E arrastar o novo bloco para o procedimento “quando BotãoIniciar.Clique”





ADICIONANDO TEMPO AO JOGO

Agora o jogo deve estar funcionando perfeitamente!

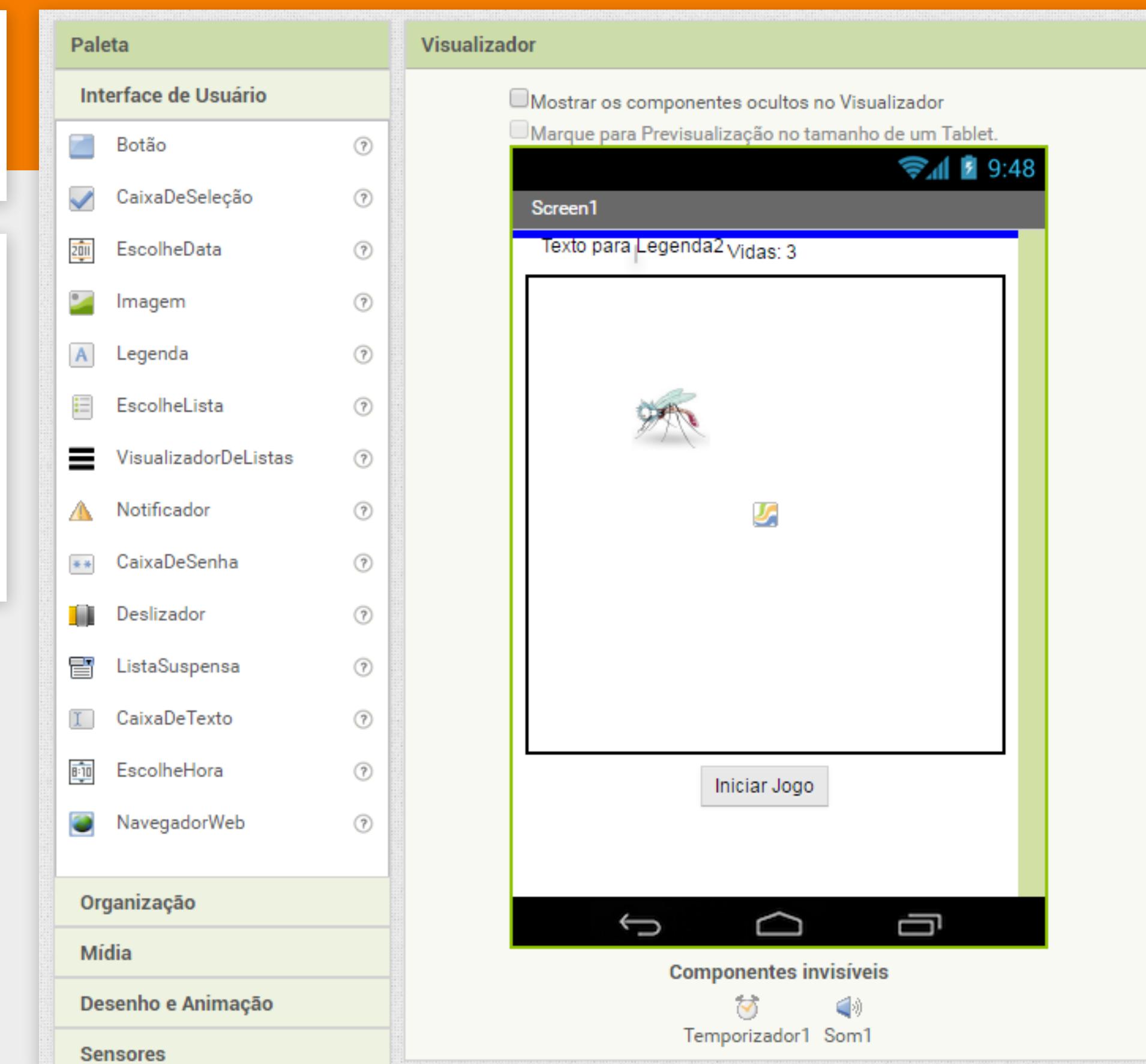
Você pode notar que o jogador ainda não possui nenhuma maneira de ser derrotado.

Para isto ocorrer, vamos fornecer um tempo máximo para o jogador eliminar o mosquito: caso o tempo acabe antes do mosquito perder todas as vidas, o jogador perde o jogo.

ADICIONANDO TEMPO

Volte para a tela de “Designer”.

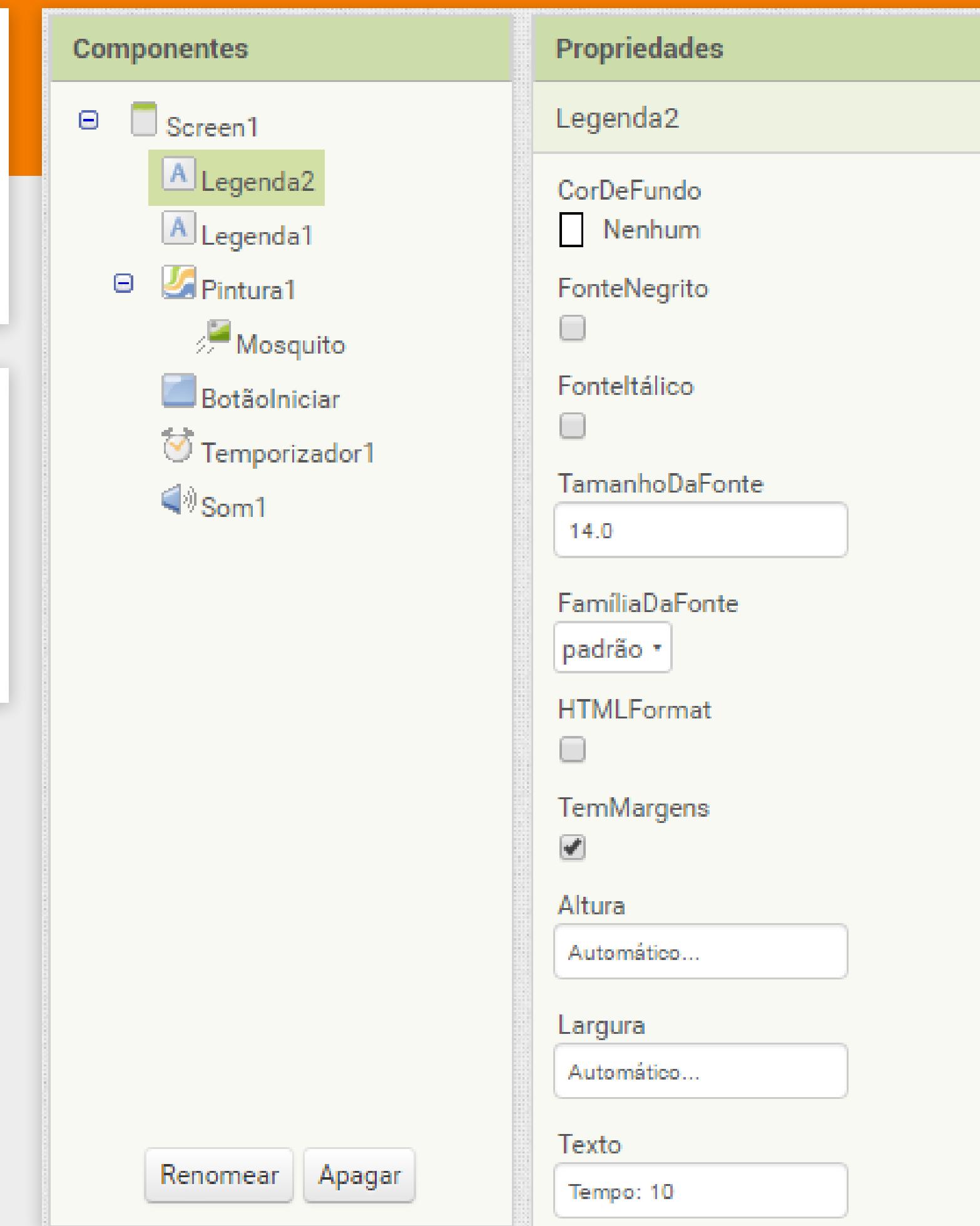
Vamos adicionar um novo componente Legenda para mostrar o tempo restante, preferencialmente acima da **Legenda1** (da vida).



ADICIONANDO TEMPO

Vamos alterar o texto da “Legenda2” para mostrar quanto tempo o jogador ainda tem para eliminar o mosquito.

Para fazer isto é necessário selecionar o componente “Legenda2” e no campo Texto digitar “Tempo: 10”

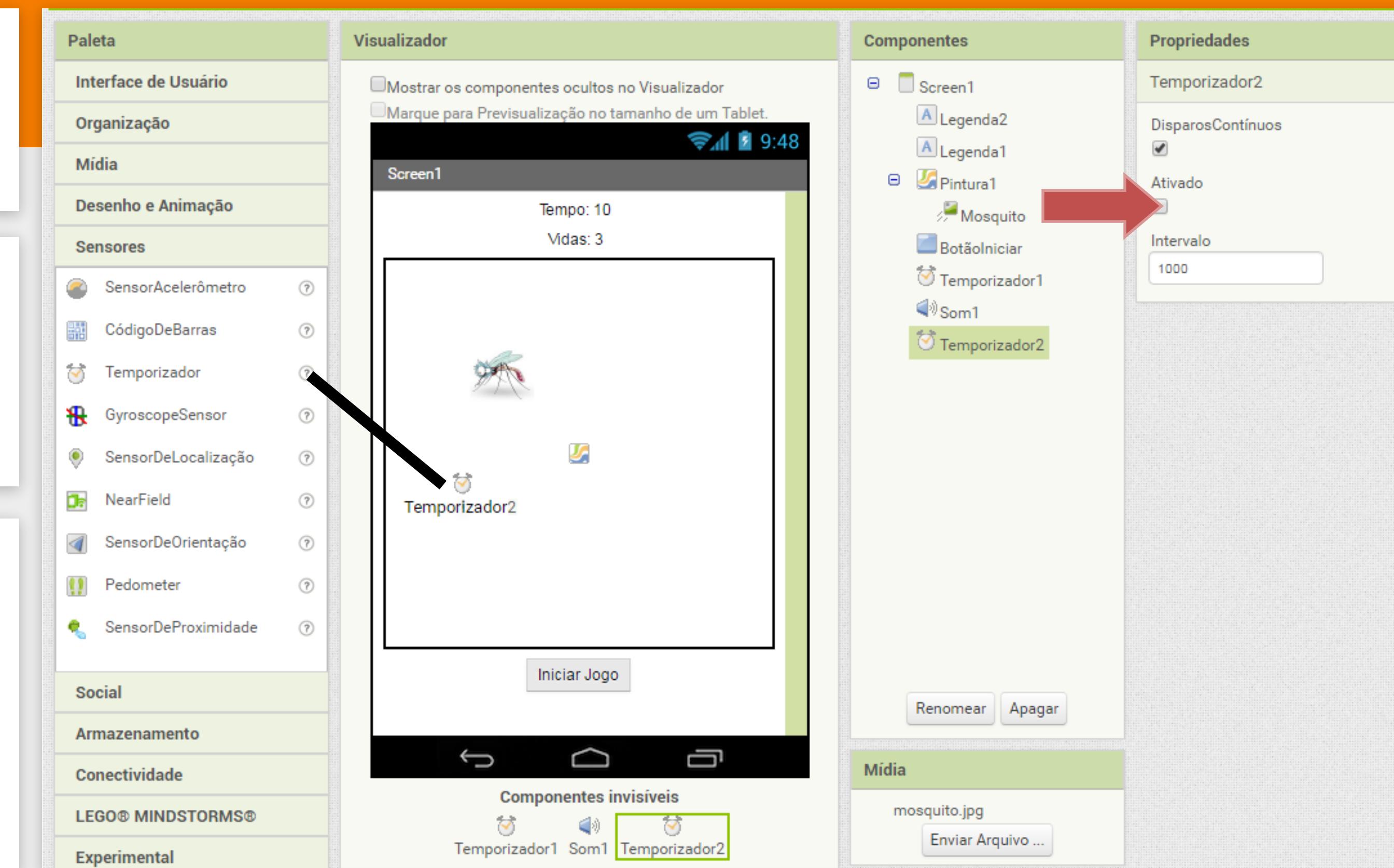


ADICIONANDO TEMPO

Para contar o tempo vamos utilizar um novo Temporizador.

Selecione a paleta Sensores e arraste o componente Temporizador até a tela.

Configure o **Temporizador2** para realizar disparos a cada 1 segundo (1000 millisegundos), vamos deixá-lo desativado por enquanto.



ADICIONANDO TEMPO

Voltamos a tela de “Blocos” para adaptar o nosso jogo.



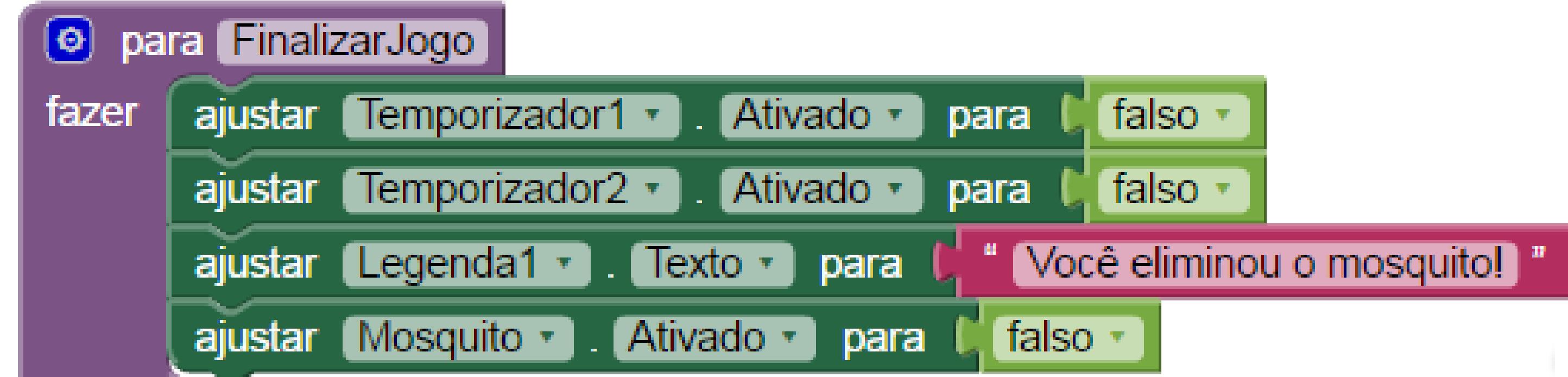
Primeiramente, vamos ativar o nosso **Temporizador2** quando o jogo for iniciado, incluindo o bloco **ajustar Temporizador2.Aтивado** com valor Verdadeiro dentro do procedimento quando **Botaoiniciar.Clique**.

DICA

Você pode Duplicar o bloco “ajustar Temporizador1.Aктивado” como já ensinamos anteriormente. Lembre-se de mudar o componente para “Temporizador2”.

ADICIONANDO TEMPO

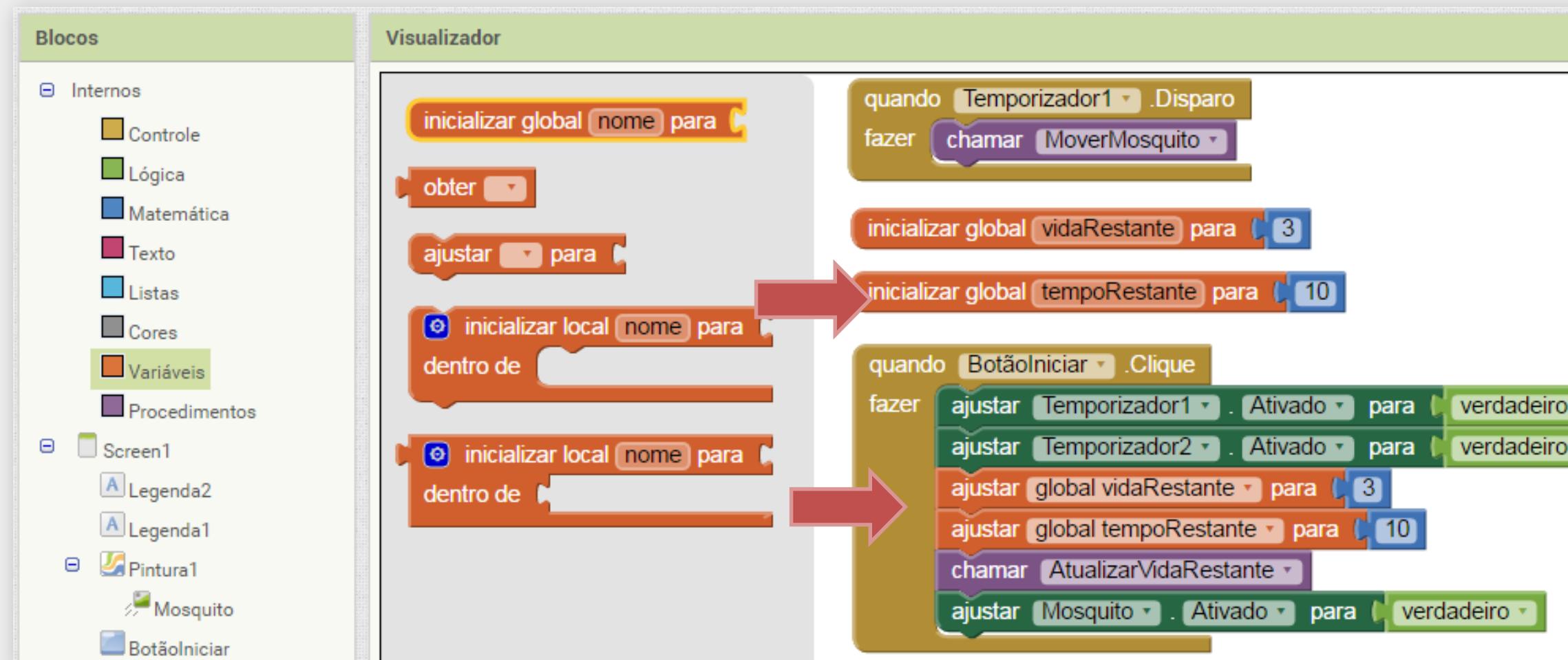
Também precisamos lembrar de desativar o **Temporizador2** quando o jogo for finalizado.



Inclua o mesmo bloco com valor Falso no procedimento **FinalizarJogo**.

ADICIONANDO TEMPO

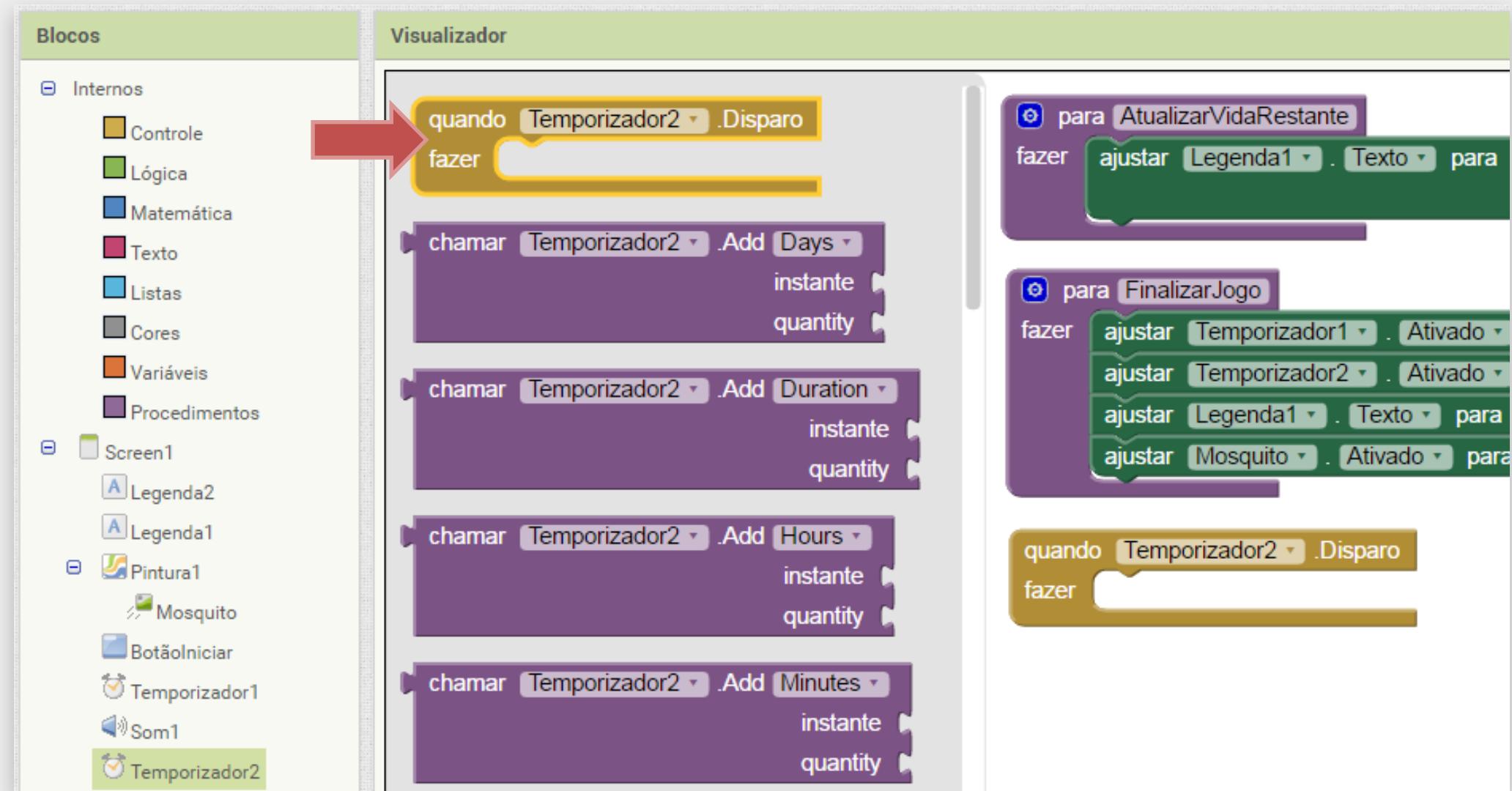
Agora precisamos criar uma variável global, responsável por armazenar o tempo restante de jogo. Vamos iniciá-la com o valor "10".



Também devemos ajustar o seu valor para “10” no procedimento de iniciar o jogo (BotãoIniciar.Clique).

ADICIONANDO TEMPO

Precisamos atualizar o tempo restante a cada segundo, ou seja, toda vez que o “Temporizador2” disparar.



Para isso, vamos adicionar no Visualizador o bloco “quando Temporizador2.Disparo”.

ADICIONANDO TEMPO

Dentro do **Temporizador2.Disparo** vamos diminuir o valor da variável “tempoRestante” em “1”.



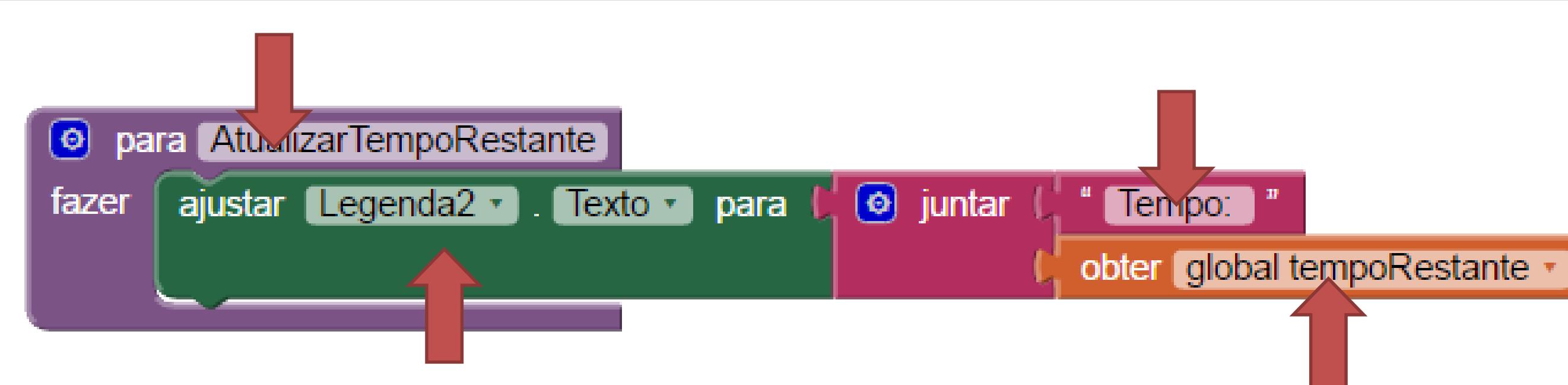
Usamos o bloco “ajustar global tempoRestante”, os blocos Matemáticos de Subtração e número inteiro “1” e o bloco “obter globalRestante”.

ADICIONANDO TEMPO

O valor do **tempoRestante** está sendo diminuído, porém o jogador ainda não está percebendo isto. Precisamos atualizar o texto da **Legenda2** que é mostrado na tela.

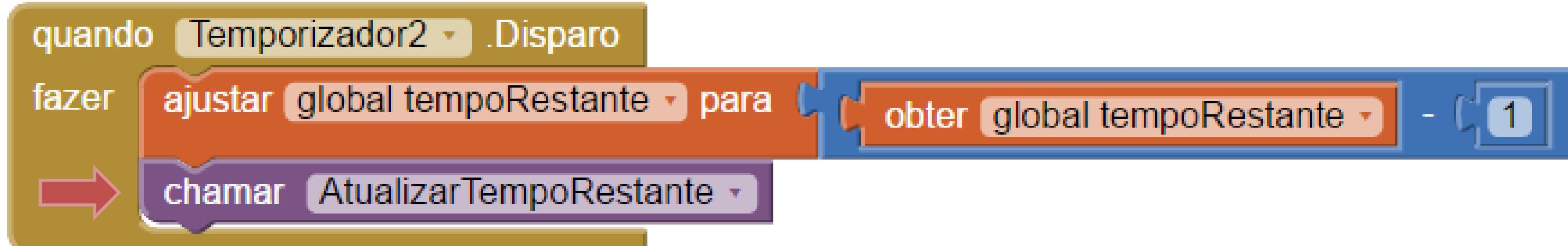
Para isto, vamos criar um procedimento “AtualizarTempoRestante”, similar ao procedimento “AtualizarVidaRestante”, responsável por trocar o texto da **Legenda2**.

Lembre-se que você pode duplicar o procedimento “AtualizarVidaRestante”, e apenas mudar os parâmetros, para ficar conforme o bloco abaixo.



ADICIONANDO TEMPO

Temos que chamar o procedimento para atualizar a tela logo após diminuir o valor “tempoRestante”, dentro do bloco “Temporizador2.Disparo”.



ADICIONANDO TEMPO

Também precisamos chamar o procedimento “AtualizarTempoRestante” dentro do bloco “quando BotãoIniciar.Clique”, para mostrar o tempo correto quando o jogo iniciar.

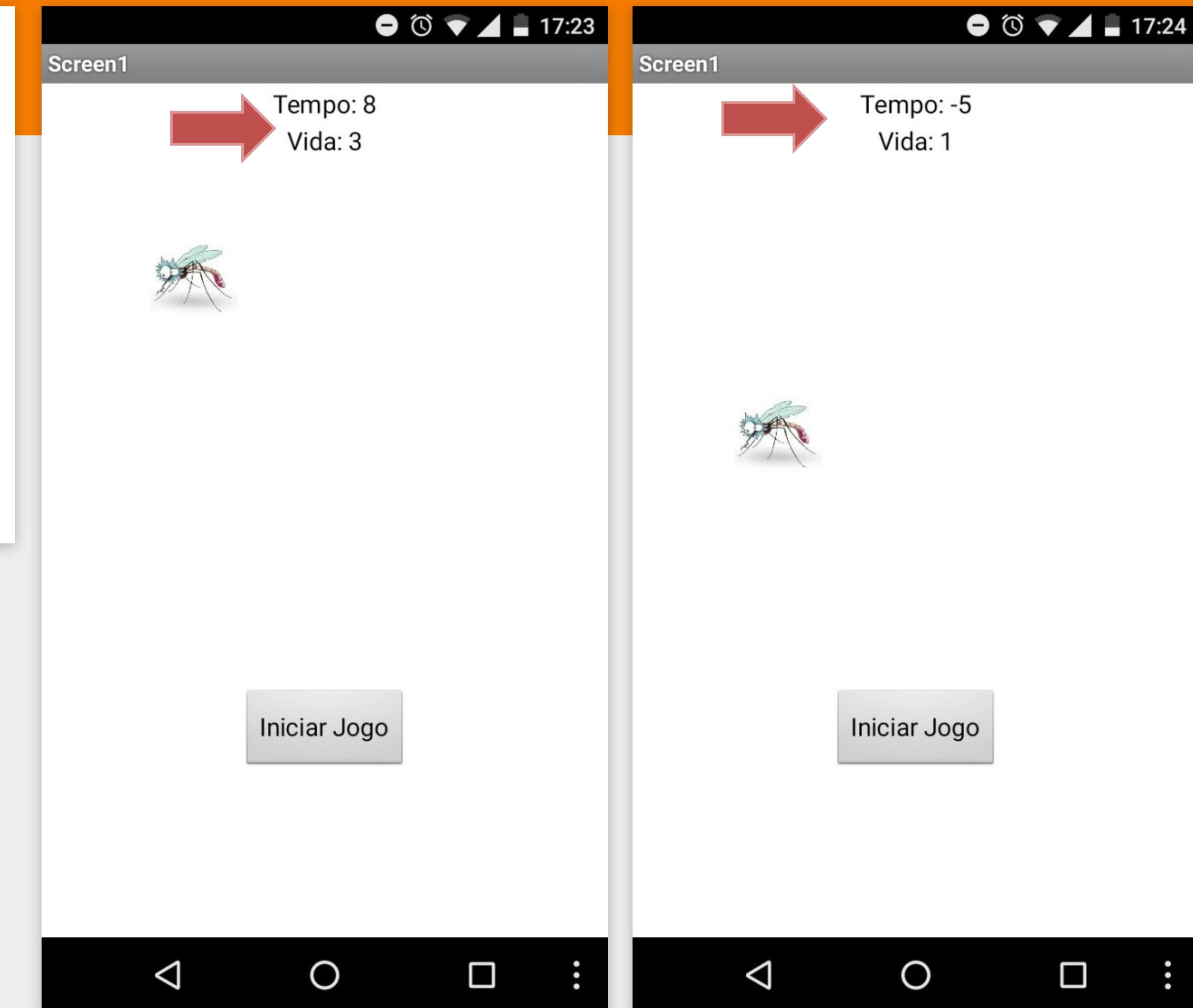


ADICIONANDO TEMPO

VAMOS TESTAR!

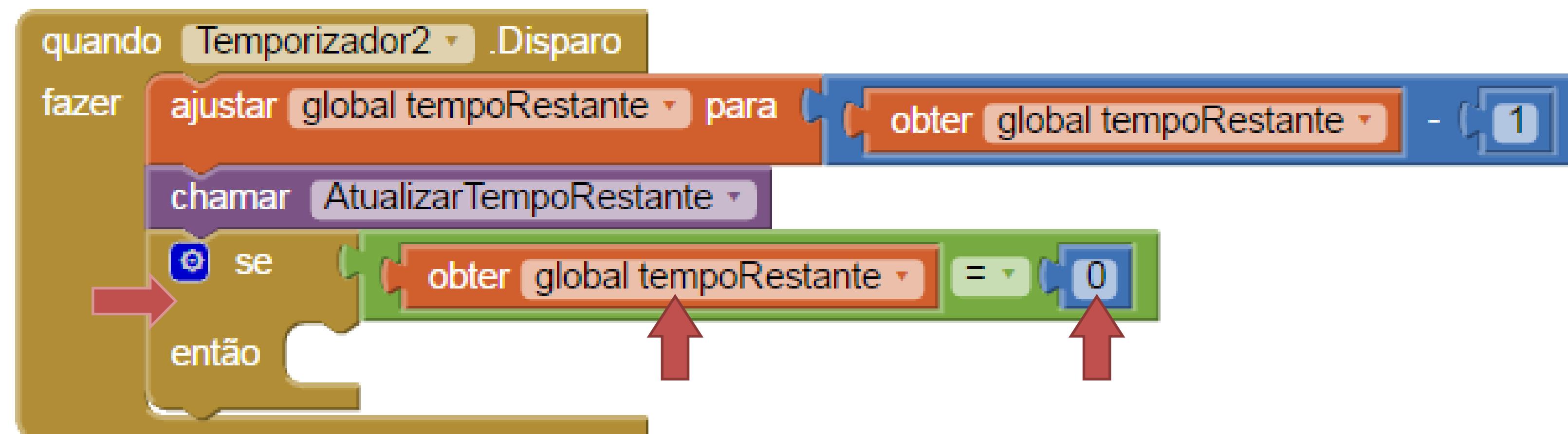
Verifique se o tempo está diminuindo na tela e se está se comportando corretamente.

Você deve ter percebido que o jogo ainda não está terminando quando o valor do “tempoRestante” chega a “0” (zero).



ADICIONANDO TEMPO

Sendo assim, vamos começar alterando o procedimento “Temporizador2.Disparo” adicionando o bloco de controle “Se, então” e utilizar a operação lógica “tempoRestante = 0” para terminar o jogo.



ADICIONANDO TEMPO

Devemos finalizar o jogo caso o tempo tenha esgotado, então vamos chamar o procedimento [FinalizarJogo](#).

Teste o se o jogo está finalizando corretamente quando o tempo acaba! O jogador deve ser derrotado.

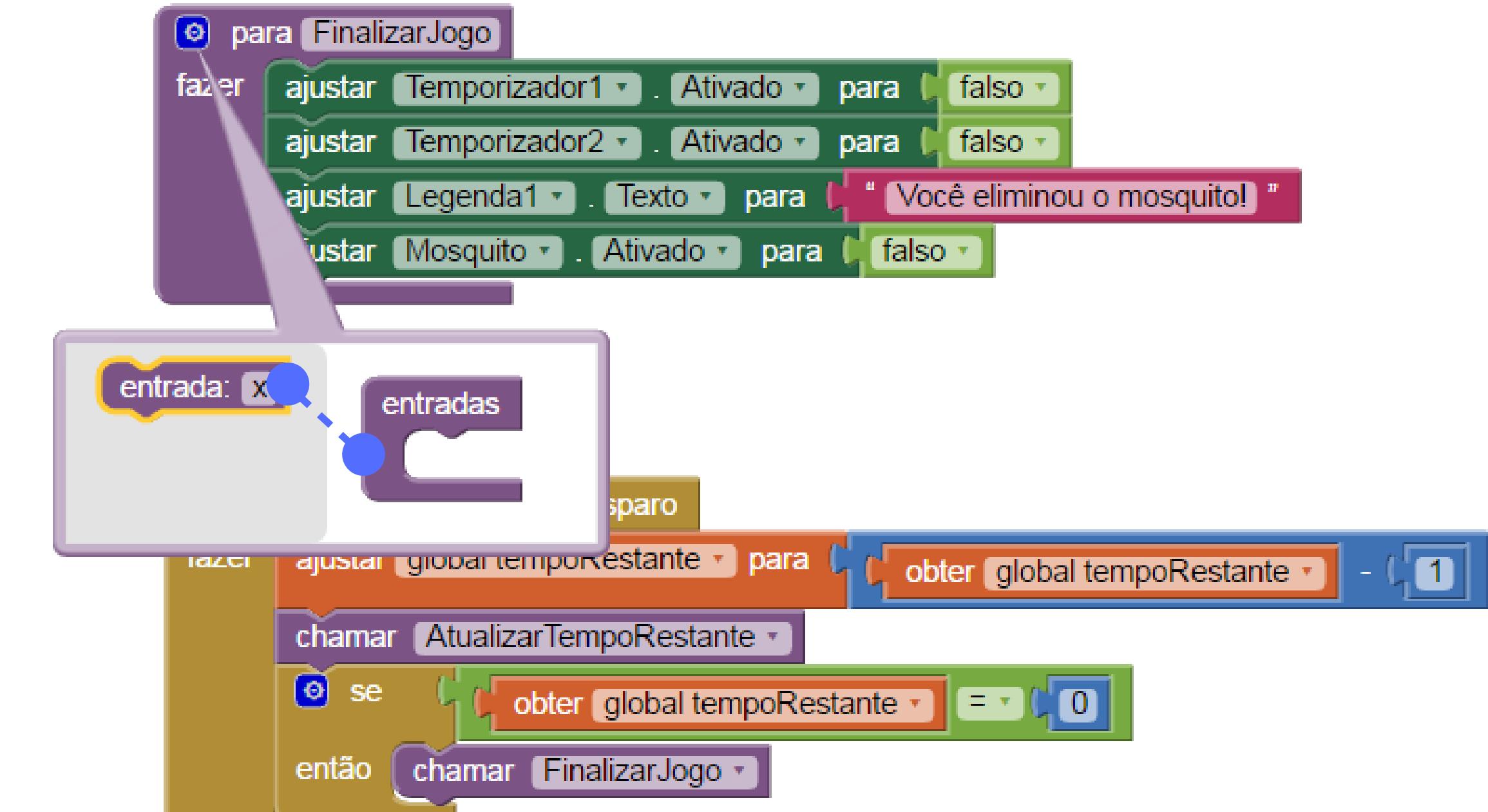
Você verificou que apesar do tempo ter se esgotado ainda apareceu que o jogador conseguiu eliminar o mosquito?



ADICIONANDO TEMPO

Temos que modificar o procedimento “FinalizarJogo” para receber a informação se o jogador ganhou ou perdeu, e mudar o texto da **Legenda1** corretamente.

Para isto, clique no ícone azul no canto superior do procedimento “FinalizarJogo” e arraste o bloco “entrada” para encaixar no bloco “entradas” e troque o texto de “x” para “resultadoFinal”.



ADICIONANDO TEMPO

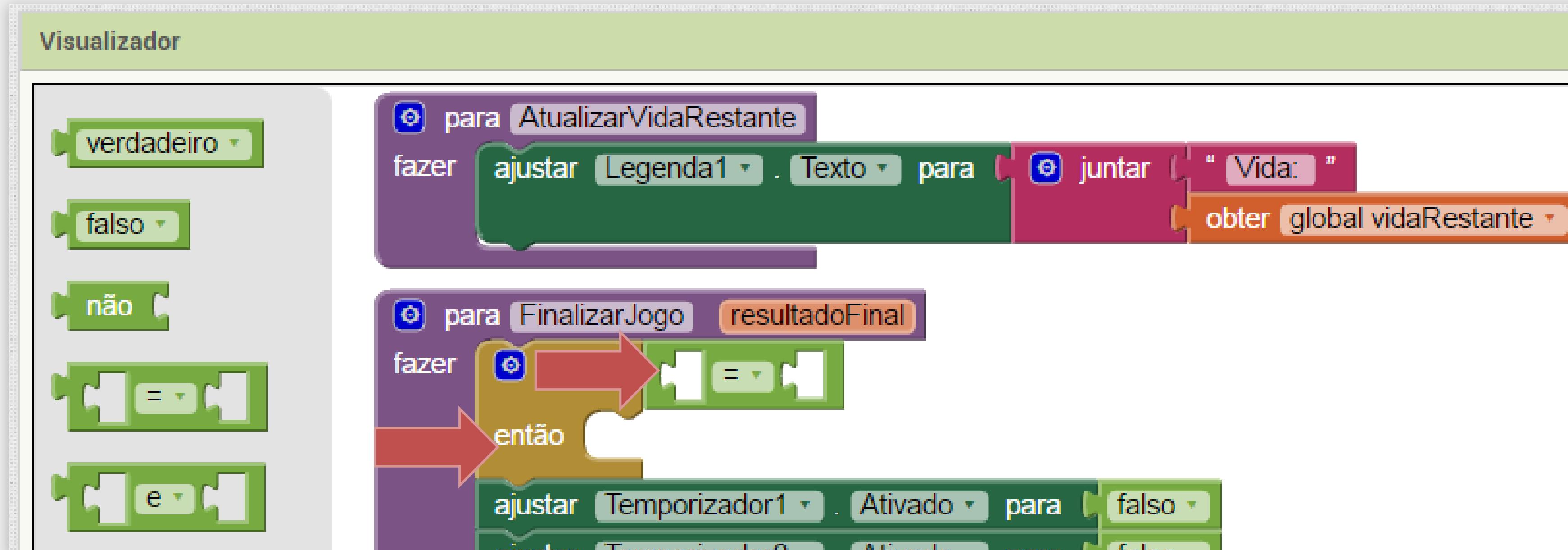
DEVE FICAR ASSIM:



ADICIONANDO TEMPO

Agora precisamos verificar o valor da variável local “resultadoFinal”. Caso esse resultado seja Verdadeiro o jogador foi o vencedor, caso contrário (senão) ele perdeu.

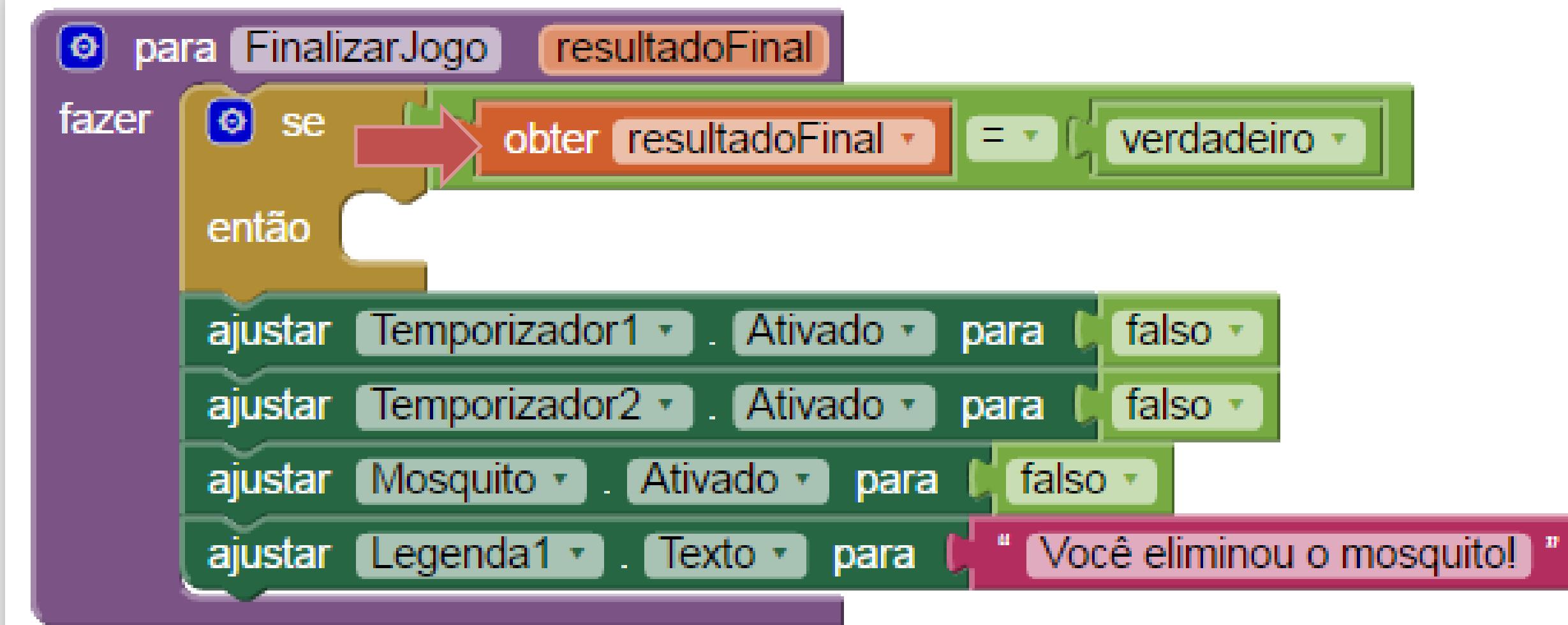
Primeiro adicione o bloco “se então” ao “FinalizarJogo”, e coloque uma bloco Lógico de comparação (=).



ADICIONANDO TEMPO

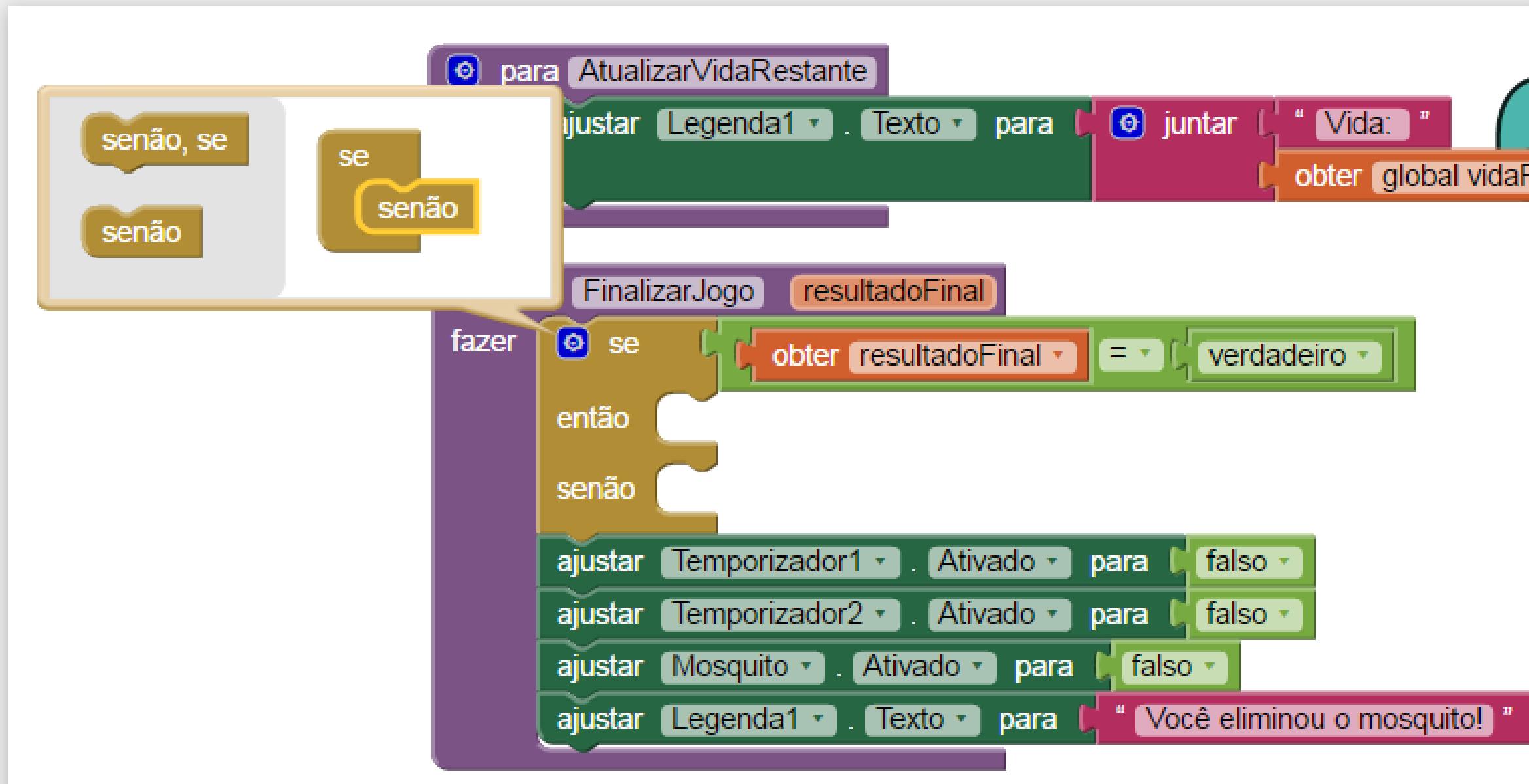
Adicione na comparação a variável **resultadoFinal** e o valor Verdadeiro.

A variável local é obtida da mesma maneira que as variáveis globais, mas ela só pode ser acessada dentro do procedimento **FinalizarJogo**.



ADICIONANDO TEMPO

Para mudar o bloco “Se então” para “Se então senão”, clique no ícone azul do bloco e arraste o bloco “senão” para dentro do “se”.



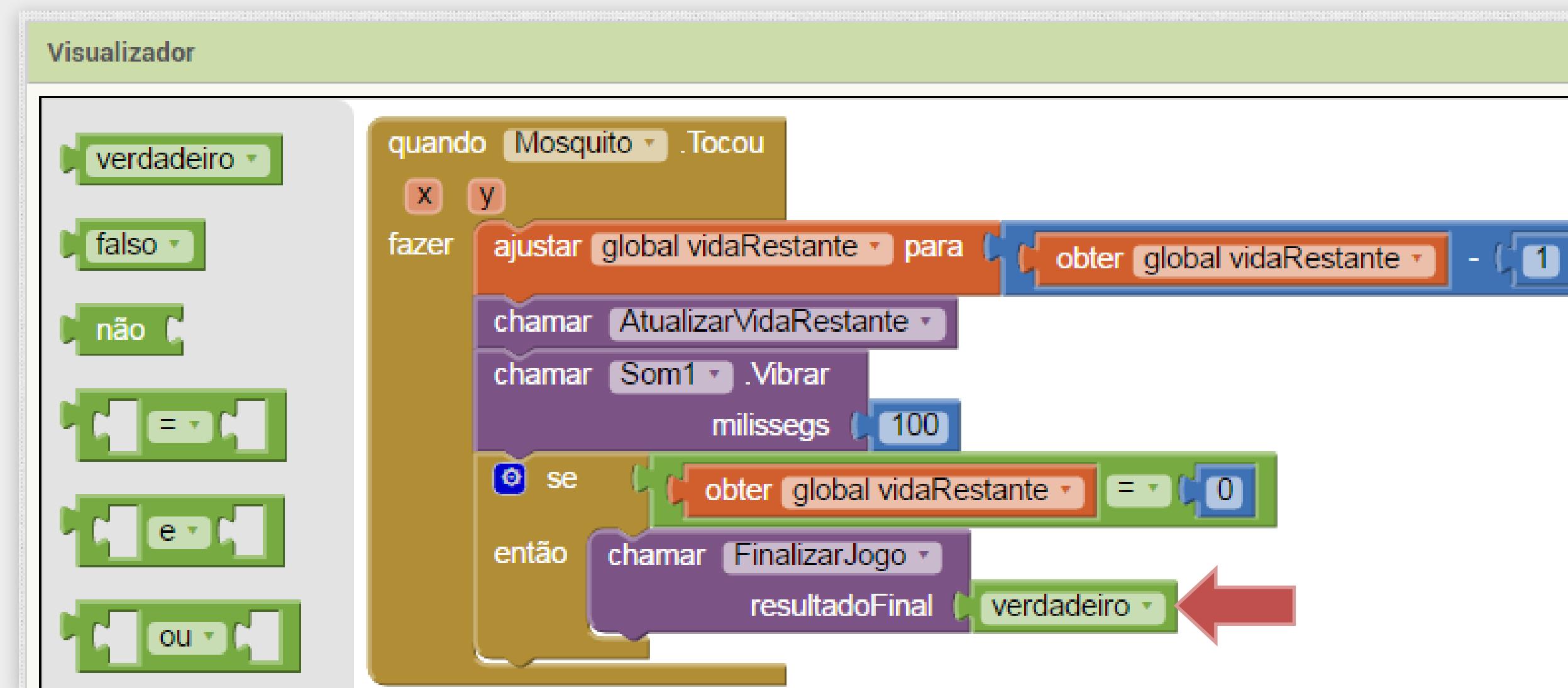
ADICIONANDO TEMPO

Quando o resultadoFinal for verdadeiro, vamos atualizar a **Legenda1** com o texto de vitória, senão vamos atualizar com o texto de derrota ("Você perdeu!").



ADICIONANDO TEMPO

Agora, sempre que chamamos o procedimento **FinalizarJogo** temos que passar o valor de **resultadoFinal** que queremos.



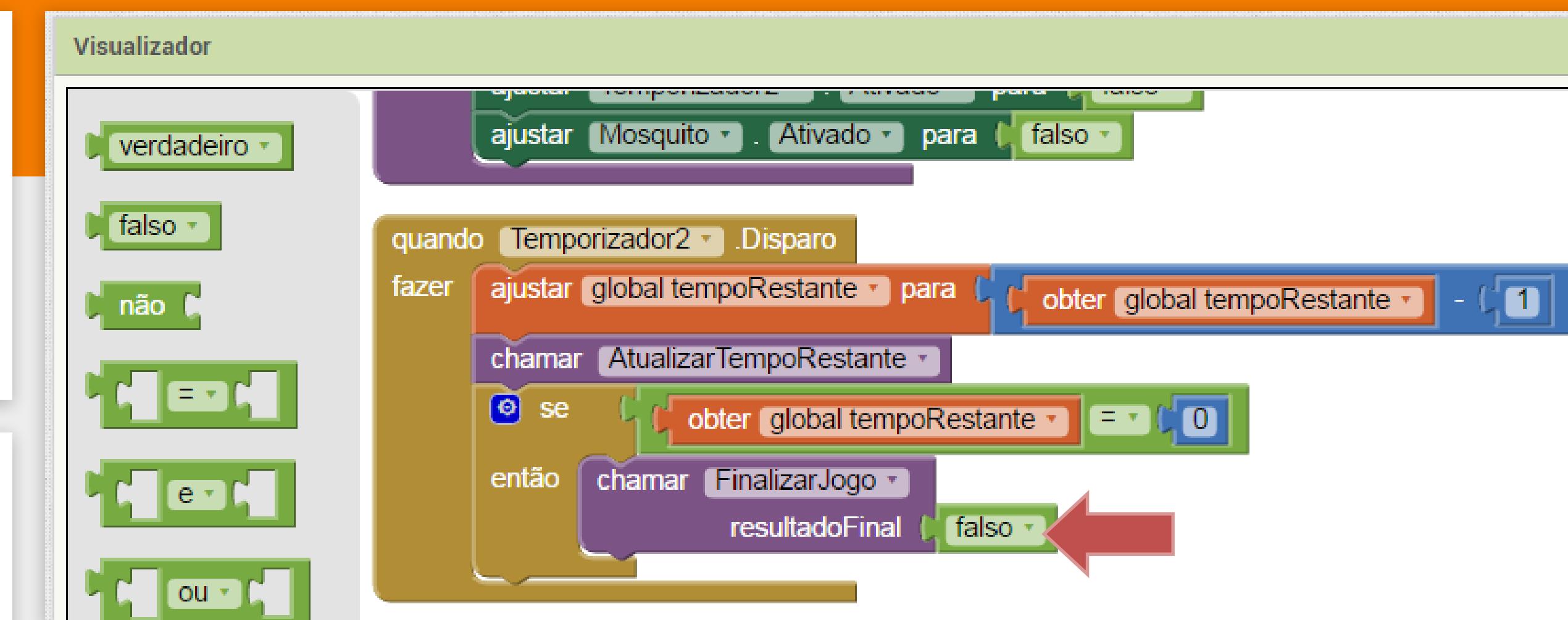
Vamos atualizar o bloco “Mosquito.Tocou” passando o valor “Verdadeiro” como parâmetro, pois neste procedimento o jogador foi vencedor.

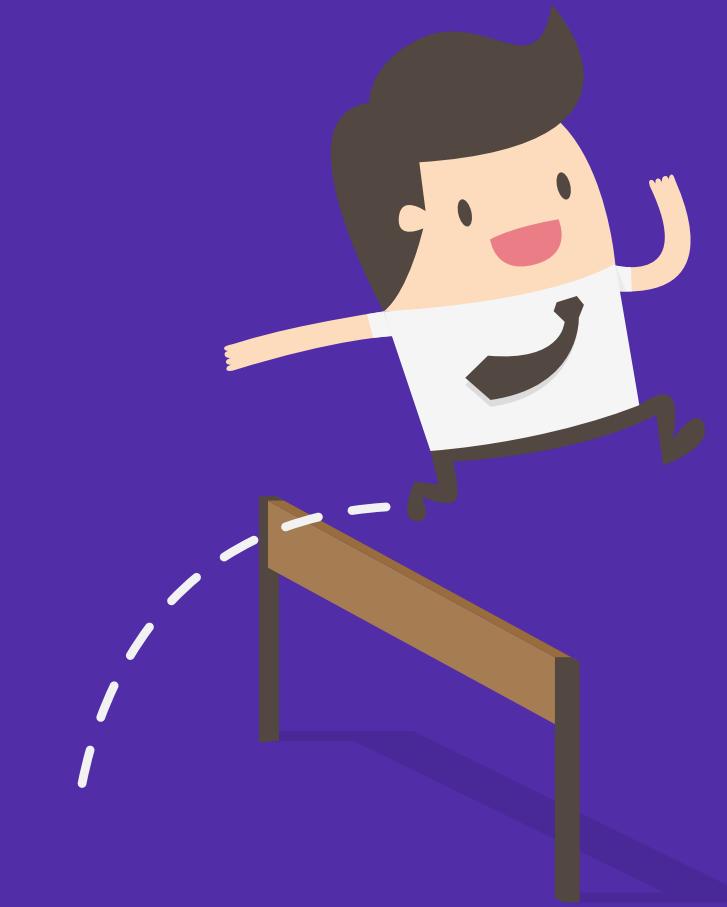
ADICIONANDO TEMPO

E também atualizar a chamada do procedimento **FinalizarJogo** no bloco “quando Temporizador2.Disparo” passando o valor Falso como parâmetro, pois neste procedimento o jogador perdeu.

Agora verifique se o seu jogo está funcionando perfeitamente com o tempo para matar o mosquito!

Assim conseguimos finalizar a funcionalidade de Adicionar Tempo para matar o mosquito.





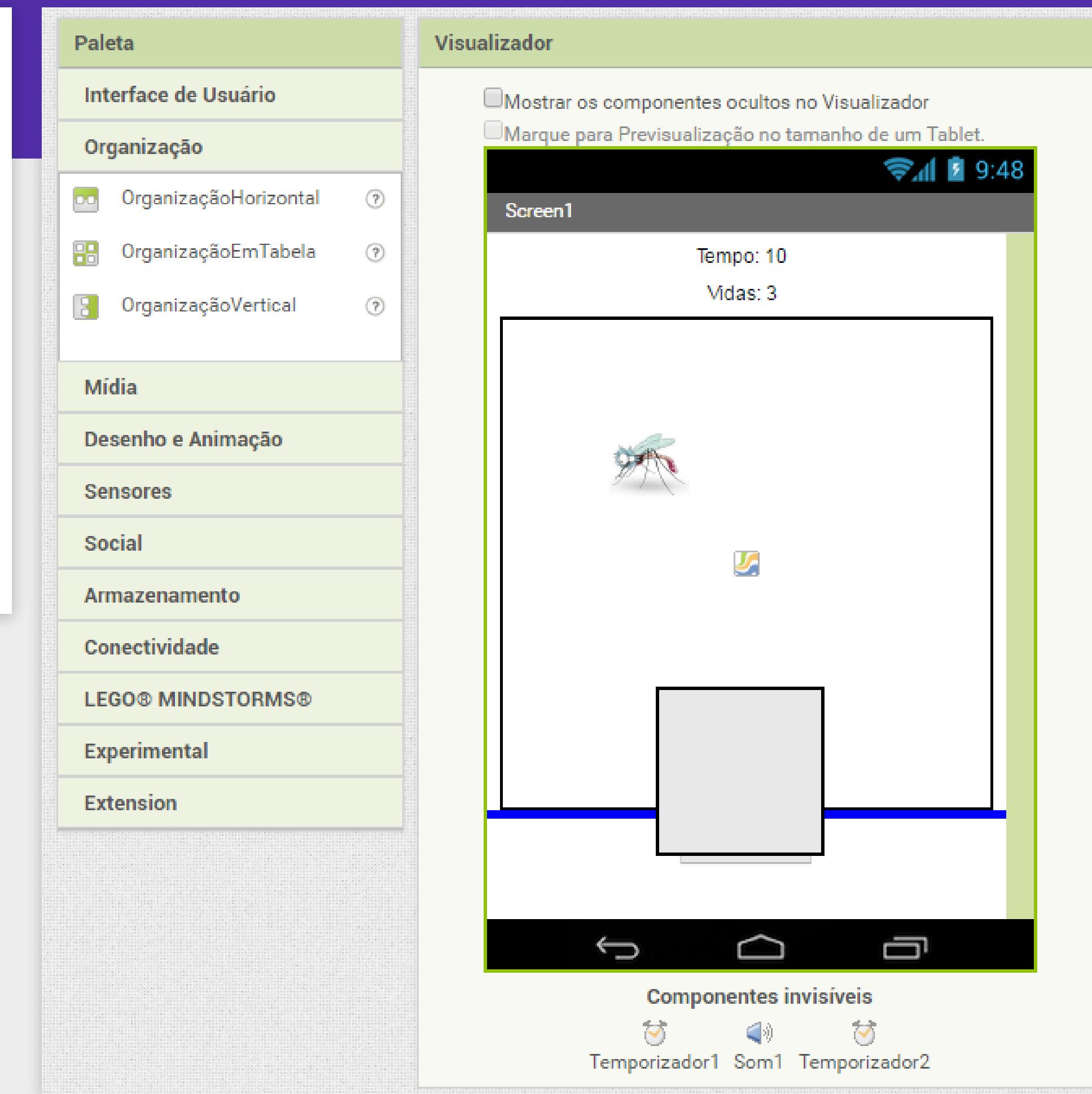
ADICIONAR DIFICULDADE

Para deixar o jogo mais divertido, vamos deixar o jogador escolher a dificuldade do jogo.

A variação de dificuldade vai atuar sobre a variável tempoRestante e a velocidade em que o mosquito irá se movimentar.

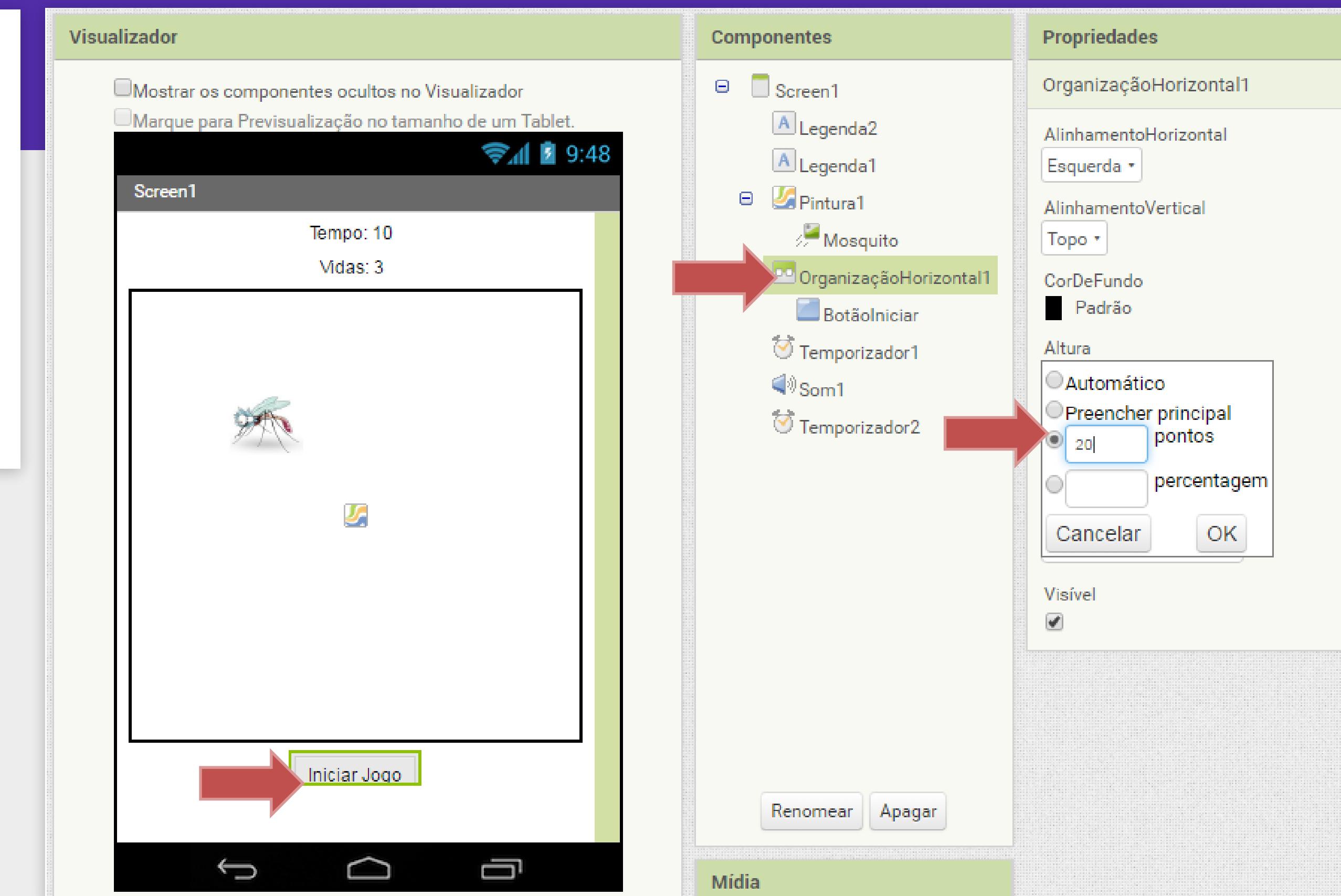
ADICIONAR DIFICULDADE

Para exibir na tela os níveis de dificuldade vamos adicionar o componente **ListaSuspensa** ao lado do **BotãoIniciar**. Para isto devemos ir ao “Designer” e selecionar a paleta Organização e arrastar o componente **OrganizaçãoHorizontal** para uma posição acima do **BotãoIniciar**.



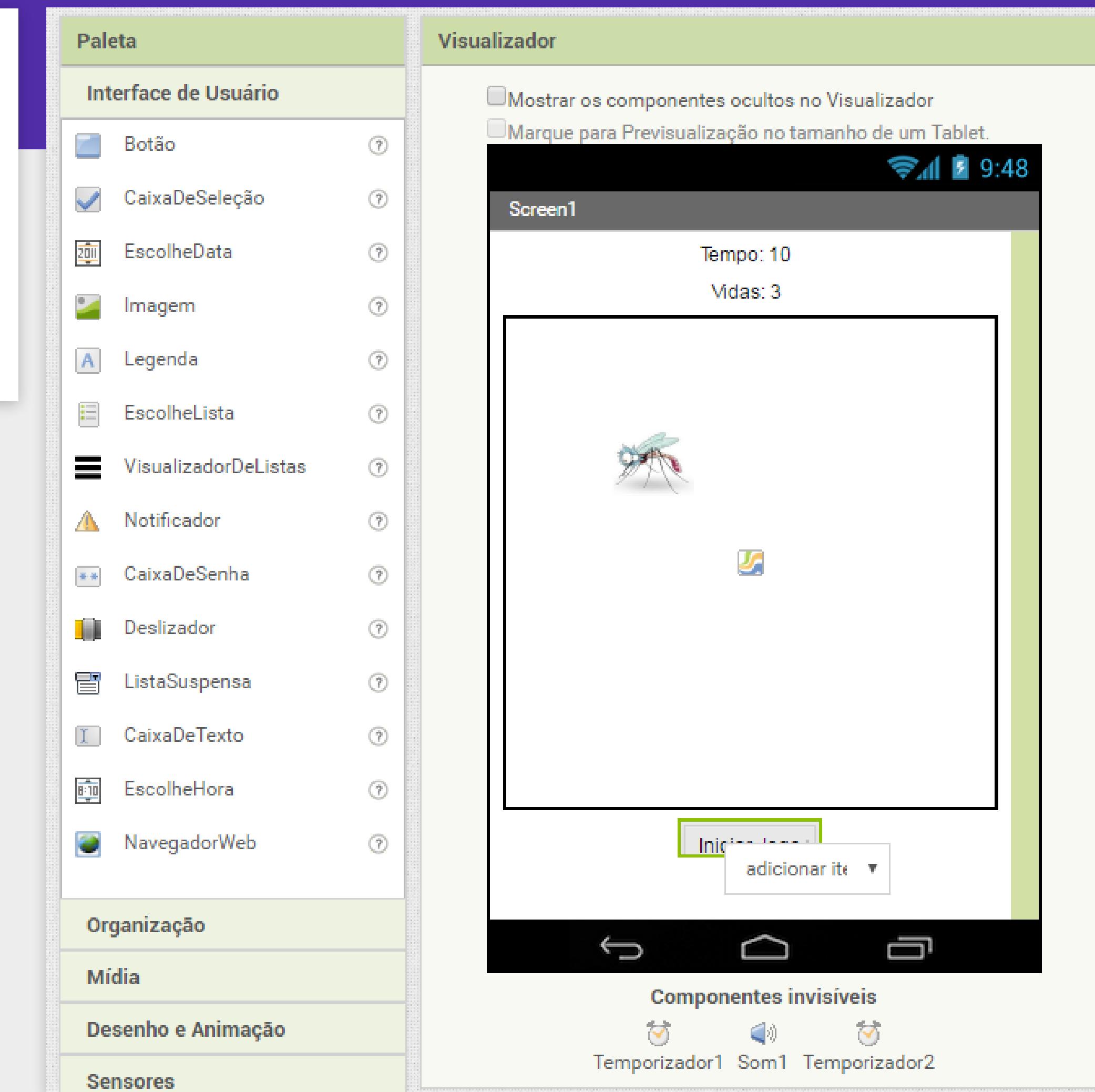
ADICIONAR DIFICULDADE

O componente **OrganizaçãoHorizontal1** acabou por colocar o **Botaoiniciar** para fora da tela, por isso temos que configurar a sua altura como “20”, e arrastar o **Botaoiniciar** para dentro da **OrganizacaoHorizontal1**.



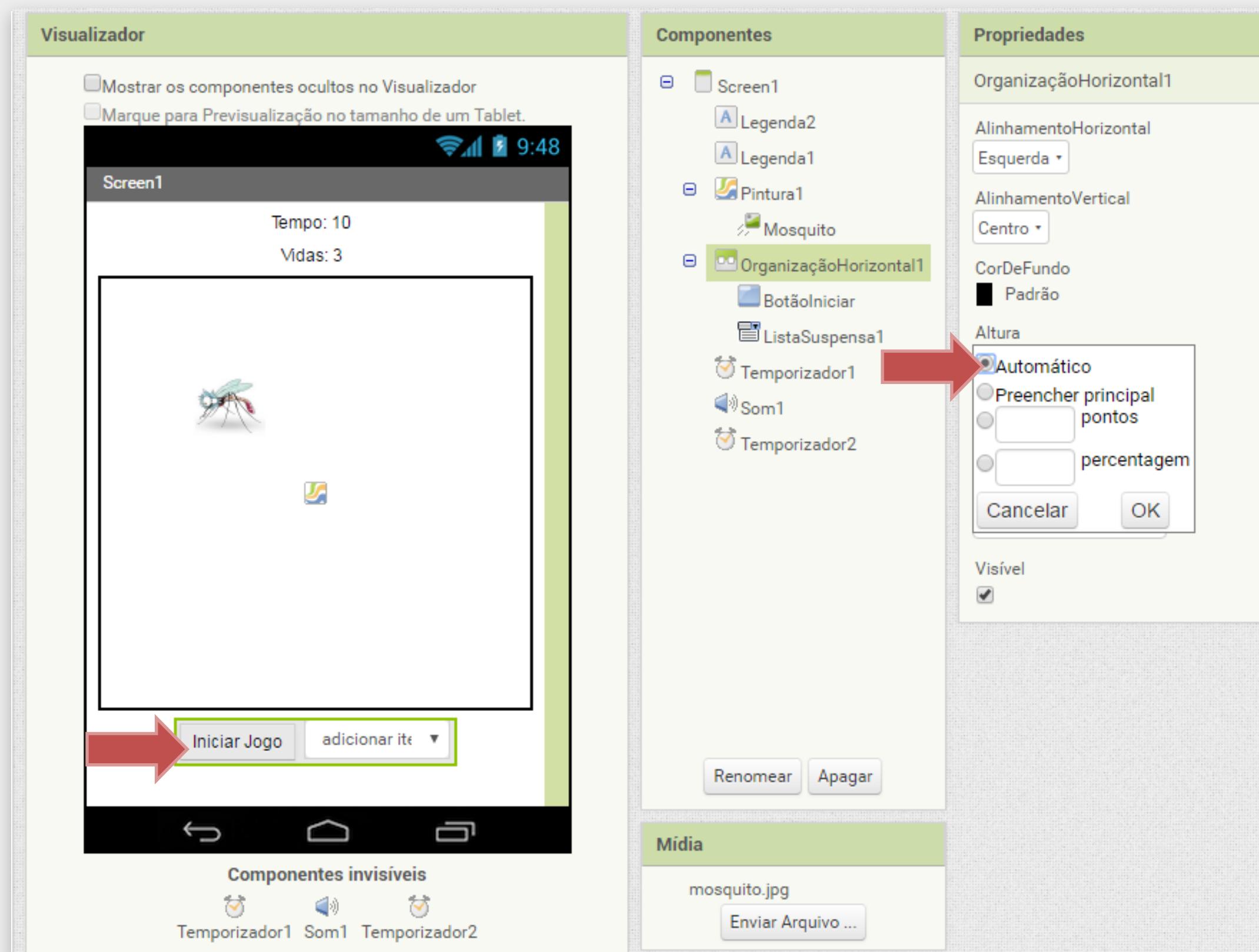
ADICIONAR DIFICULDADE

Agora devemos clicar na paleta Interface de Usuário e arrastar o componente **ListaSuspensa** para dentro da **OrganizaçãoHorizontal1** recém adicionada, ao lado do **BotãoIniciar**.



ADICIONAR DIFICULDADE

Após colocar a lista no lugar, podemos trocar a altura da **OrganizaçãoHorizontal1** para automática novamente, assim ela vai ficar no tamanho adequado para o botão e a lista.

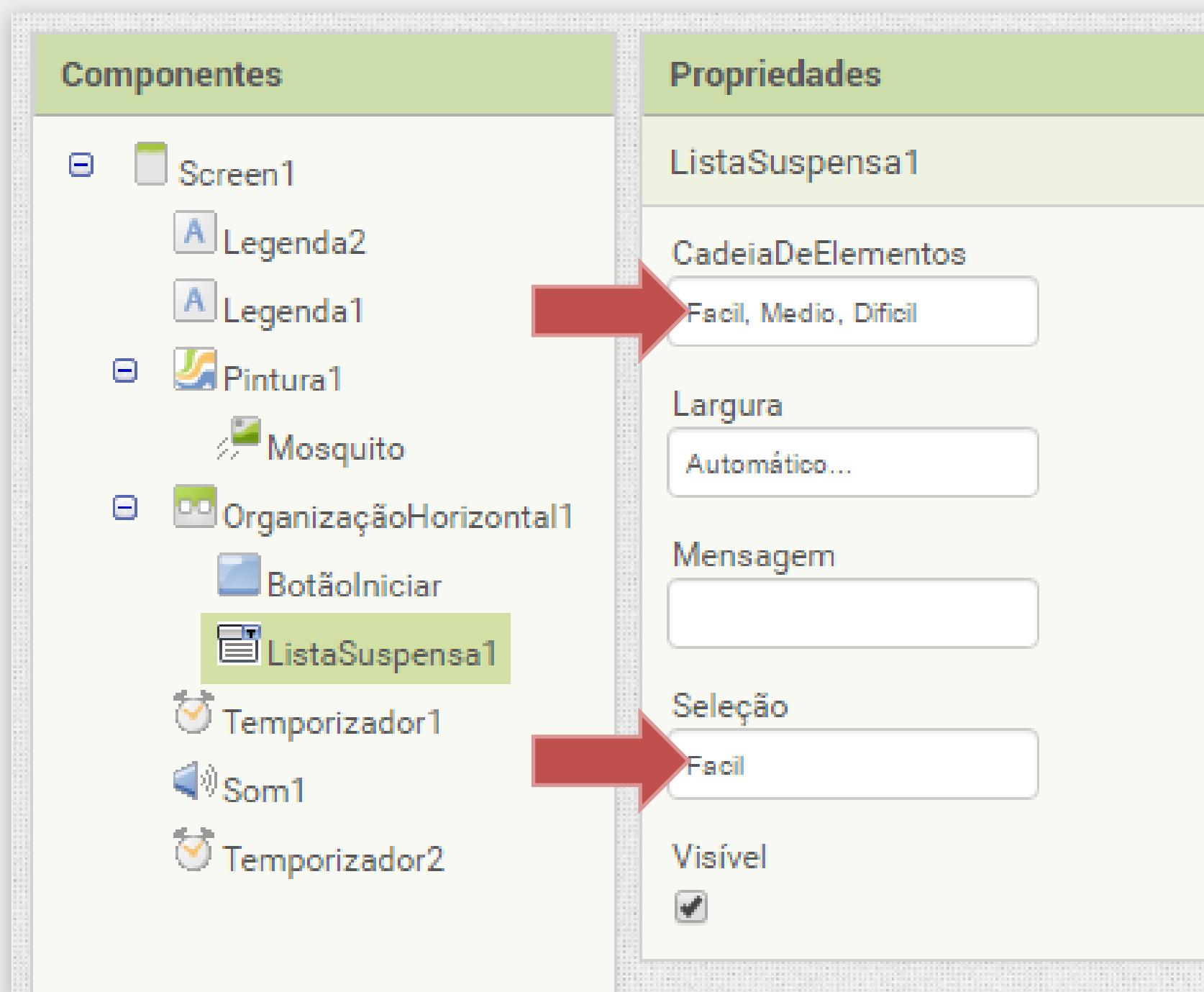


ADICIONAR DIFICULDADE

Agora devemos configurar o componente **ListaSuspensa1** para apresentar os valores “Facil, Medio e Difícil”. Para isto preencha os campos da seguinte maneira:

DICA

Para adicionar os elementos da lista coloque cada item separado por vírgula e sem espaço entre a “,” e o texto. Por exemplo: “Facil,Medio,Dificil”.



ADICIONAR DIFICULDADE

Após adicionar a **ListaSuspensa1** no bloco “Designer” precisamos voltar ao “Blocos” para configurar a sua funcionalidade.

Primeiro precisamos inicializar uma variável global chamada de **nivelSelecionado** com o valor inicial em “1”.

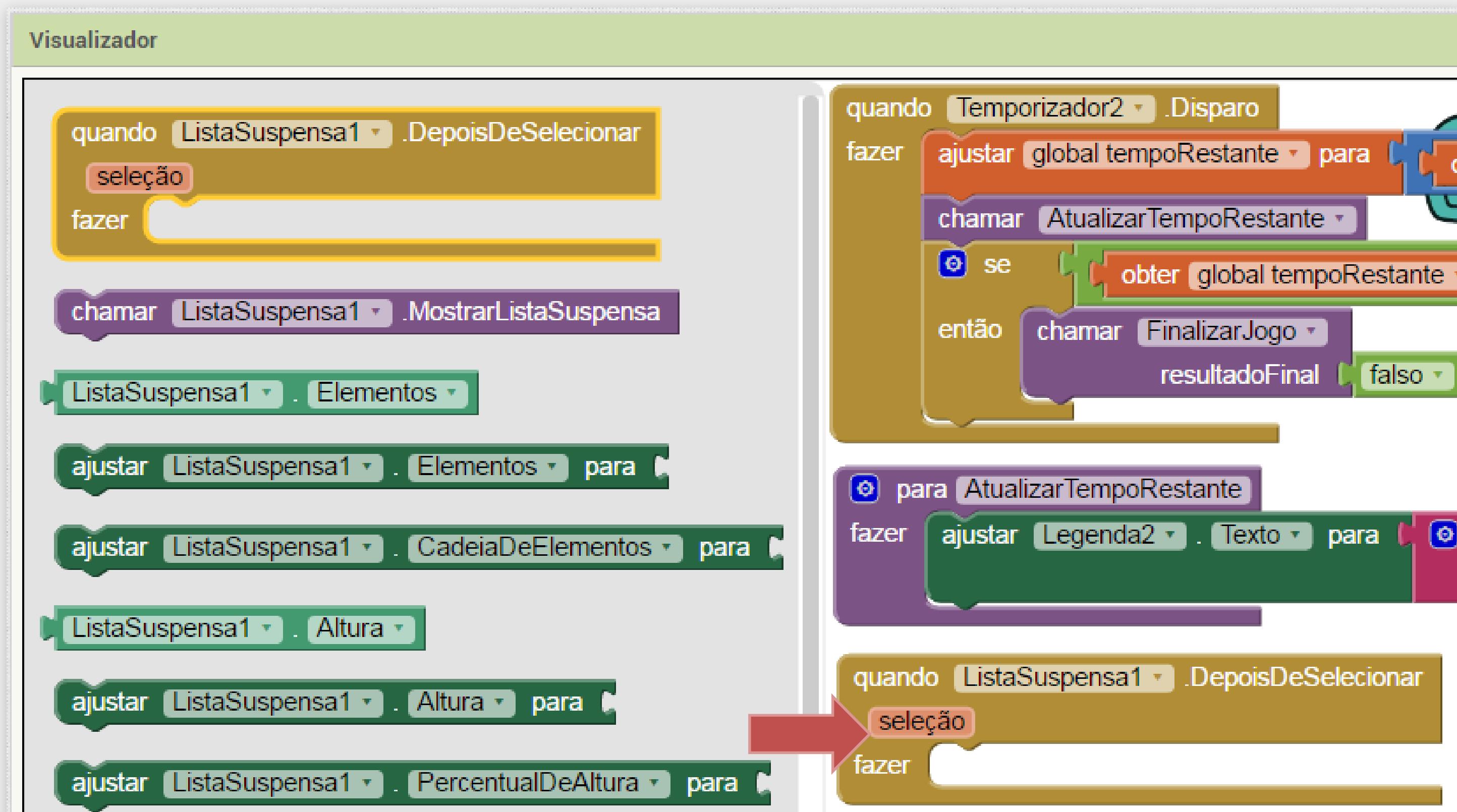


Iremos definir que o **nivelSelecionado** receberá os seguintes valores:

- 1 = Fácil
- 2 = Médio
- 3 = Difícil

ADICIONAR DIFICULDADE

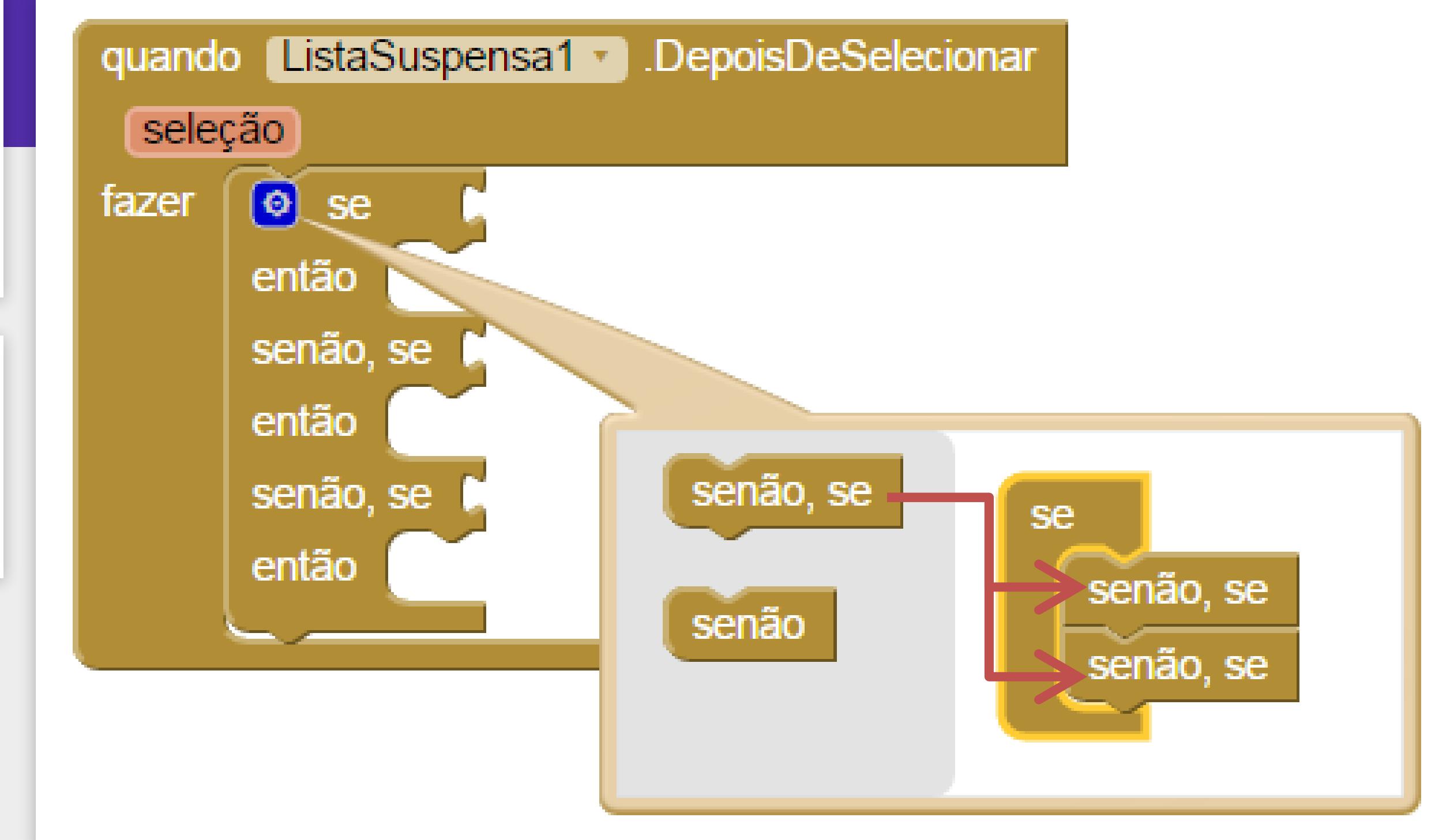
Utilizaremos o bloco “quando ListaSuspensa1.DepoisDeSelecionar” para definir qual o valor da variável nívelSelecionado. Este bloco possui uma variável local chamada de seleção que contém o valor escolhido (Fácil, Médio ou Difícil).



ADICIONAR DIFICULDADE

Iremos utilizar o bloco “Se então” e utilizar o ícone azul do canto superior para configurá-lo de acordo com a nossa lógica.

Precisamos adicionar duas vezes o bloco “senão, se” dentro do bloco “Se”.



ADICIONAR DIFICULDADE

NOSSA LÓGICA SERÁ

Se (seleção = "Facil")
Então nívelSelecionado = 1

Senão Se (seleção = "Medio")
Então nívelSelecionado = 2

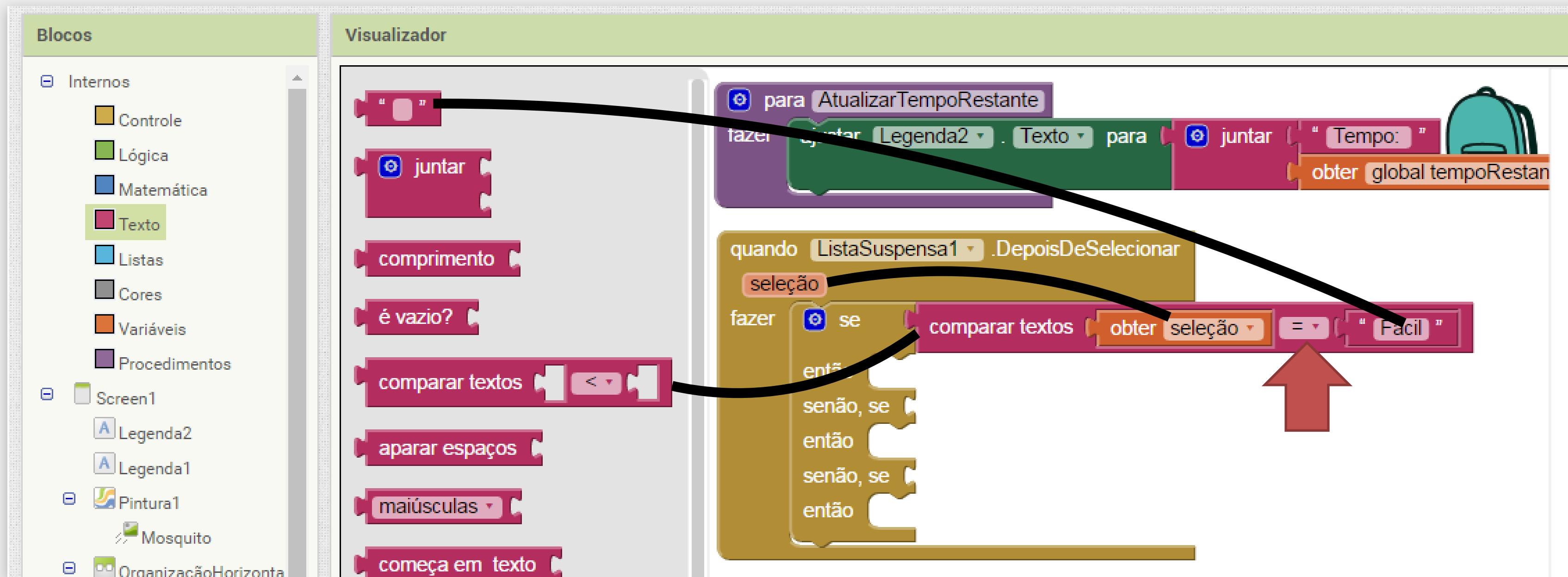
Senão Se (seleção = "Difícil")
Então nívelSelecionado = 3

ADICIONAR DIFICULDADE

Como o valor da seleção é um texto, devemos utilizar o bloco comparar textos para saber se o valor de seleção = “Facil” ou algum dos outros valores definidos.

Substitua o símbolo “<” por “=”.

Adicione o bloco obter seleção e o texto “Facil”.



ADICIONAR DIFICULDADE

O procedimento inteiro para seleção do nível de dificuldade na **ListaSuspensa1** deve ficar assim:



ADICIONAR DIFICULDADE

Agora que temos a variável **nivelSelecionado** configurada corretamente, devemos utilizar esta informação no procedimento de iniciar o jogo (**quando Botaoiniciar.Clique**) para alterar os valores de **tempoRestante** e Intervalo do **Temporizador1** de acordo com a dificuldade selecionada.

DICA

Lembrar que o Temporizador1 é o responsável por mandar o mosquito se mover.

ADICIONAR DIFICULDADE

Primeiro devemos adicionar o mesmo bloco “Se, então” com dois blocos “Senão, Se”, igual realizamos no passo anterior.

Nesse caso, vamos usar operações Lógicas “=” para saber qual o valor do nível Selecionado.



ADICIONAR DIFICULDADE

Vamos definir as dificuldades de acordo com a seguinte regra:

FÁCIL

tempoRestante = 20 s

Intervalo Temporizador1 = 500 ms

MÉDIO

tempoRestante = 10 s

Intervalo Temporizador1 = 400 ms

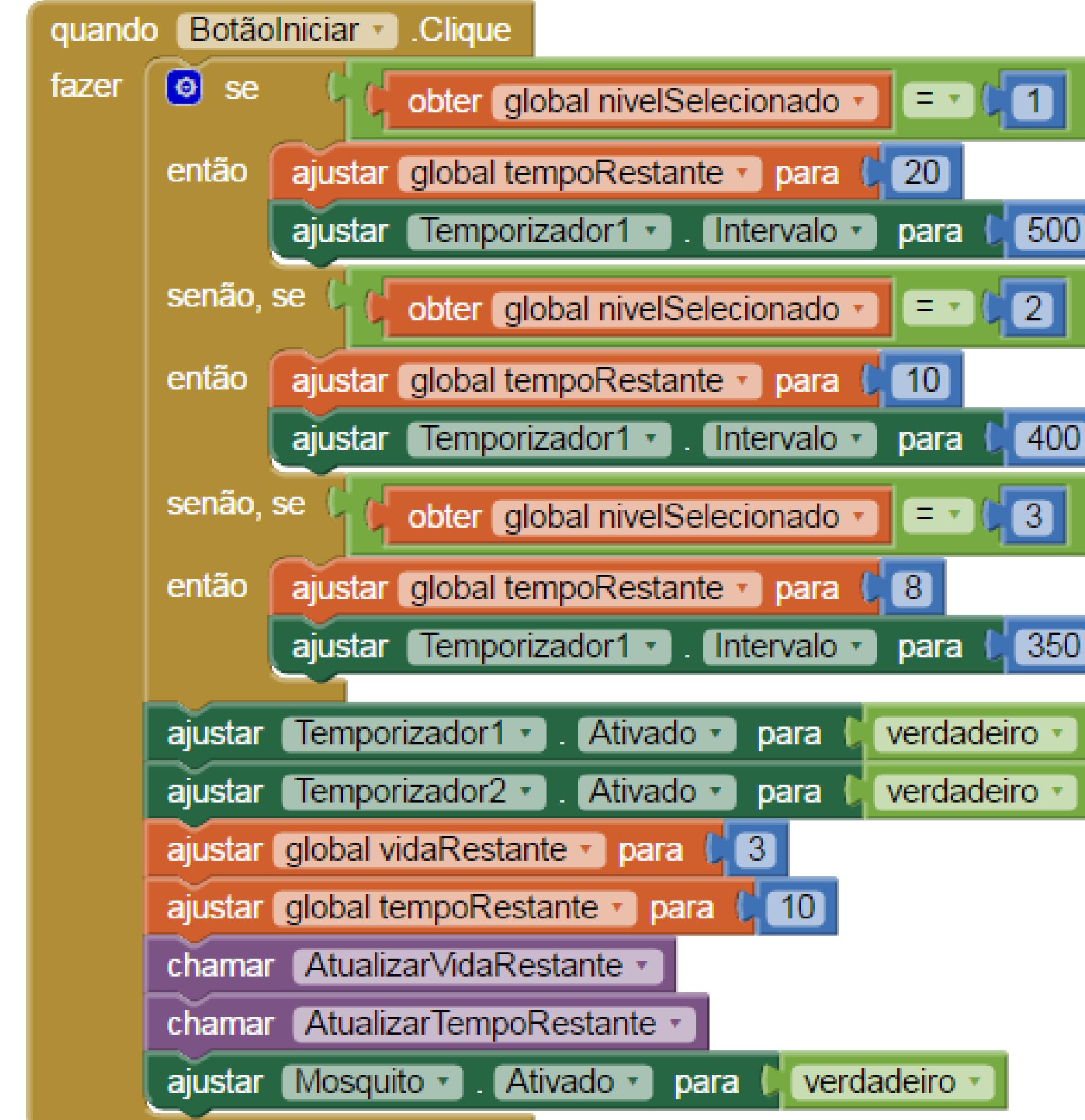
DIFÍCIL

tempoRestante = 8 s

Intervalo Temporizador1 = 350 ms

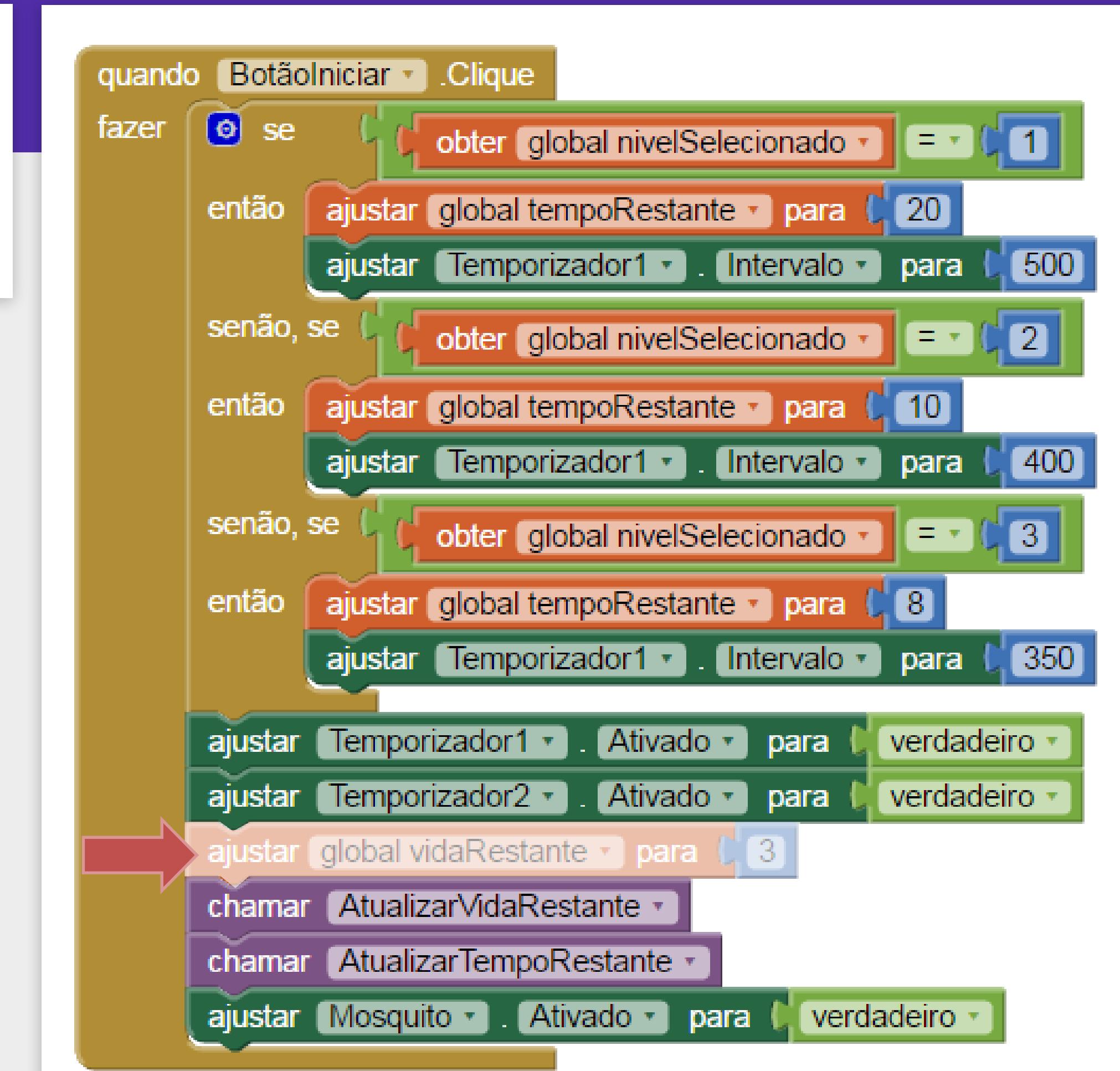
DICA

Lembre que você pode duplicar os blocos para ser mais rápido.



ADICIONAR DIFICULDADE

A última coisa a fazer é excluir o bloco “ajustar global tempoRestante para 10” que já tínhamos.

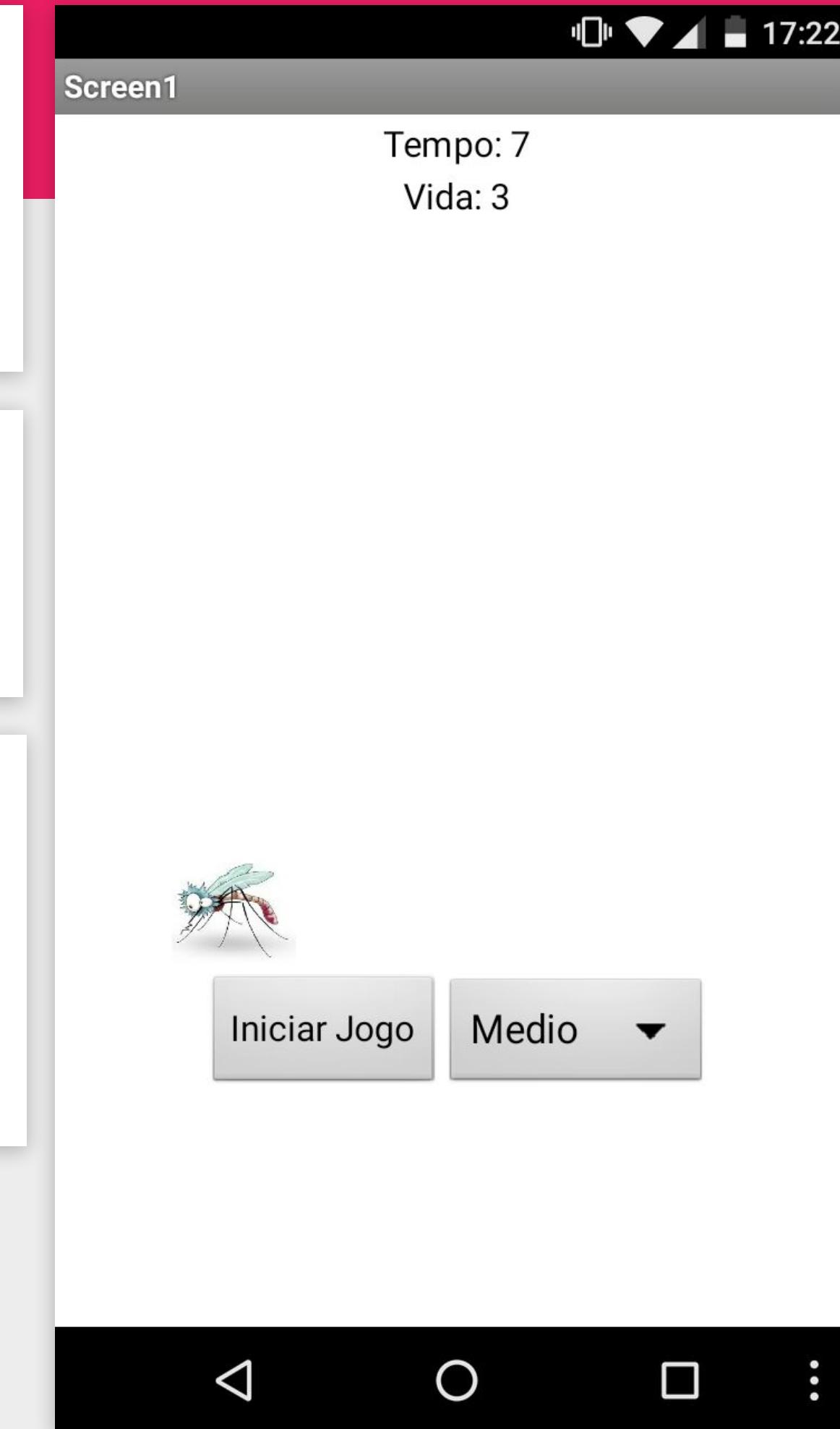


HORA DE TESTAR!

Teste o seu aplicativo e verifique se os níveis de dificuldade estão realmente funcionando.

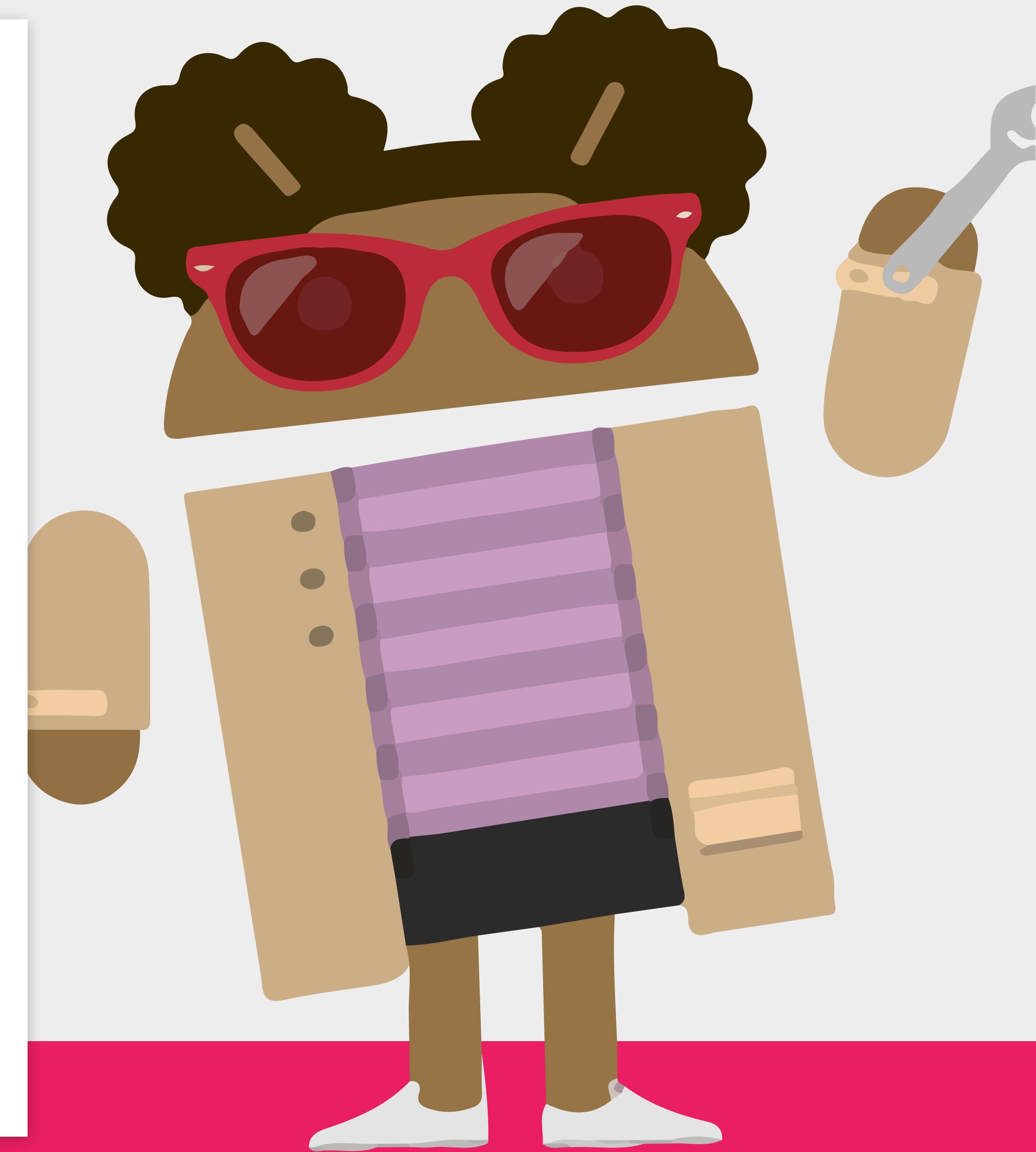
Agora o jogo deve estar funcionando perfeitamente!

Deste modo, finalizamos o desenvolvimento de todas as funcionalidades do jogo PegaMosquito.



AGORA É COM
VOÇÊ!

SEJA CRIATIVO



COMPUTAÇÃO NA ESCOLA

