

COMPUTAÇÃO
NA ESCOLA



TUTORIAL DO JOGO CAÇA MOSQUITO

OFICINA DE APP INVENTOR

INFORMAÇÕES GERAIS

HORÁRIO



DAS 8:00 ÀS 12:00

CAFÉ



A PARTIR DAS 10:00

BANHEIRO

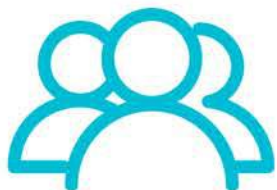


PERTO DO ELEVADOR



INFORMAÇÕES GERAIS

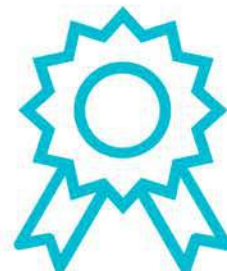
EQUIPE



MATERIAL DIDÁTICO



CERTIFICADO



USO DA IMAGEM



O QUE VAMOS **APRENDER?**

JOGO CAÇA MOSQUITO

OBJETIVO

Caçar o mosquito voando pela tela

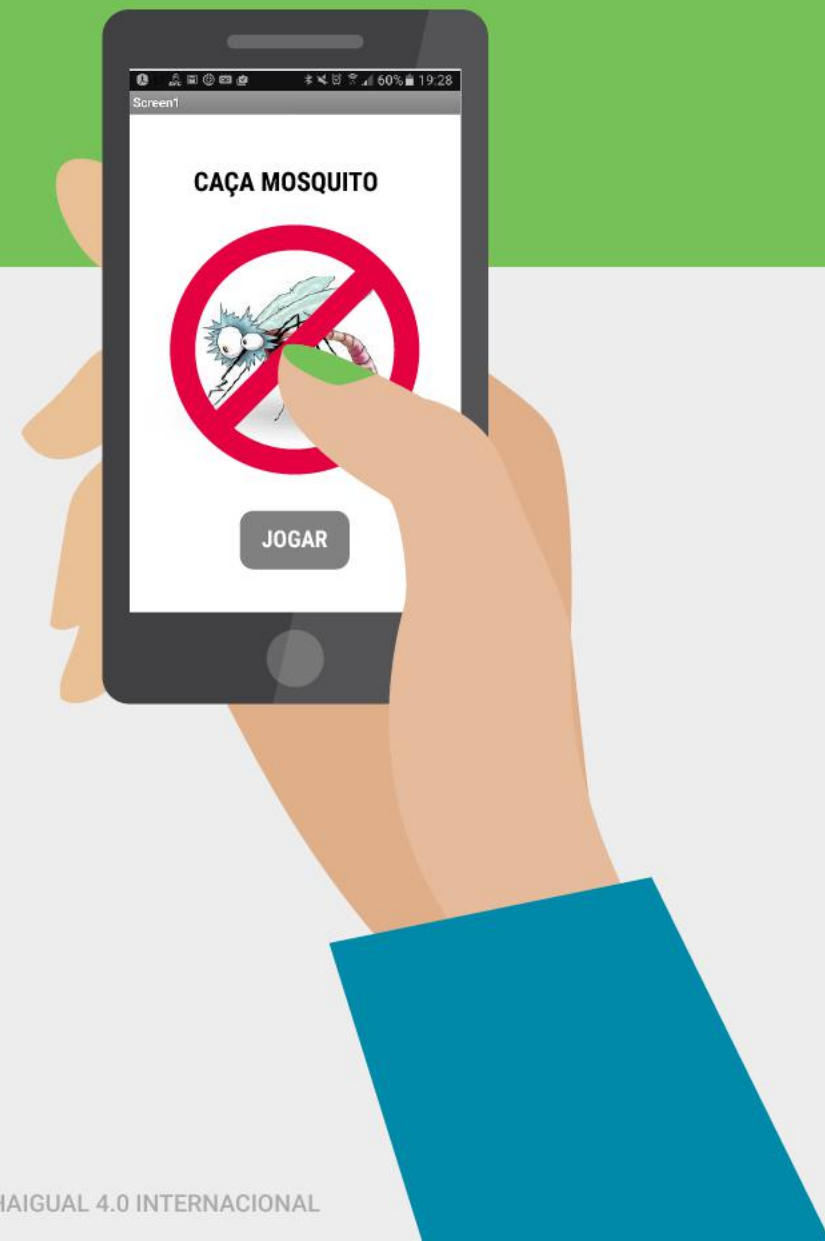
INFORMAÇÕES

Mosquito terá 3 vidas.

Jogador tem 10 segundos para eliminar mosquito.

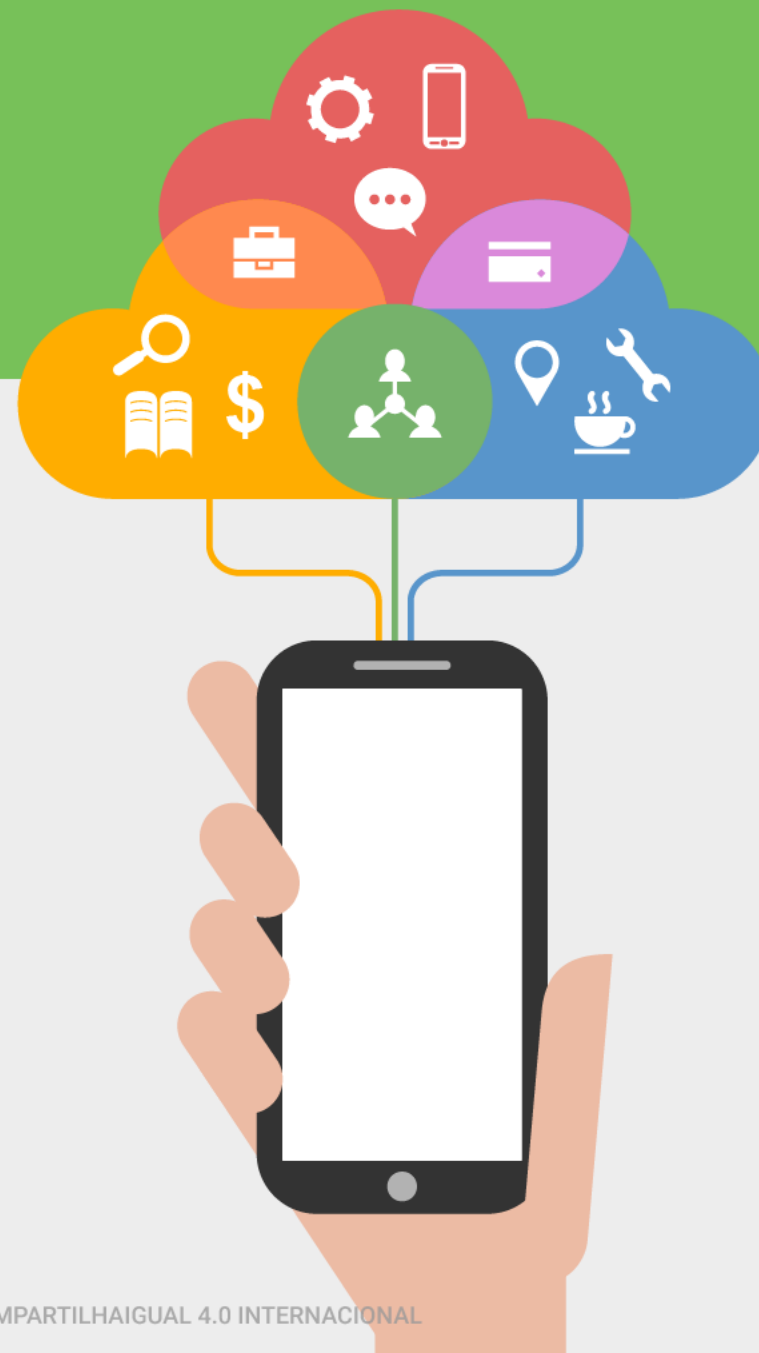


Fazer um mosquito se mover pela tela.
Adicionar vida ao mosquito.
Adicionar um tempo para matar o mosquito.
Incluir níveis de dificuldade ao jogo.



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

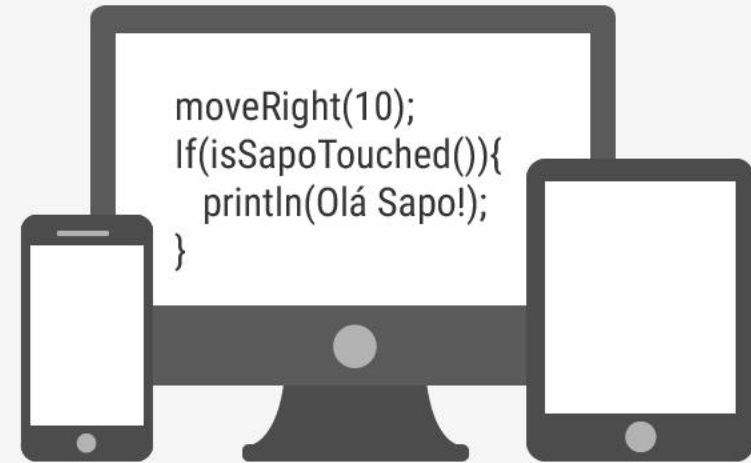
Um algoritmo é um conjunto de instruções para realizar uma determinada tarefa.

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

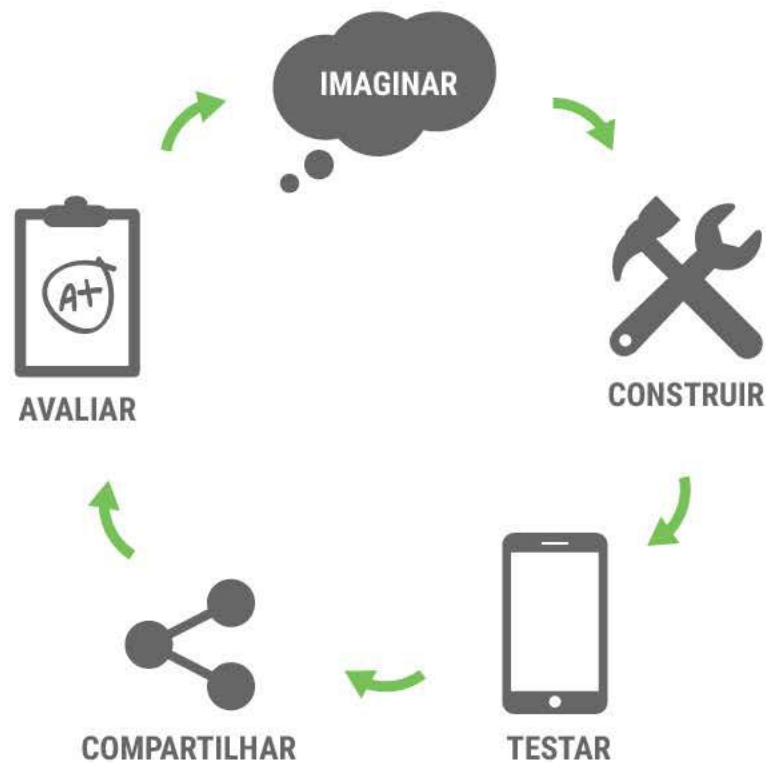


SOFTWARE

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



COMO FAZER APPS DE CELULAR



APP INVENTOR

appinventor.mit.edu

Ambiente de programação visual para programar apps para celulares android.
Software livre disponível gratuitamente, desenvolvido pelo MIT Media Lab/google.

The screenshot displays the MIT App Inventor web interface. At the top, there's a navigation bar with the MIT App Inventor logo, links for Home, Blog, and Support, and a prominent orange 'Create' button. Below this is a social media section with 'Follow Us' and icons for Facebook, Twitter, YouTube, and Email. A Google Custom Search bar is also present. The main workspace is divided into three panels: a left sidebar with a component palette (Logic, Math, Text, Lists, Colors, Variables, Procedures, Screen1, TextBox1, Button1, TextToSpeech1, AccelerometerSensor1), a central visual programming area with block-based code (e.g., 'when Button1.Click do call TextToSpeech1.Speak message TextBox1.Text'), and a right panel showing a mobile app preview and a large green 'Invent Now' button. The app preview shows a screen with a text box containing 'hello' and a 'Talk To Me' button.

VAMOS COMEÇAR

APPINVENTOR.MIT.EDU





MIT App Inventor

About ▾

News & Stories ▾

Resources ▾

Create apps!



Anyone Can Build Apps That Impact the World

Google Custom S



Get Involved with MIT App Inventor



Verizon Innovative Learning
App Challenge



MIT MASTER TRAINERS PROGRAM
IN EDUCATIONAL MOBILE COMPUTING

In-Person
Workshop
at MIT
July 31 - Aug 2,
2017

Online
Coursework
begins
Feb 7, 2017



MIT APP INVENTOR

Part 1 begins Feb 7 with
edX App Inventor MOOC

Tweets by @MITAppInventor



MIT App Inventor Retweeted



Sonja Upton
@SonjaUpton1

#CODEGirlCODE practicing their pitch for
the Obstacle challenge at



Digite sua senha novamente

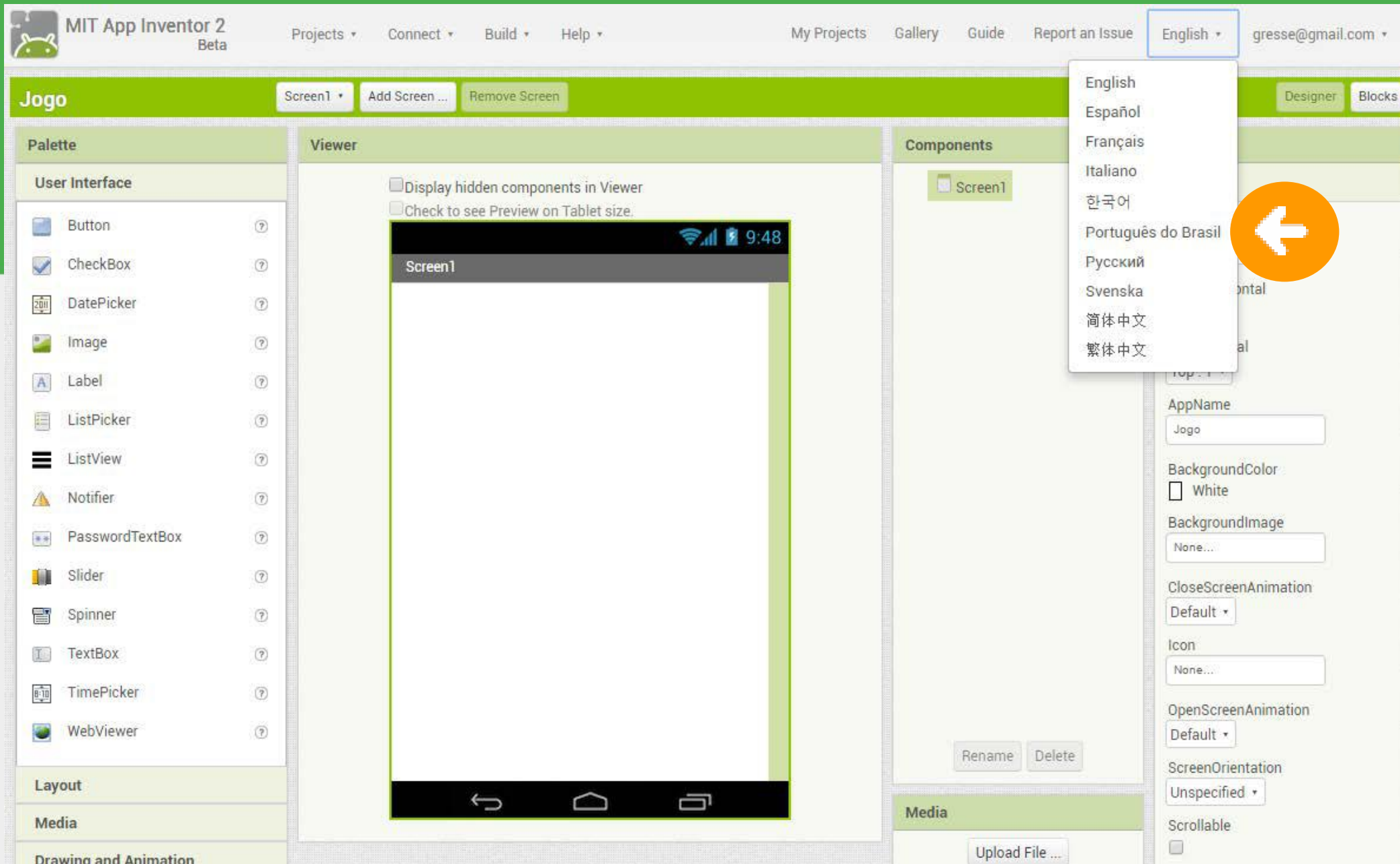


incod@incod.ufsc.br

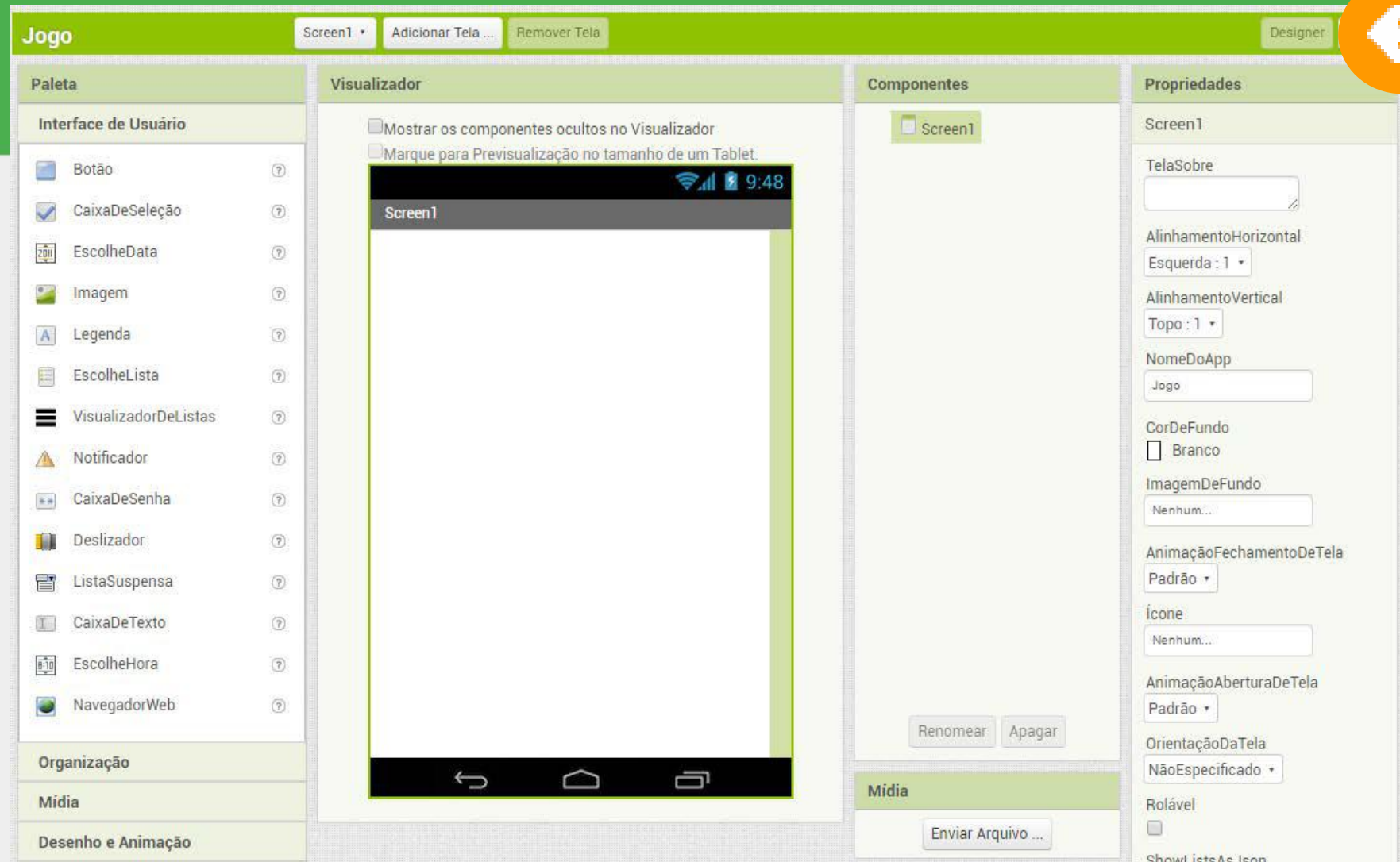
Fazer login

[Precisa de ajuda?](#)

[Acesse com uma conta diferente](#)



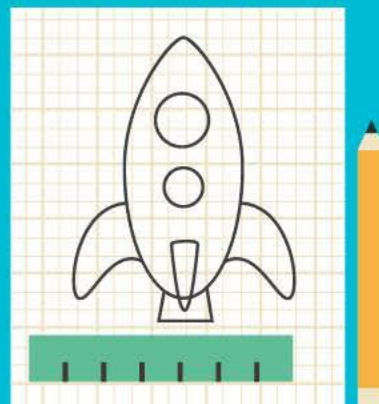
DESIGN DE INTERFACE



BLOCOS DE FUNCIONALIDADE

The screenshot displays the Scratch IDE for a project titled "CacaMosquito". The interface is divided into several sections:

- Top Bar:** Contains the project name "CacaMosquito", a dropdown menu set to "Screen1", and buttons for "Adicionar Tela ..." and "Remover Tela". On the right, there are tabs for "Designer" and "Blocos".
- Blocos (Blocks) Panel:** Located on the left, it shows a categorized list of blocks under "Internos":
 - Controle
 - Lógica
 - Matemática
 - Texto
 - Listas
 - Cores
 - Variáveis
 - ProcedimentosBelow these, it lists objects on the stage: "Screen1", "Legenda2", "Button2", "Button1", "Legenda1", "Label1", "Pintura1", "mosquito", and "OrganizaçãoHorizontal". At the bottom of this panel are "Renomear" and "Apagar" buttons.
- Visualizador (Stage) Panel:** The main workspace on the right where the script is built. It contains a "mosquito" sprite and a "Legenda1" text object. The script is as follows:
 - Global Variables:** Two "inicializar global" blocks. The first sets "tempoRestante" to 20. The second sets "vidaRestante" to 3.
 - When Green Flag Clicked:** A "quando" block with "Temporizador1" set to "Disparo", followed by a "fazer" block containing a "chamar" block for "MoverMosquito".
 - Loop for MoverMosquito:** A "para" loop for "MoverMosquito" with two "ajustar" blocks. Both "ajustar" blocks set the "X" and "Y" coordinates of "mosquito" to "inteiro aleatório de 0 até 270".
 - Loop for AtualizarTempoRestante:** A "para" loop for "AtualizarTempoRestante" with a "fazer" block. This block contains an "ajustar" block for "Legenda2" . "Texto" and a "juntar" block. The "juntar" block concatenates "Tempo Restante: " with the value of the "global tempoRestante" variable.
 - Loop for FinalizarJogo:** A "para" loop for "FinalizarJogo" with a "resultadoFinal" variable. It contains a "fazer" block with a "se" (if) statement:
 - If Condition:** "obter resultadoFinal" equals "verdadeiro".
 - Then:** "ajustar Legenda1" . "Texto" to "Você eliminou o mosquito!".
 - Else:** "ajustar Legenda1" . "Texto" to "Você Perdeu".



CRIANDO UM NOVO PROJETO

CRIANDO UM **NOVO PROJETO**

Vamos iniciar criando um novo projeto no AppInventor.



CRIANDO UM **NOVO PROJETO**

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

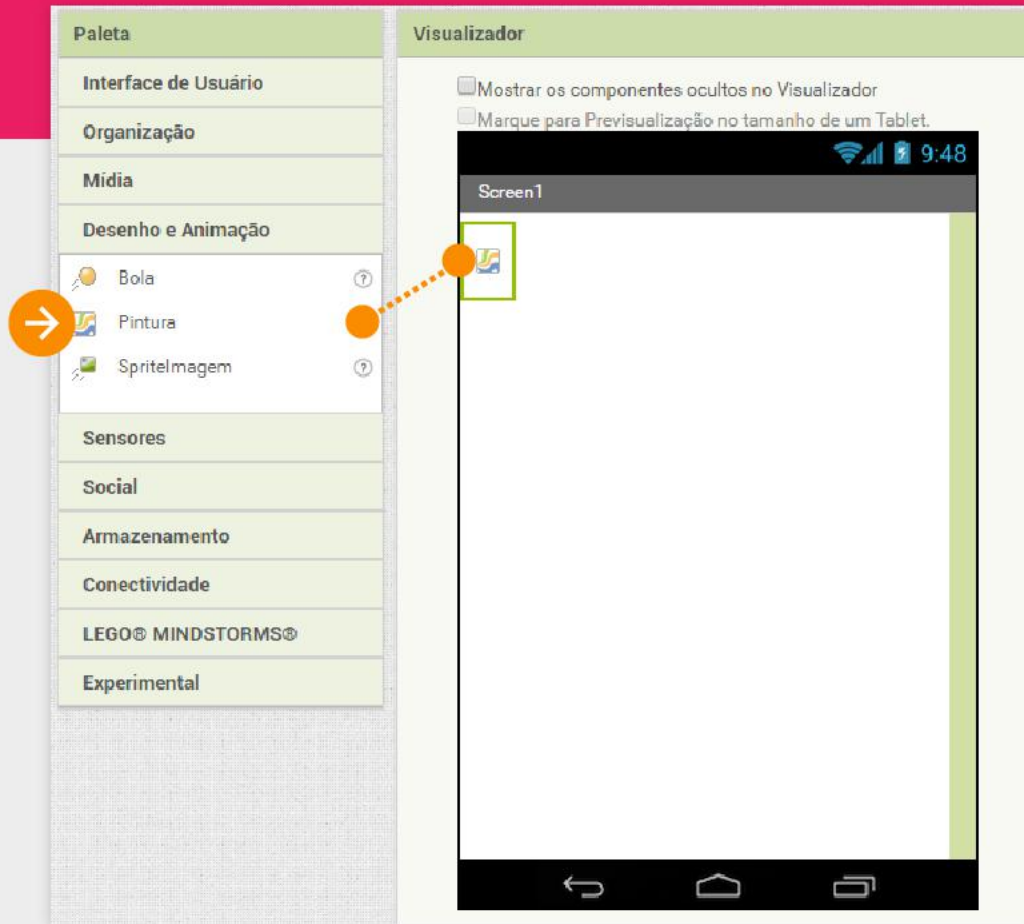


The image shows a dialog box titled "Criar um novo projeto no App Inventor" (Create a new project in App Inventor). The dialog box has a green header bar with the title in white text. Below the header, there is a label "Nome do projeto:" (Project name:). To the right of the label is a text input field containing the text "PegaMosquito". At the bottom of the dialog box, there are two buttons: "Cancelar" (Cancel) on the left and "OK" on the right. The dialog box is set against a light gray background with a subtle grid pattern.

DEFININDO A **INTERFACE**



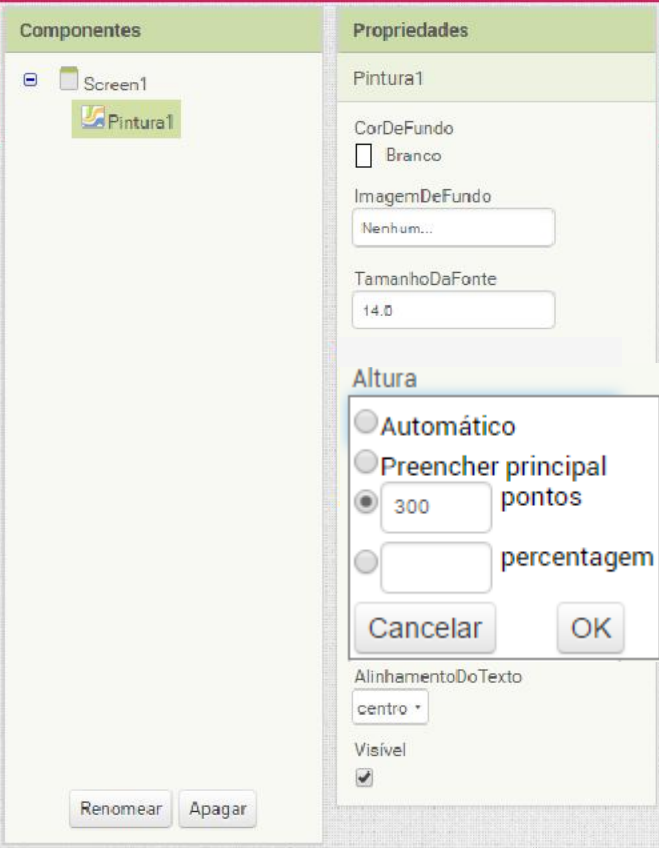
DEFININDO A INTERFACE



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).

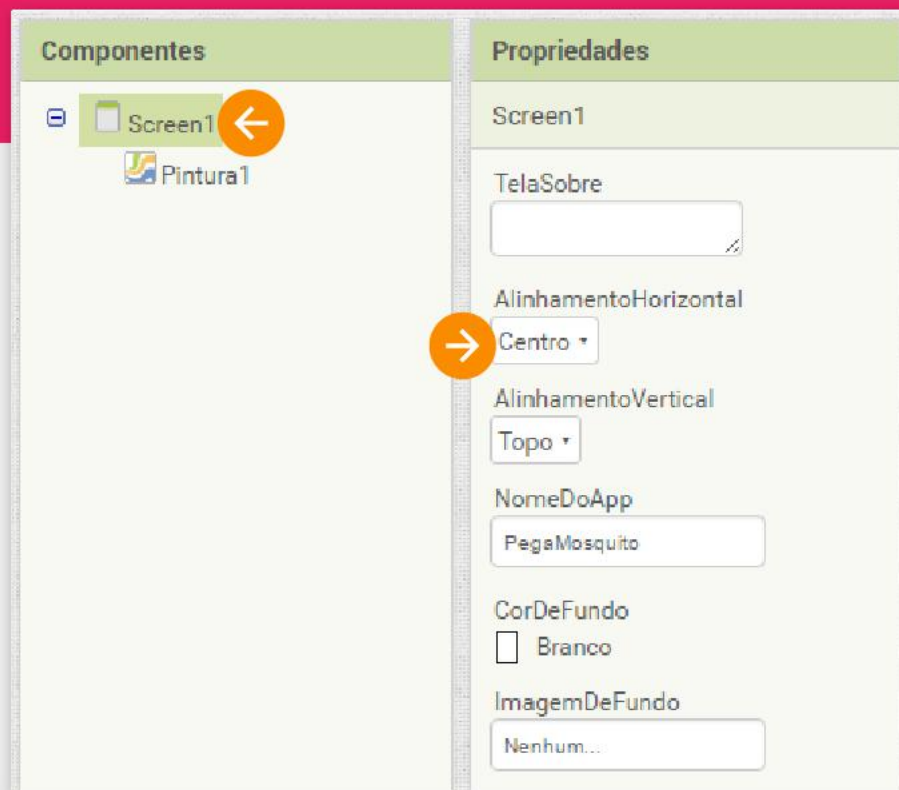
DEFININDO A INTERFACE



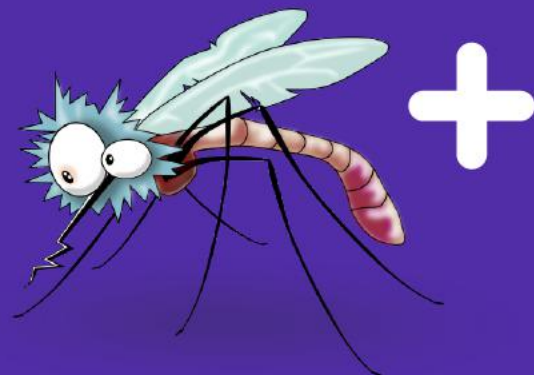
The screenshot shows a software interface with a 'Componentes' panel on the left containing 'Screen1' and 'Pintura1'. The 'Propriedades' panel for 'Pintura1' is open, showing settings for 'CorDeFundo' (set to Branco), 'ImagemDeFundo' (set to Nenhum...), and 'TamanhoDaFonte' (set to 14.0). Below these, the 'Altura' and 'Largura' dialog boxes are open. Both dialog boxes have 'Automático' selected, but 'Preencher principal' is also selected, and the value '300' is entered in the 'pontos' field. The 'AlinhamentoDoTexto' is set to 'centro' and 'Visível' is checked. At the bottom of the 'Propriedades' panel are 'Renomear' and 'Apagar' buttons.

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".

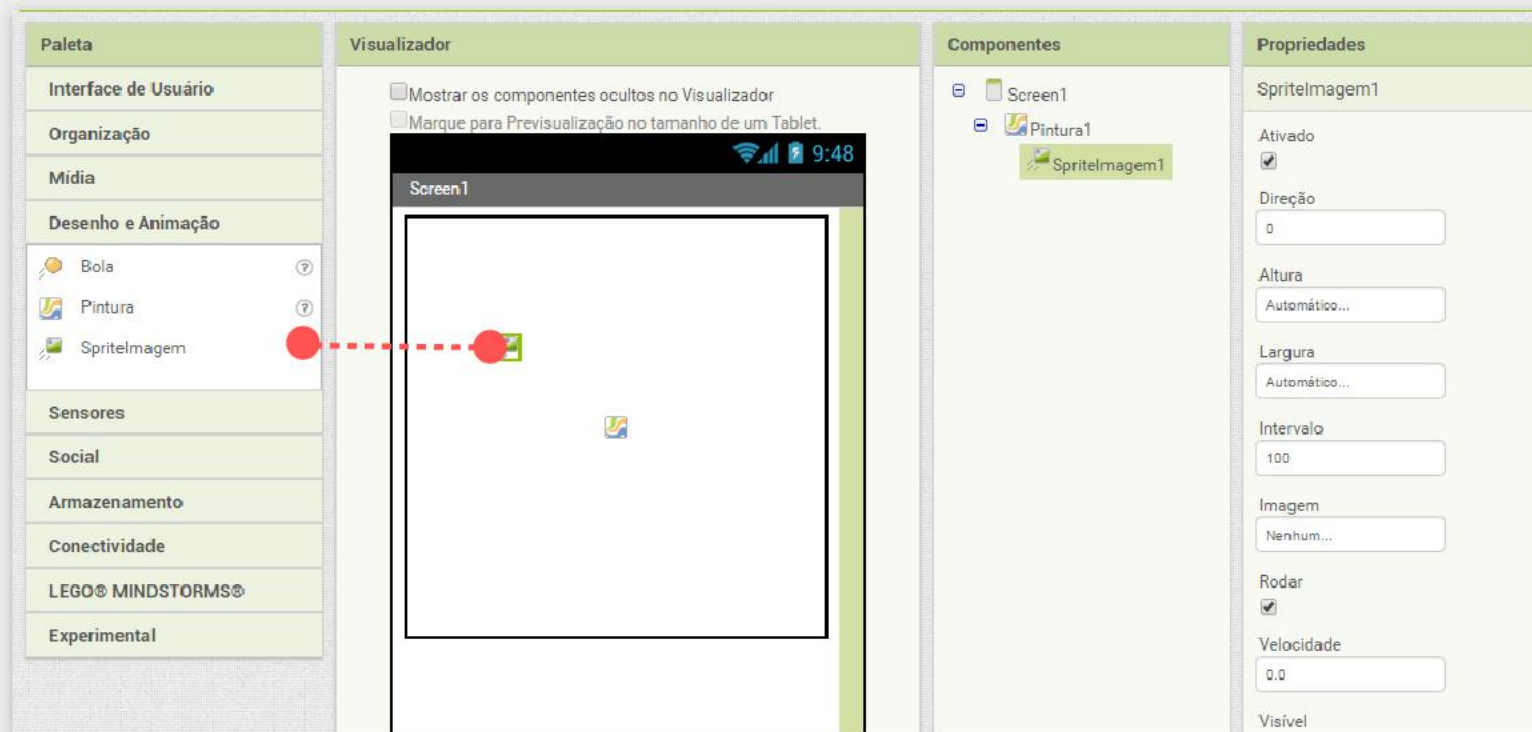


ADICIONANDO O
MOSQUITO

ADICIONANDO O MOSQUITO

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

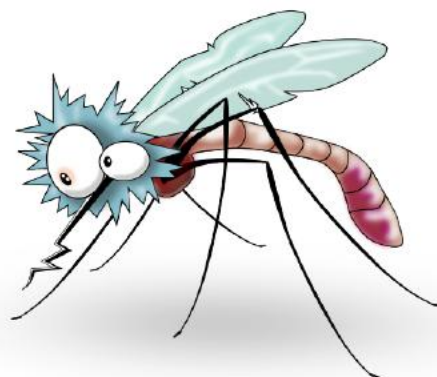
Selecione o componente **"SpriteImagem"** na Paleta de **"Desenho e Animação"** e arraste para a **"Pintura"**.



ADICIONANDO O MOSQUITO

Primeiro baixe a imagem do mosquito aqui:

<http://goo.gl/XRy7jL>



ADICIONANDO O MOSQUITO

Agora devemos definir a imagem dessa Sprite.

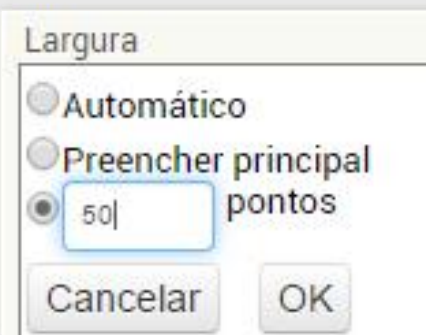
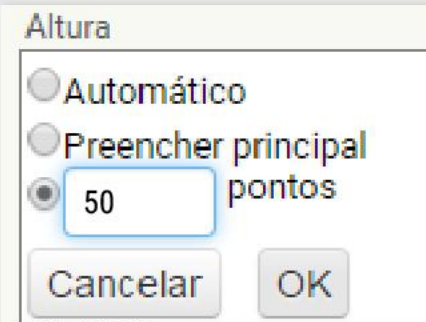
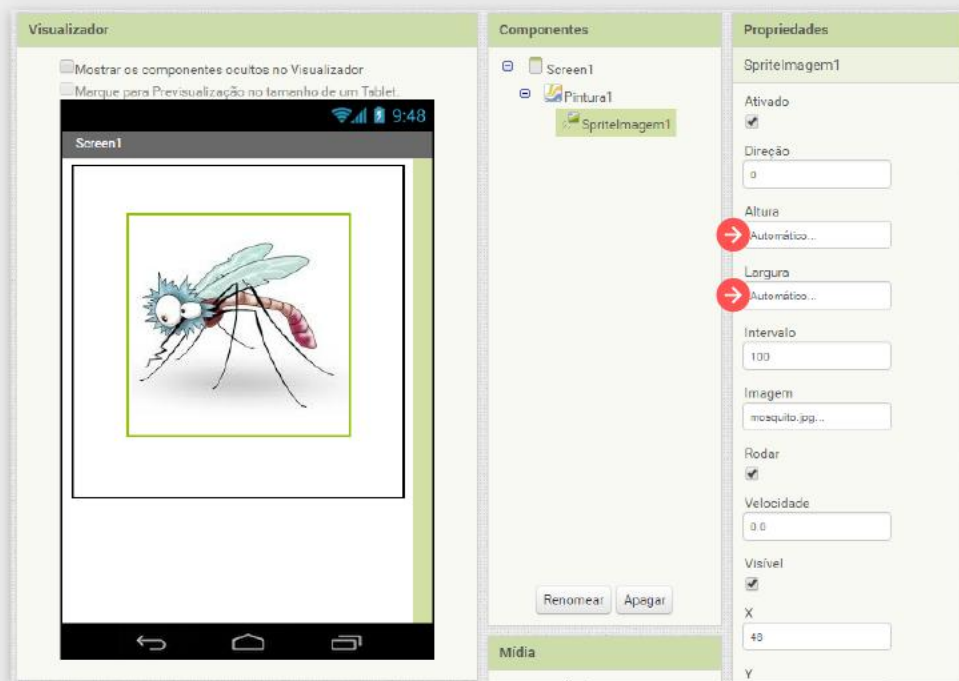
The screenshot displays the Scratch IDE interface with four main panels: Paleta, Visualizador, Componentes, and Propriedades.

- Paleta:** The 'Desenho e Animação' category is selected. The 'SpriteImagem' option is highlighted.
- Visualizador:** A 'Enviar Arquivo ...' dialog box is open, showing 'Choose File' with 'mosquito.jpg' selected. Below the dialog, a small mosquito image is visible on the stage.
- Componentes:** The 'Screen1' container holds 'Pintura1' and 'SpriteImagem1'.
- Propriedades:** The 'SpriteImagem1' properties panel is active. The 'Imagem' dropdown menu is open, showing 'Nenhum' as the selected option. The 'Enviar Arquivo ...' button is highlighted at the bottom of the panel.

ADICIONANDO O MOSQUITO

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

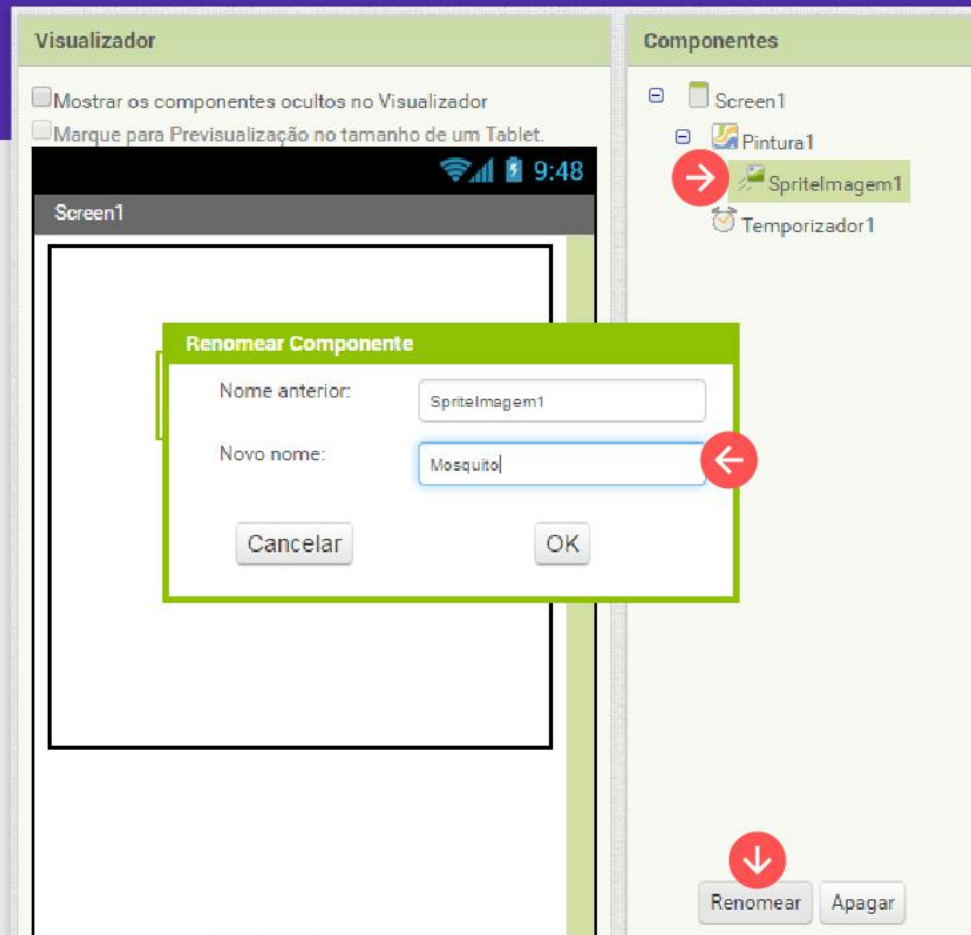
Mude a altura e largura do mosquito para 50.

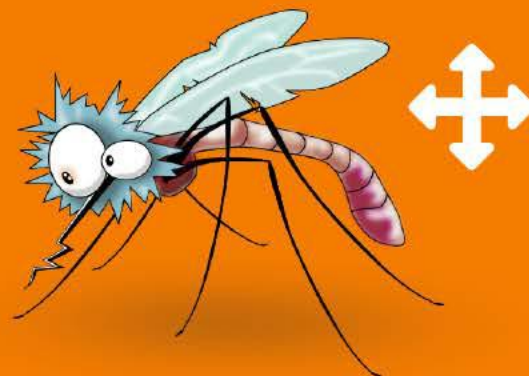


ADICIONANDO O MOSQUITO

Para ficar mais fácil reconhecer que o componente Spritelmagem1 é a imagem do nosso mosquito, vamos mudar o nome dele para **"Mosquito"**.

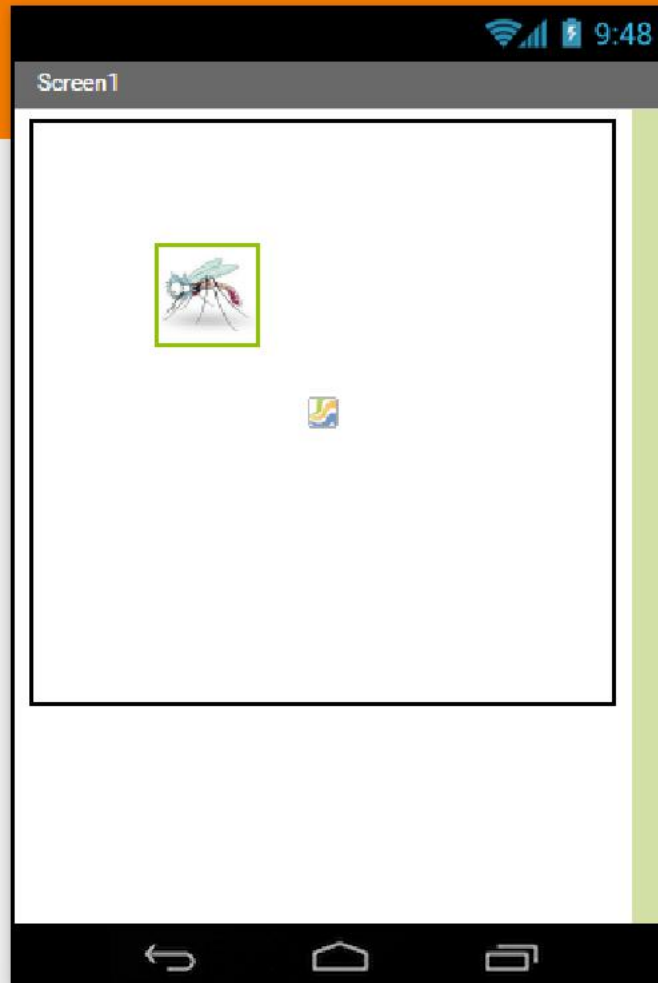
Selecione o componente e clique em Renomear. Digite **"Mosquito"** e clique em OK.





MOVER O MOSQUITO
PELA TELA

MOVER O MOSQUITO **PELA TELA**



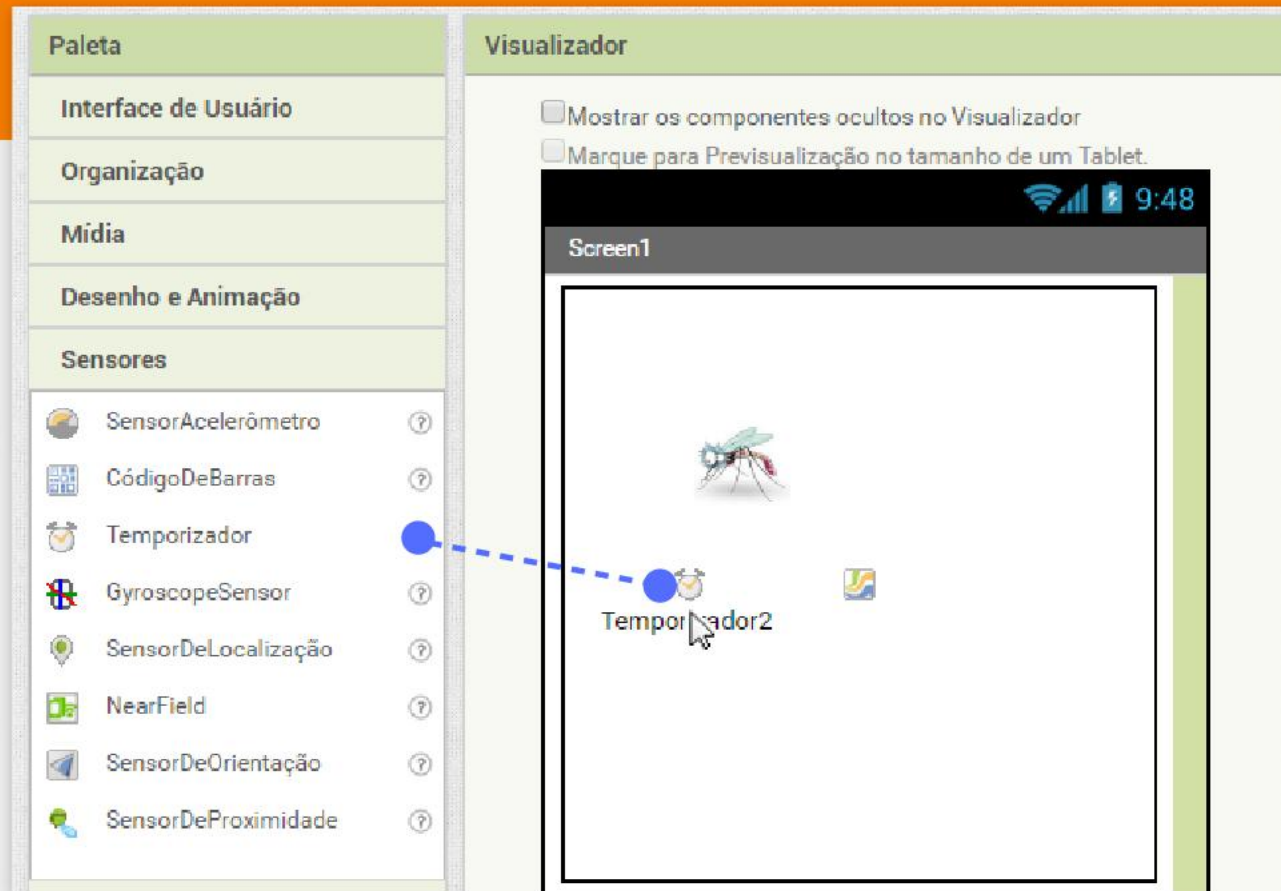
Por enquanto só temos um mosquito parado na tela.

AGORA VAMOS FAZÊ-LO SE MOVIMENTAR.

MOVER O MOSQUITO PELA TELA

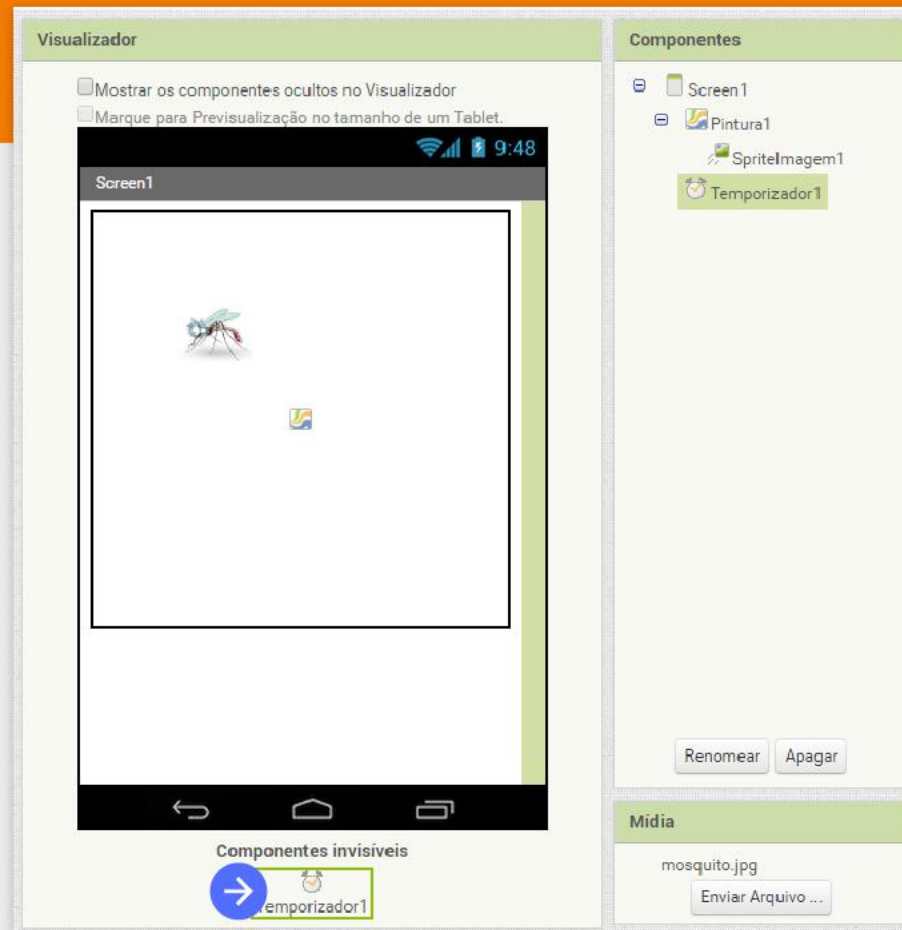
Para o mosquito se movimentar iremos utilizar o componente **Temporizador**. Este será responsável por mandar o mosquito se movimentar a cada meio segundo.

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**”.

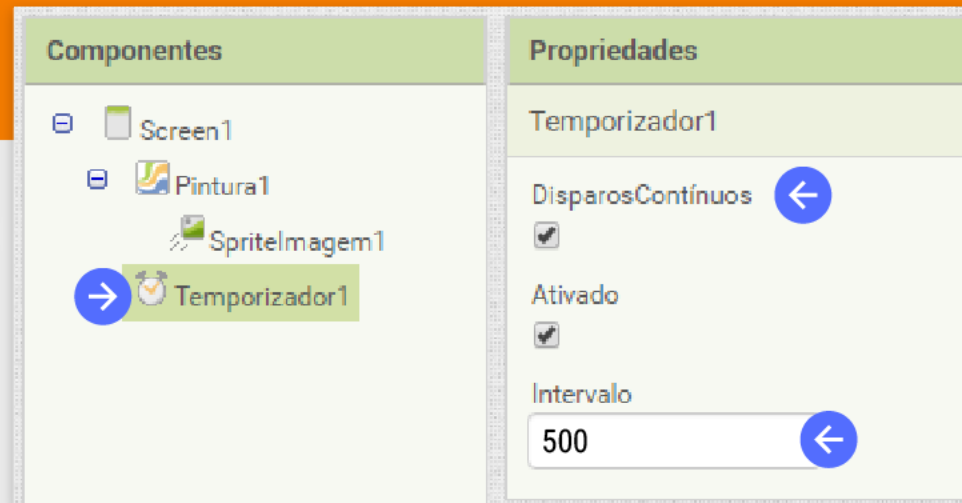


MOVER O MOSQUITO **PELA TELA**

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.



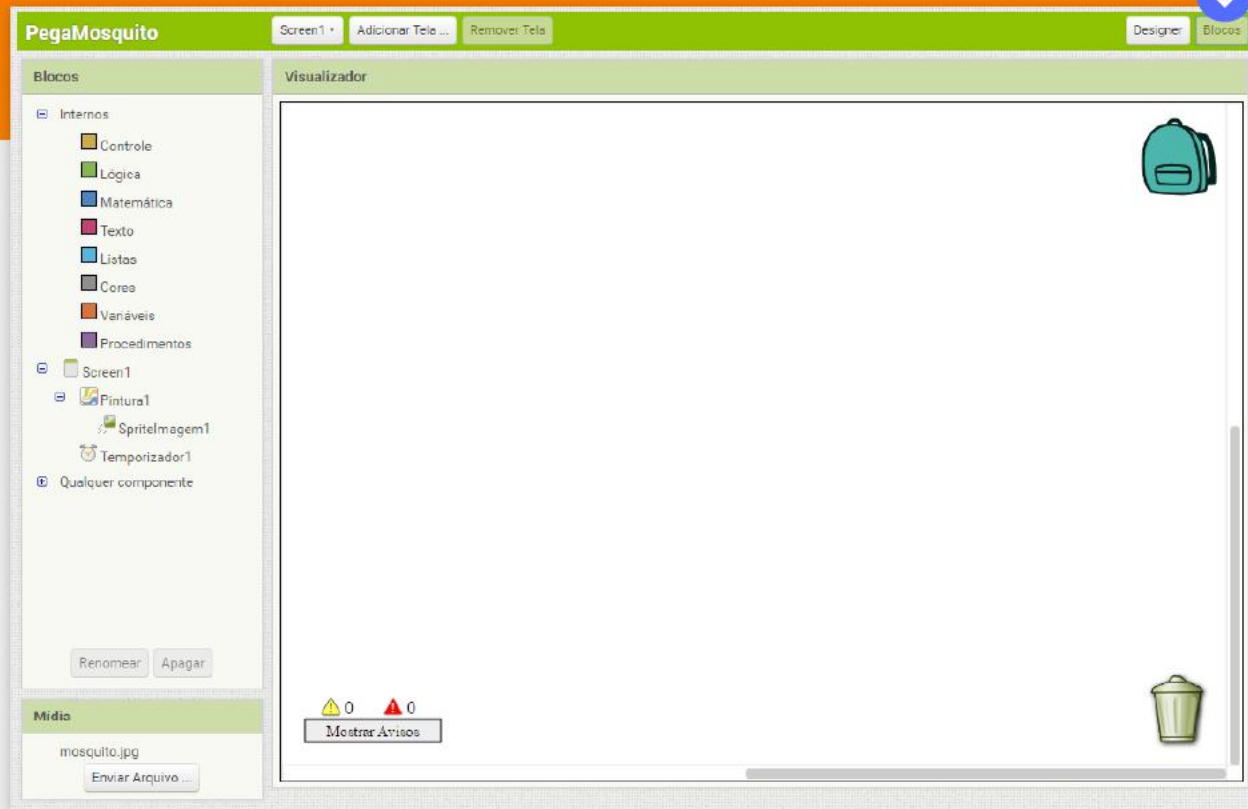
DICA

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.

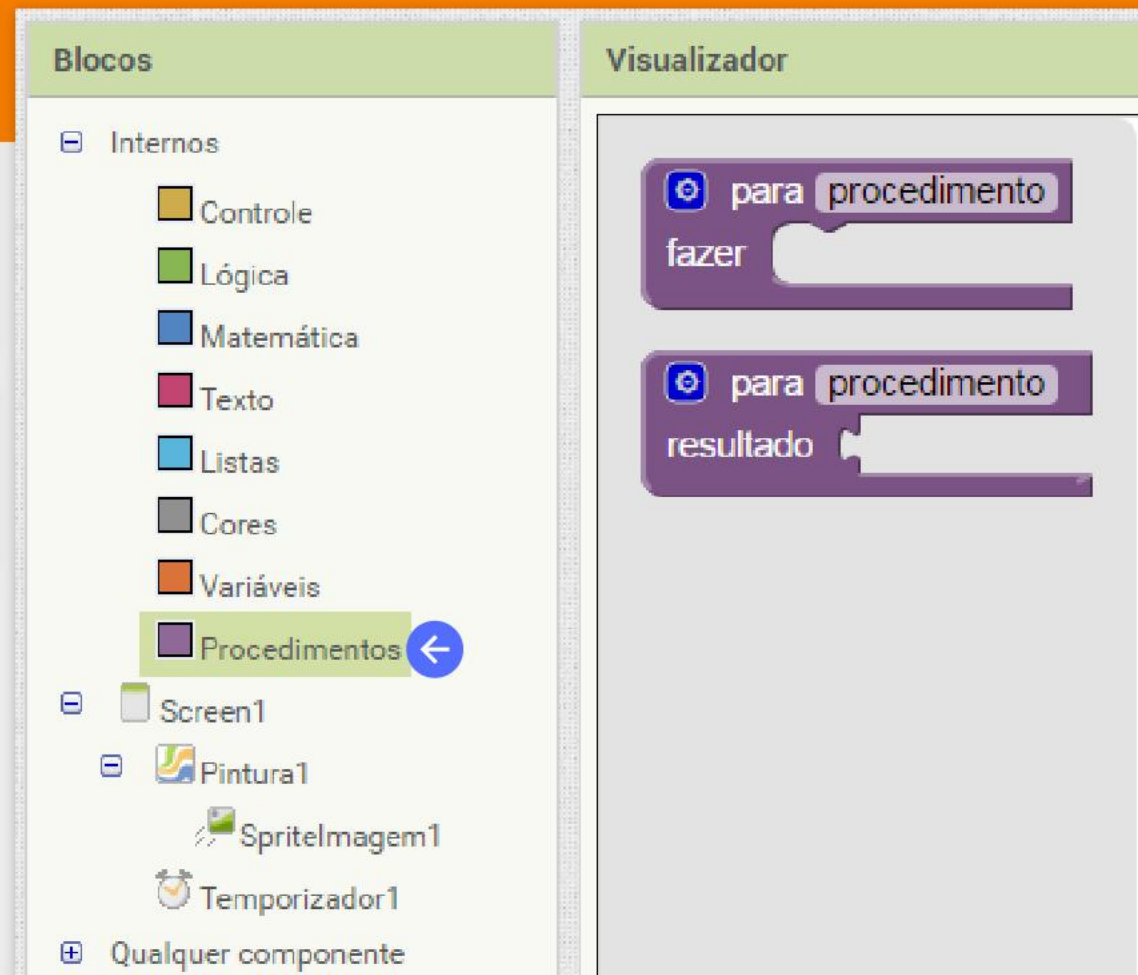
Até então estávamos trabalhando na tela de Design do nosso aplicativo, agora vamos passar para a tela de programação, ou **Blocos**.



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.



MOVER O MOSQUITO **PELA TELA**

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

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

Visualizador

para procedimento
fazer

para procedimento
resultado

chamar MoverMosquito

para MoverMosquito
fazer



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 **SpriteImagem**, ou seja, do “Mosquito”.

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

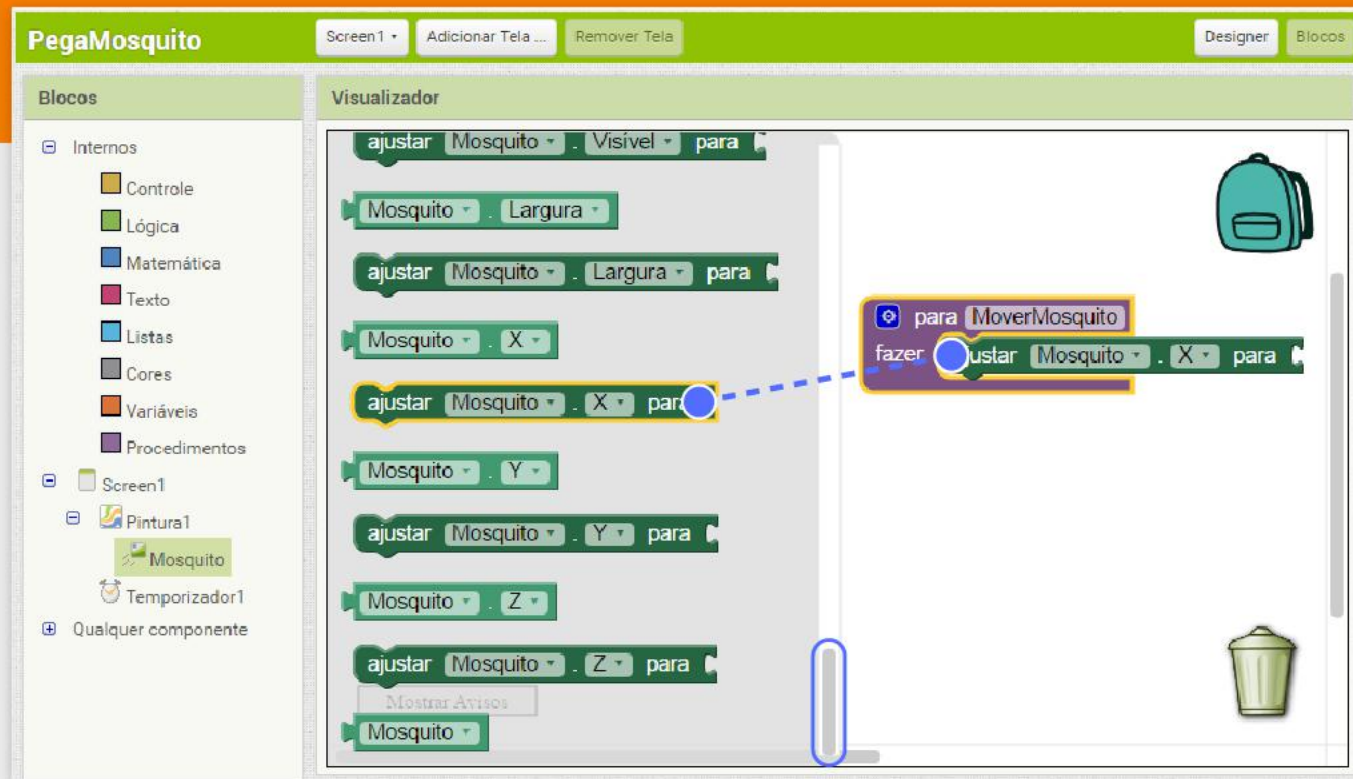


MOVER O MOSQUITO PELA TELA

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.

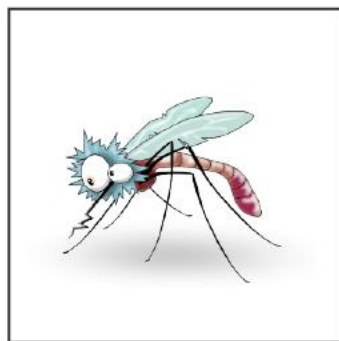


MOVER O MOSQUITO PELA TELA

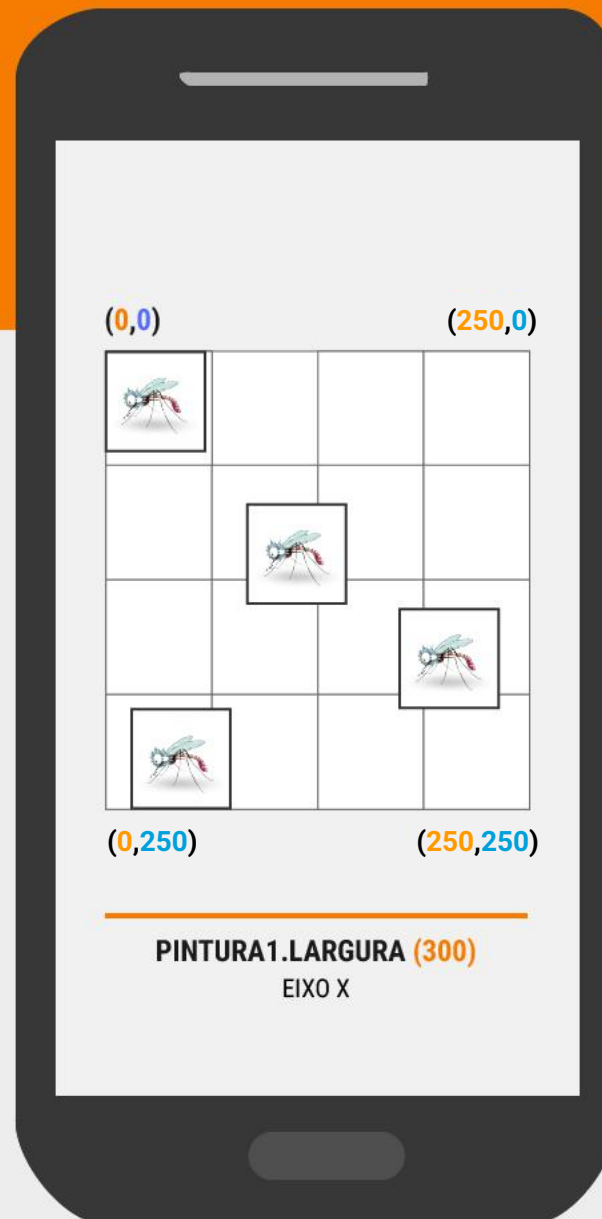
Para que posição queremos mover o mosquito?

A posição deve ficar entre 0 (zero) e 250 tamanho da tela menos o tamanho do mosquito)

MOSQUITO.ALTURA (50)



MOSQUITO.LARGURA (50)



MOVER O MOSQUITO PELA TELA

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

The image shows the Scratch interface with two panels: 'Blocs' and 'Visualizador'.

Blocs Panel: A list of categories is shown. The 'Matemática' category is highlighted with a green background and a green arrow pointing to it. Other categories include 'Internos', 'Controle', 'Lógica', 'Texto', 'Listas', 'Cores', 'Variáveis', 'Procedimentos', 'Screen1', 'Pintura1', 'Mosquito', and 'Temporizador1'.

Visualizador Panel: A script is shown. It starts with a 'movemos para' block, followed by an 'ajustar Mosquito X para' block. The 'ajustar Mosquito X para' block is connected to an 'inteiro aleatório de 1 até 100' block. The 'inteiro aleatório de 1 até 100' block is highlighted with a yellow border.

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 250).



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 interface with two panels: 'Blocs' and 'Visualizador'.

Blocs Panel:

- Internos
 - Controle
 - Lógica
 - Matemática
 - Texto
 - Listas
 - Cores
 - Variáveis
 - Procedimentos
- Screen1
 - Pintura1
 - Mosquito
- Qualquer componente
 - Temporizador1

Visualizador Panel:

- quando Temporizador1 .Disparo
- fazer
- chamar Temporizador1 .Add Days
 - instante
 - quantity
- chamar Temporizador1 .Add Duration
 - instante
 - quantity
- chamar Temporizador1 .Add Hours
 - instante
 - quantity
- chamar Temporizador1 .Add Minutes
 - instante

An orange arrow points from the 'Temporizador1' block in the 'Blocs' panel to the 'quando Temporizador1 .Disparo' block in the 'Visualizador' panel.

MOVER O MOSQUITO PELA TELA

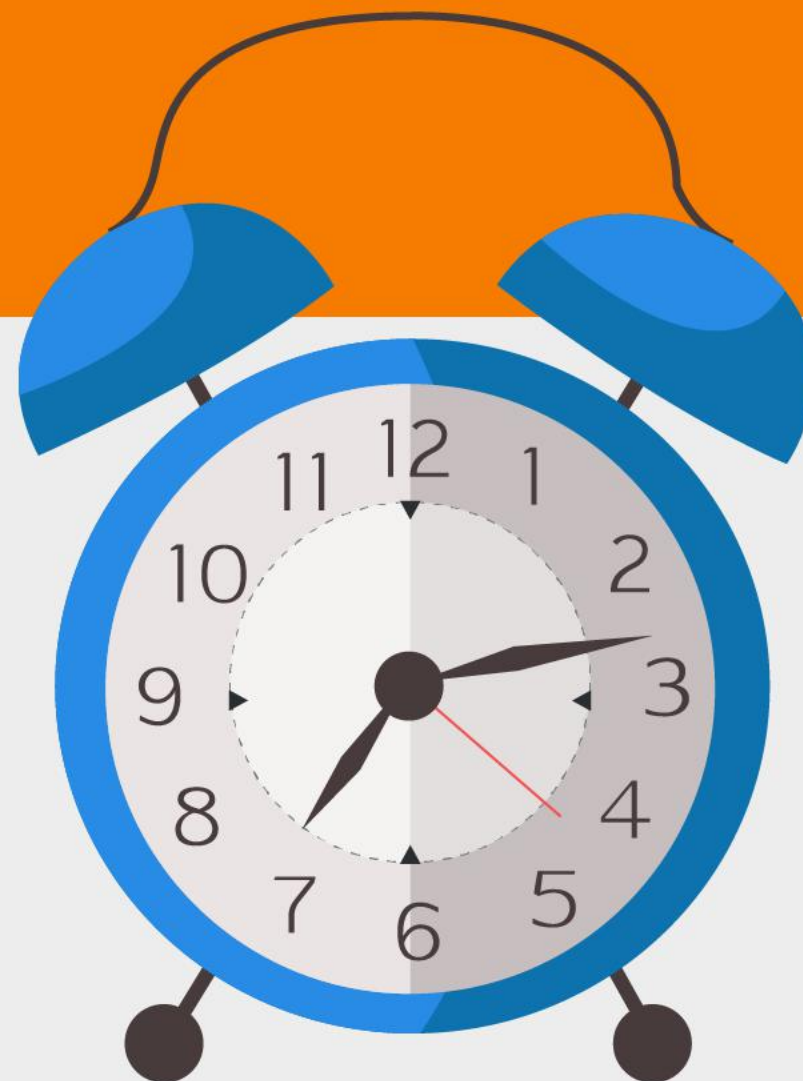
EVENTO

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

NORMAL

CLICADO

⚠ quando Button1 .Clique
fazer



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.

The screenshot shows the Scratch interface with two main panels: 'Blocos' (Blocks) on the left and 'Visualizador' (Stage) on the right. In the 'Blocos' panel, the 'Internos' (Internal) category is expanded, and the 'Procedimentos' (Behaviors) sub-category is highlighted. A yellow box highlights the 'chamar MoverMosquito' (Call MoverMosquito) block, and a white arrow points to it. In the 'Visualizador' panel, the 'MoverMosquito' object is visible on the stage. The 'Disparo' (Fire) block of 'Temporizador1' is highlighted, and the 'chamar MoverMosquito' block is being dragged into it. The 'Visualizador' panel also shows a 'para MoverMosquito' (for MoverMosquito) loop with an 'ajustar Mosquito' (set Mosquito) block and an 'X' block.

**VAMOS TESTAR
NO 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

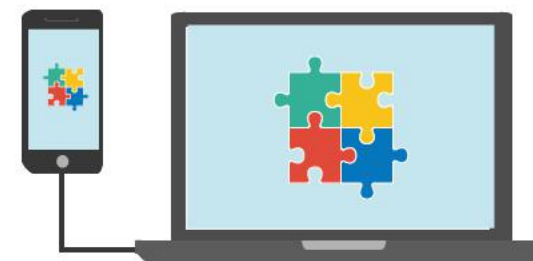
OPÇÃO 2

Não tem um dispositivo Android? Use um emulador



OPÇÃO 3

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



INSTALANDO O APLICATIVO

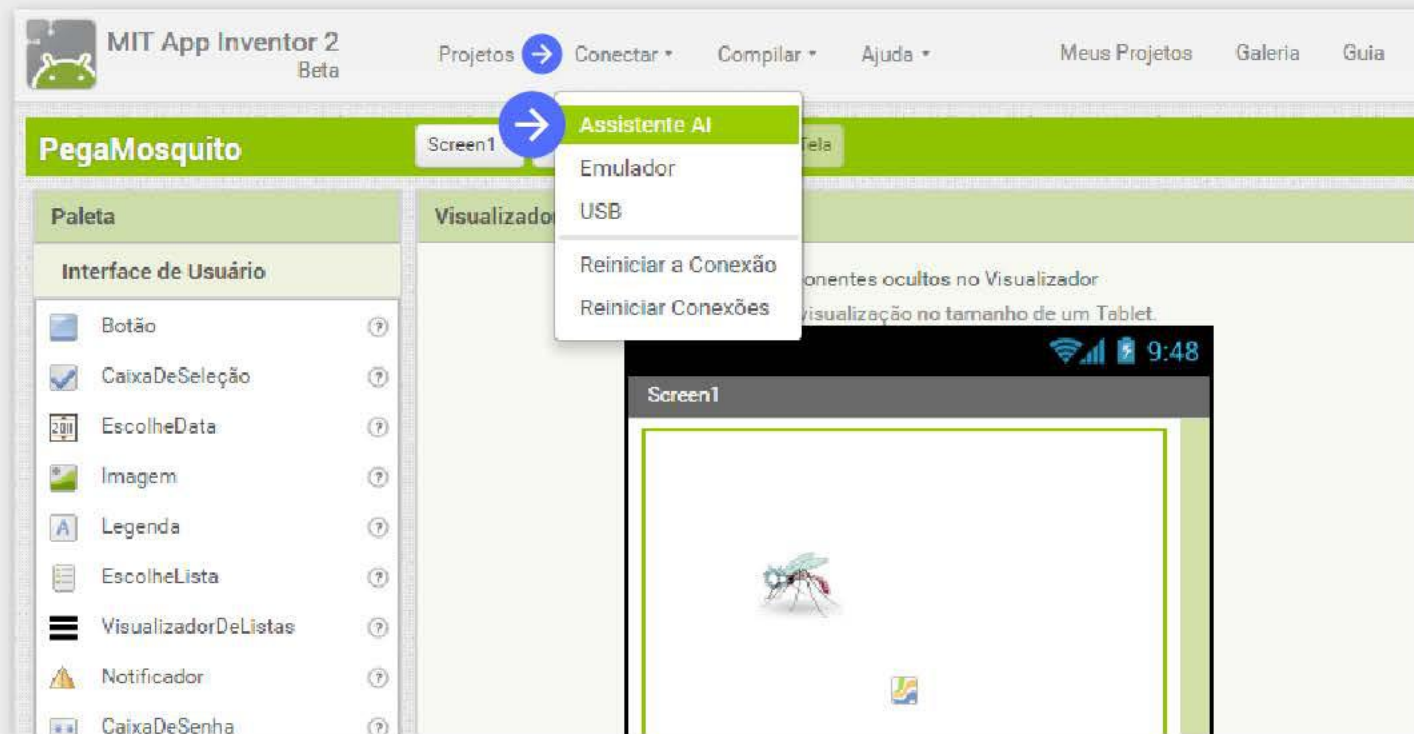
Para poder testar o seu jogo do mosquito em seu celular, primeiramente precisamos instalar o aplicativo do MIT.

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



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.

Conectar ao Assistente

Launch the MIT AI2 Companion on your device and then scan the barcode or type in the code to connect for live testing of your app.
[Need help finding the Companion App?](#)



Seu código é:

eytmlh

Cancelar

MIT App Inventor 2

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

Six Character Code

connect with code

scan QR code

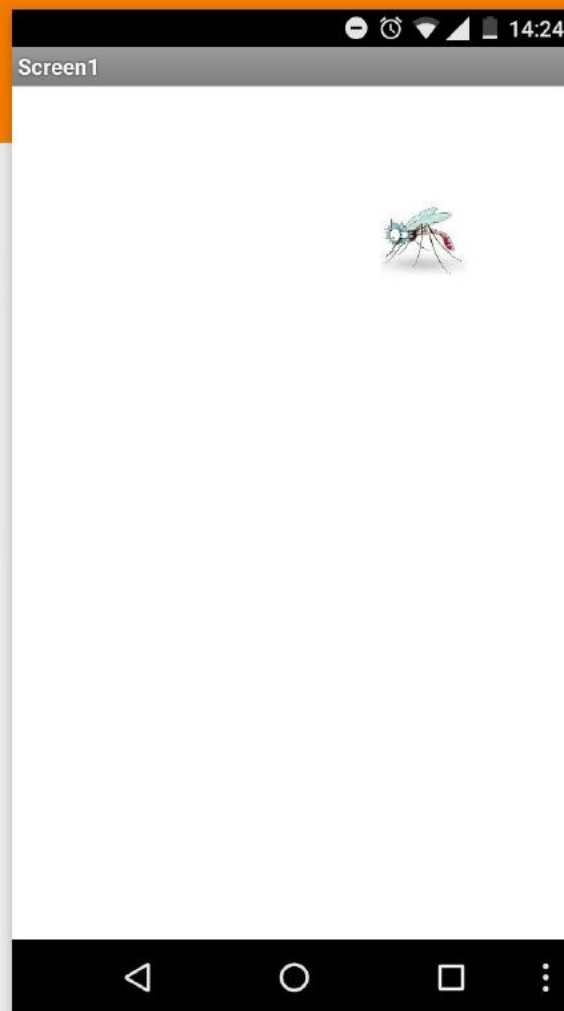
DICA

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.

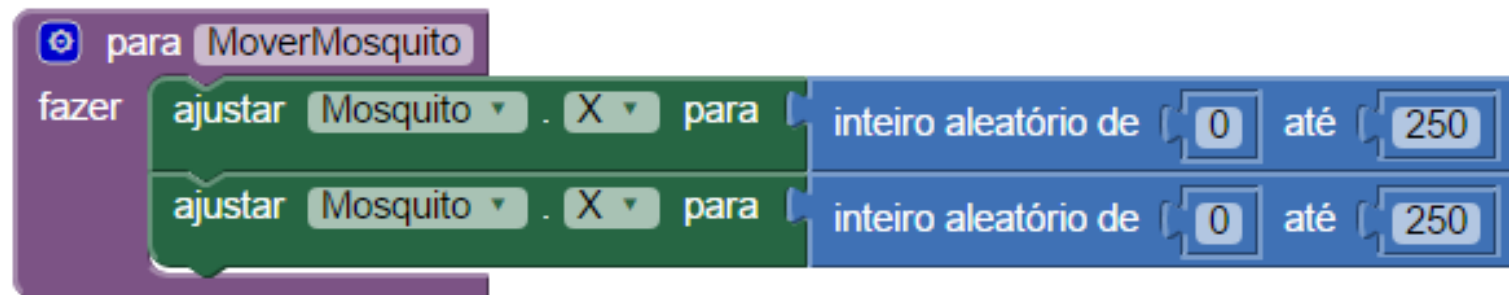


MOVER O MOSQUITO **PELA TELA**

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!



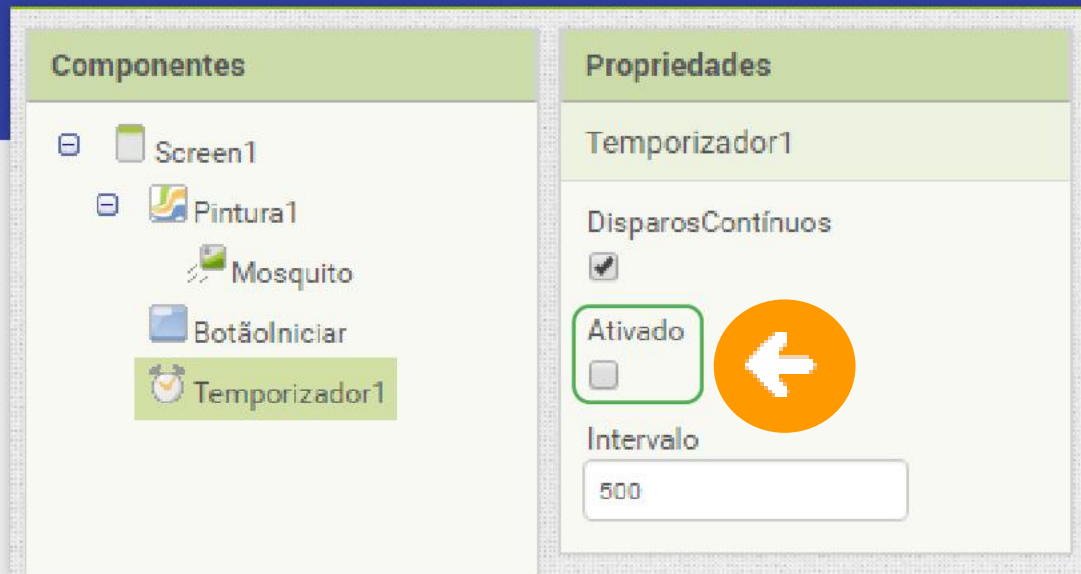
INICIAR O JOGO



INICIAR O JOGO

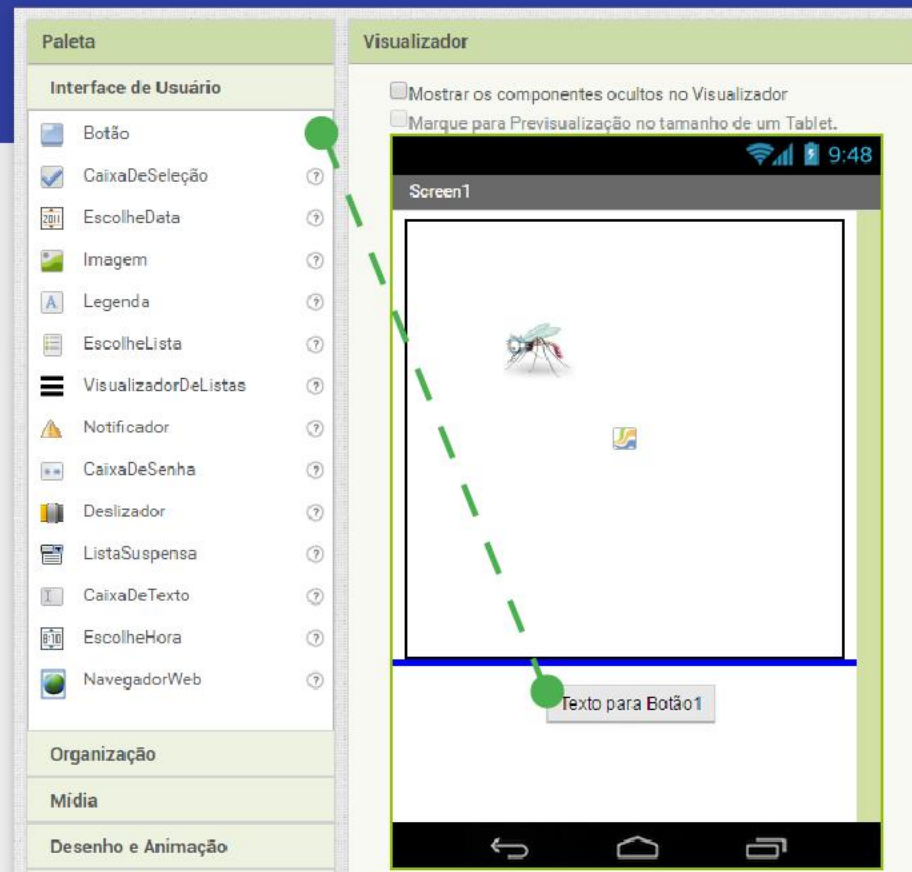
Primeiro, vamos voltar para a tela de “Designer” e desativar o **Temporizador1**. Assim, ele não vai mais disparar o movimento do mosquito até que o botão **Iniciar Jogo** seja clicado.

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**.



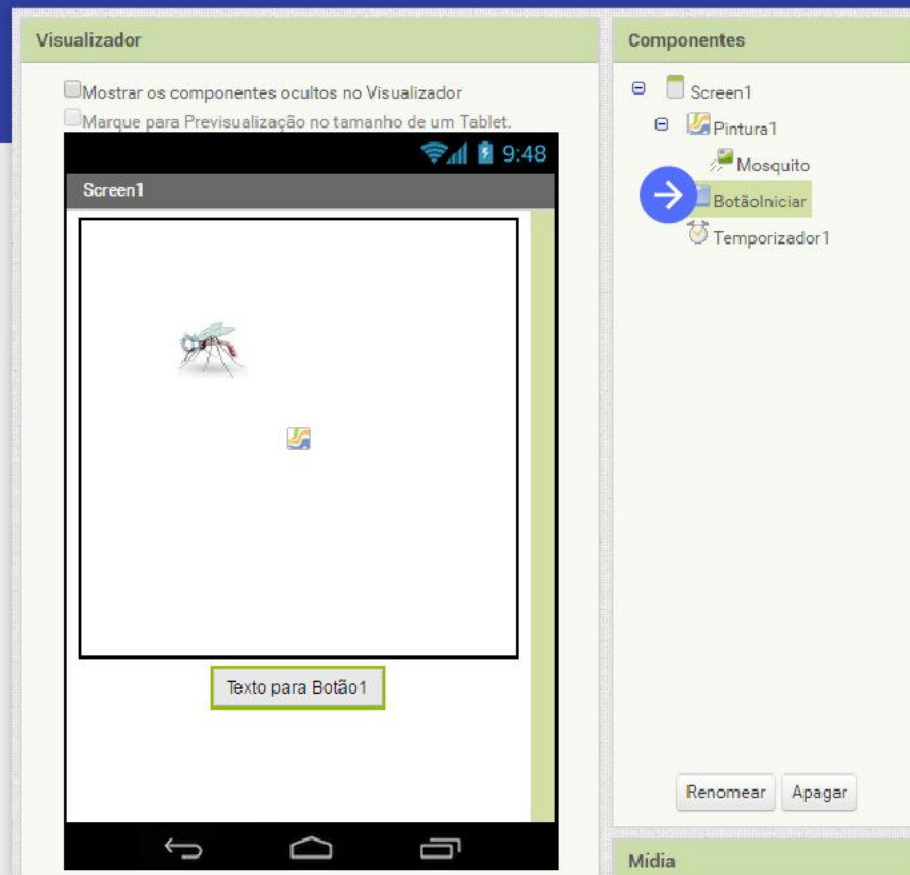
INICIAR O JOGO

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



INICIAR O JOGO

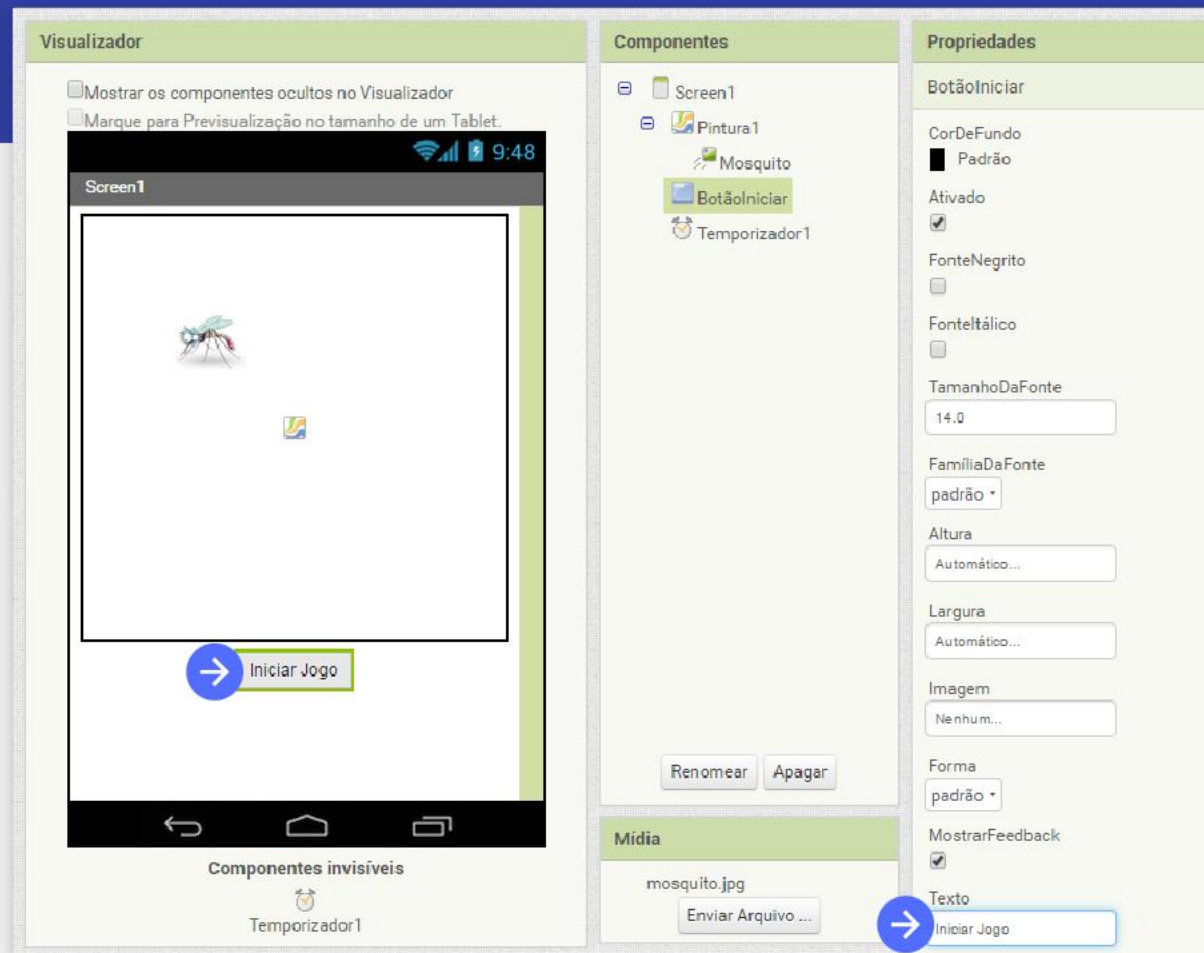
Agora, da mesma maneira que renomeamos o **Mosquito**, vamos renomear o componente -**Botao1** para **BotaoIniciar**.



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!

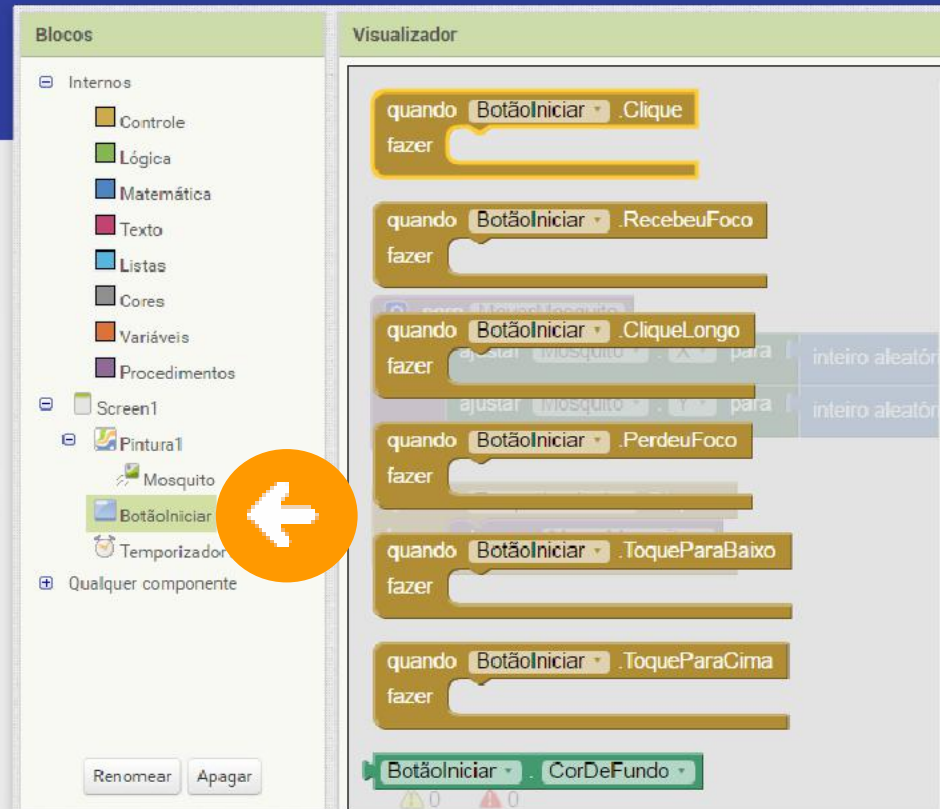


INICIAR O JOGO

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 IDE interface with two panels: 'Blocos' (Blocks) and 'Visualizador' (Viewer).

Blocos Panel:

- Internos**
 - Controle
 - Lógica
 - Matemática
 - Texto
 - Listas
 - Cores
 - Variáveis
 - Procedimentos
- Screen1**
 - Pintura1
 - Mosquito
 - BotãoIniciar
 - Temporizador1** (highlighted with a blue arrow)
- Qualquer componente

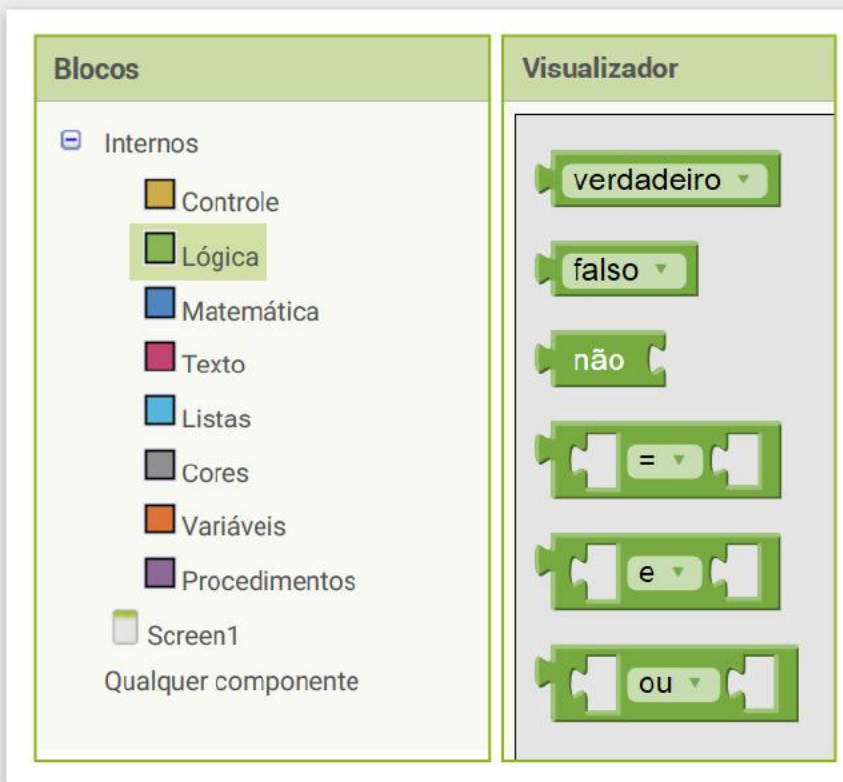
Visualizador Panel:

- para** MoverMosquito
 - fazer**
 - ajustar Mosquito . X para inteiro aleatório
 - ajustar Mosquito . Y para inteiro aleatório
- quando** Temporizador1 . Disparo
 - fazer** chamar MoverMosquito
- quando** BotãoIniciar . Clique
 - fazer** ajustar Temporizador1 . Ativado para inteiro aleatório

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 MILLISEGUNDOS

ajustar Temporizador1 . Ativado para verdadeiro

SE A VARIÁVEL ATIVADO TIVER VALOR VERDADEIRO, VAI DISPARAR!

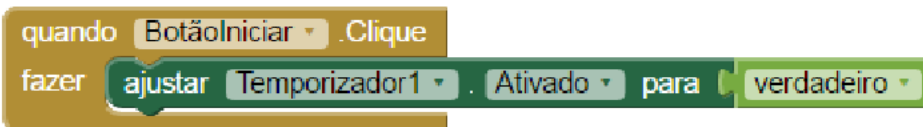
ajustar Temporizador1 . Ativado para falso

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:



INICIAR O JOGO

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



VIDAS DO **MOSQUITO**

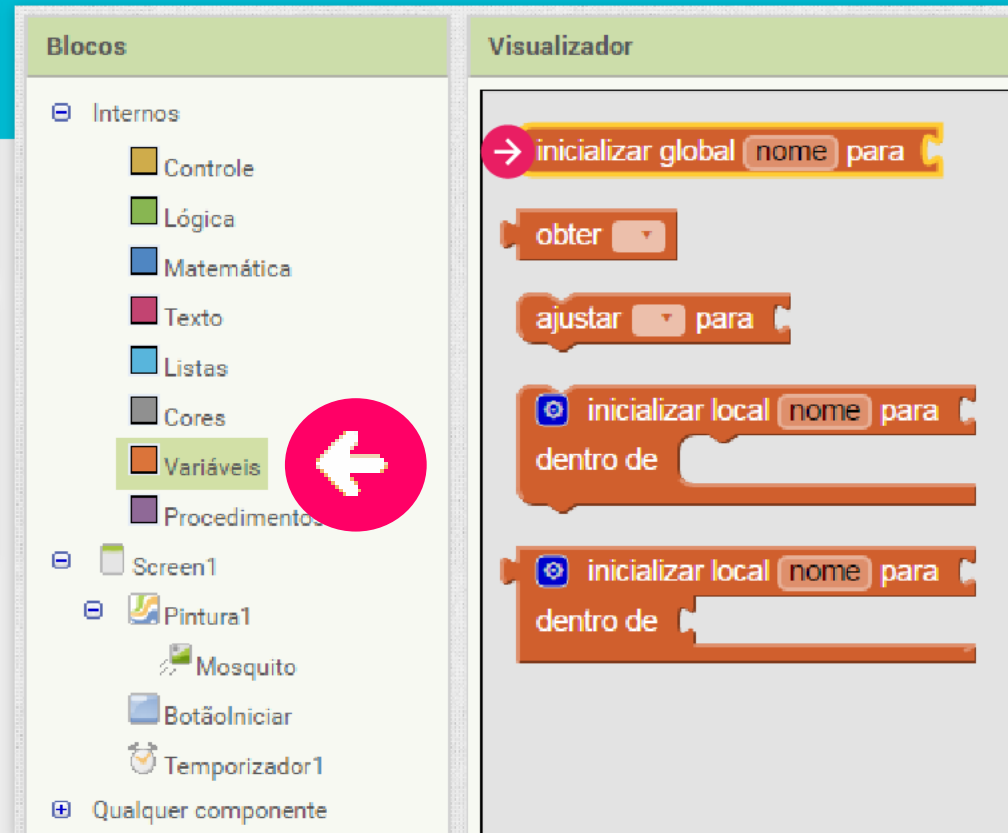
Agora vamos adicionar vida ao Mosquito.
Ele vai iniciar o jogo com 3 vidas.

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".



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).

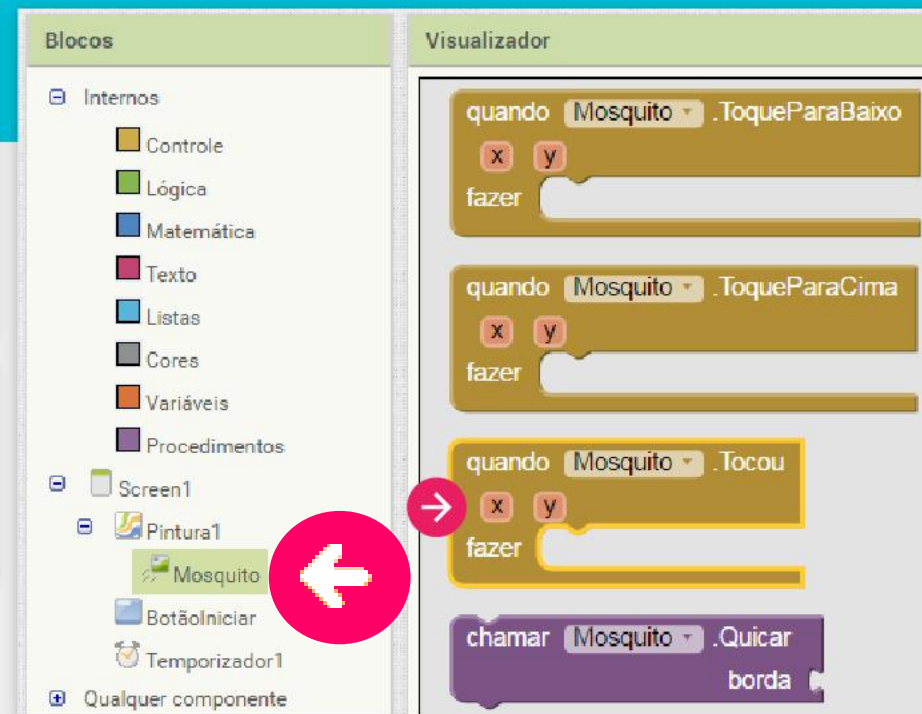
Sendo assim, basta arrastarmos ele para a área do Visualizador e colocar o valor **Matemático** "3" junto a ele.

The screenshot shows the Scratch IDE interface. On the left, the 'Blocos' (Blocks) palette is visible, with the 'Matemática' (Math) category selected. In the center, the 'Visualizador' (Inspector) panel shows the 'iniciar global' block, which is a red block with a white arrow pointing right. The block is labeled 'vidaRestante' and has a value of '3' next to it. The 'Visualizador' panel also shows other blocks, including 'para MoverMosquito' and 'quando BotãoIniciar .Clique'.

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.

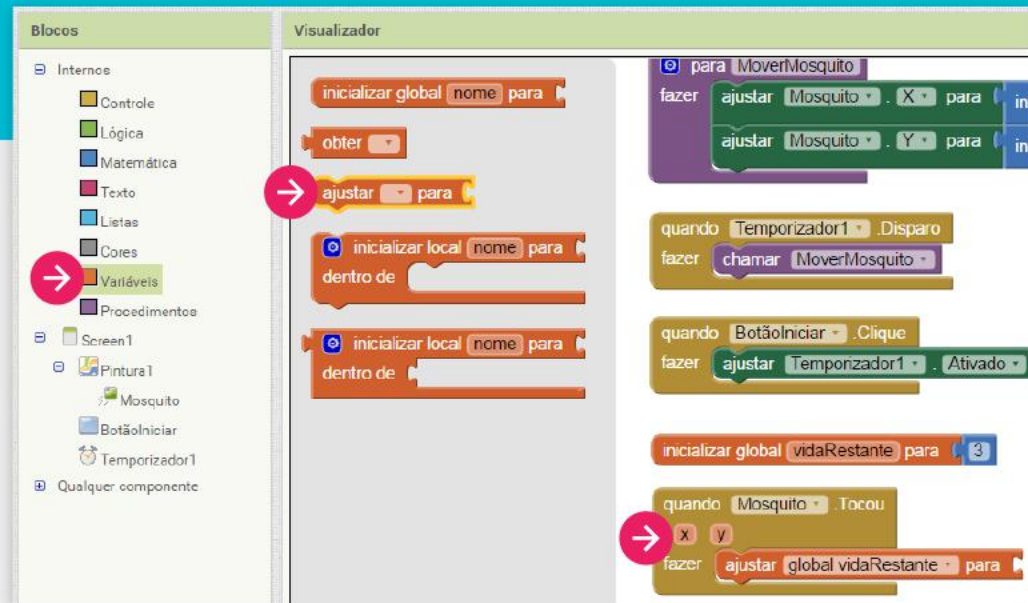


VIDAS DO MOSQUITO

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

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

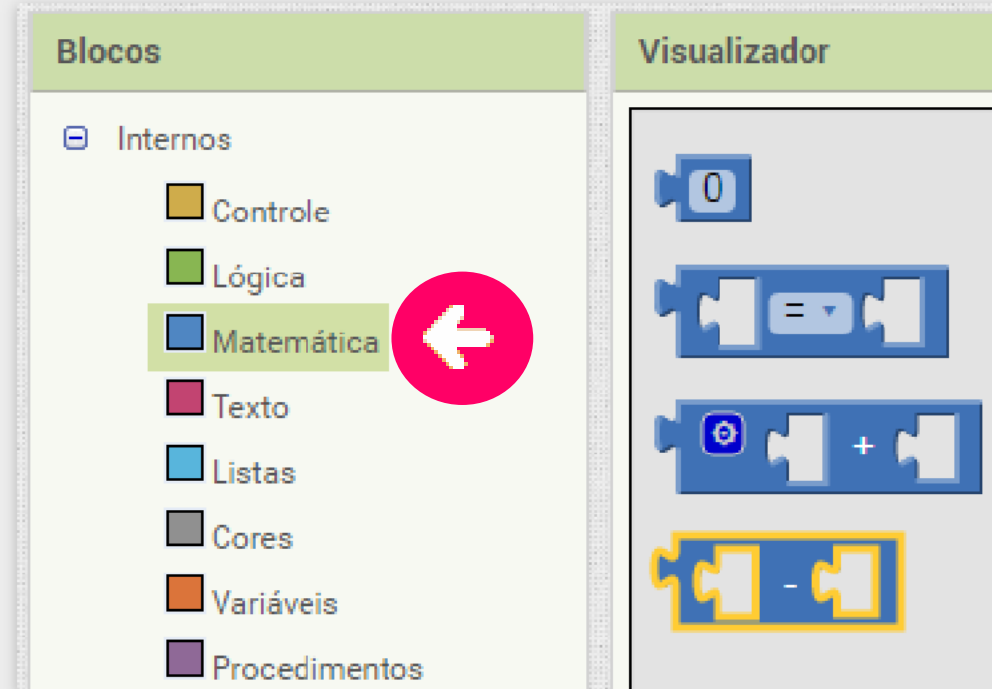
Escolha a variável “**global vidaRestante**” na lista 



VIDAS DO MOSQUITO

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

Para isso, escolha o bloco Matemático de Subtração.



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).

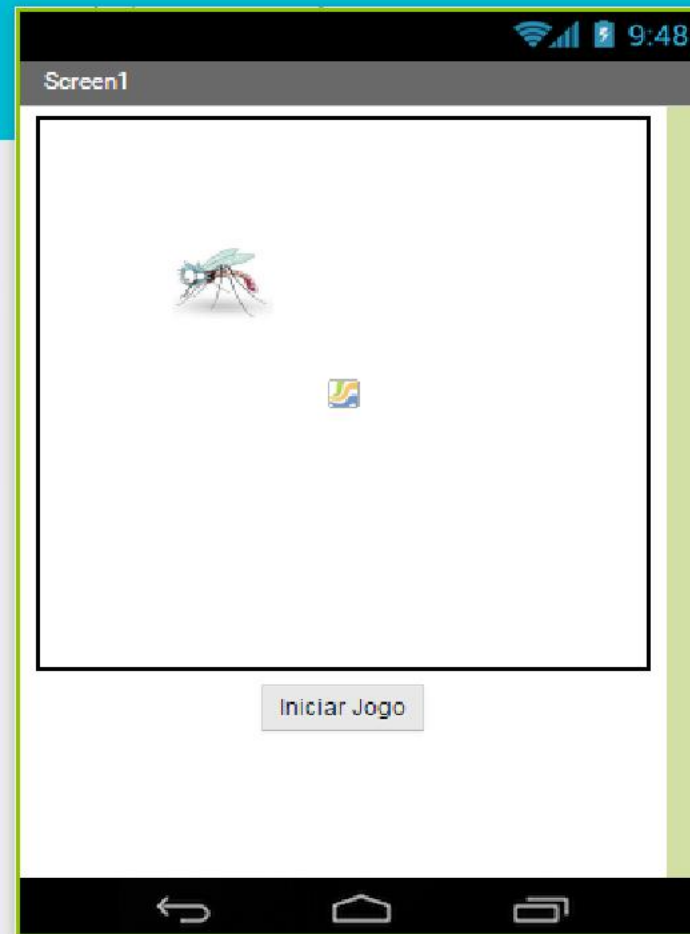
O procedimento vai ficar assim:

The image displays the Scratch programming environment. The 'Blocos' (Blocks) panel on the left shows the 'Variáveis' (Variables) category highlighted with a red arrow. The 'Visualizador' (Inspector) panel on the right shows a script with three blocks: 'inicializar global nome para', 'obter', and 'ajustar para'. Below this, a completed script is shown: 'quando Mosquito .Tocou' followed by a 'fazer' loop containing 'ajustar global vidaRestante para' and 'obter global vidaRestante' with a value of 1.

VIDAS DO MOSQUITO

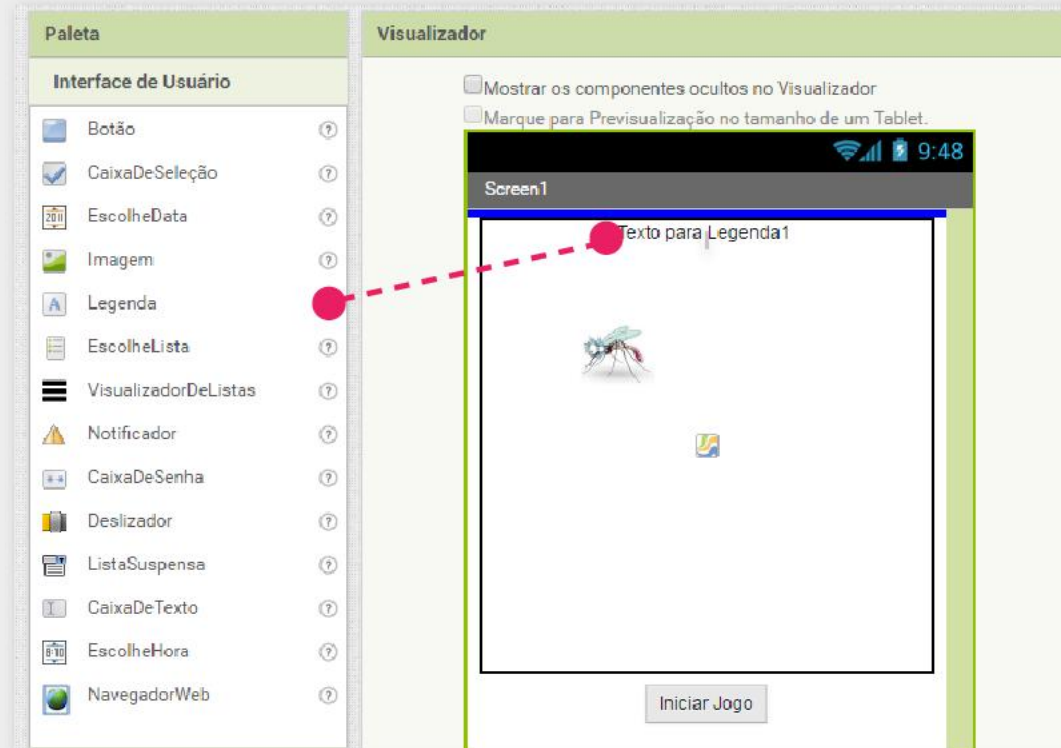
O nosso código de vida já está funcionando. Mas note que na tela do jogo não existe nada indicando para o jogador quanto é a vida do mosquito.

Precisamos mostrar o valor da vida na tela.



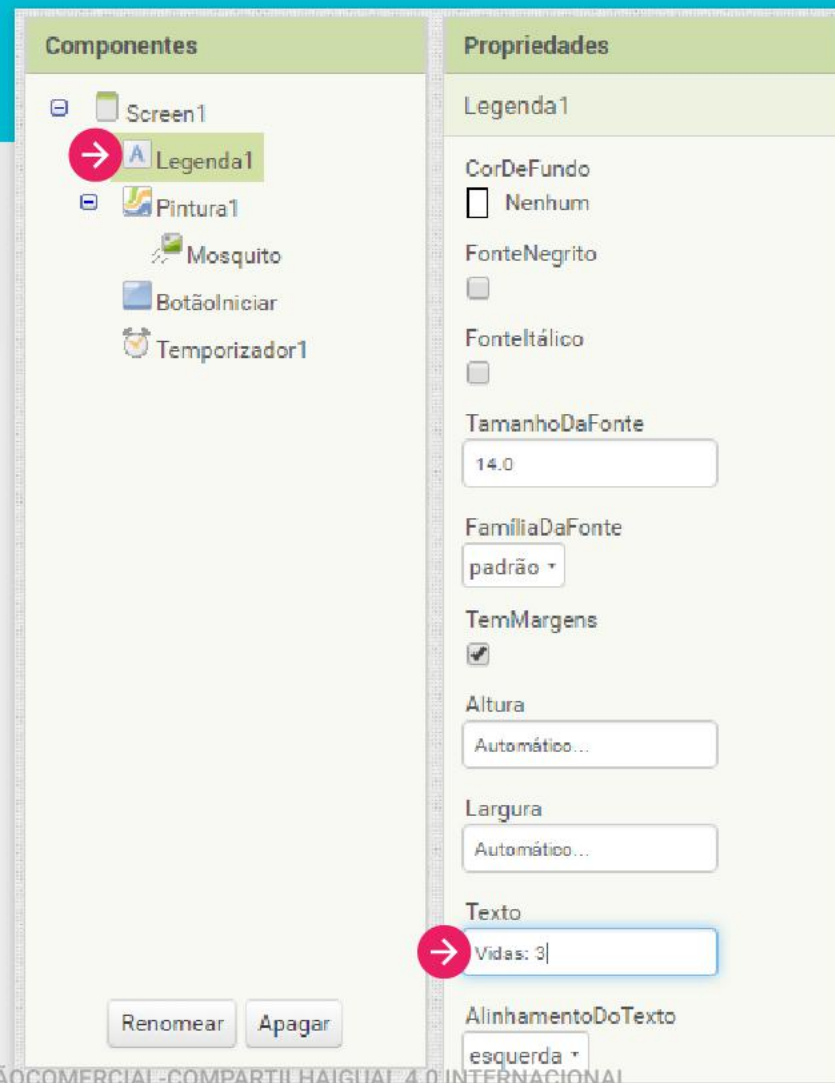
VIDAS DO MOSQUITO

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!



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 the Scratch IDE interface. On the left, the 'Blocos' (Blocks) panel is open, displaying a list of categories: Internos, Controle, Lógica, Matemática, Texto, Listas, Cores, Variáveis, and Procedimentos. The 'Texto' category is selected, and a red arrow points to the 'Legenda1' component under the 'Screen1' category. On the right, the 'Visualizador' (Inspector) panel is open, showing a list of components: Legenda1, Pintura1, Mosquito, BotãoIniciar, and Temporizador1. The 'Legenda1' component is selected, and a red arrow points to the 'ajustar Legenda1.Texto' block in the 'Visualizador' panel. The 'ajustar Legenda1.Texto' block is highlighted with a yellow border, and a red arrow points to it from the 'Legenda1' component in the 'Visualizador' panel.

STRINGS

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

STRING NOME



STRING SOBRENOME



JUNTAR



“Vidas:” JUNTAR valor da variável vidaRestante

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**.

The image shows the Scratch IDE interface. On the left, the 'Blocos' (Blocks) panel is open, with the 'Text' (Texto) category selected. On the right, the 'Visualizador' (Viewer) panel shows the 'AtualizarVidaRestante' script. The script contains a 'para' (for) loop that uses the 'ajustar' (set) block to update the 'Legenda1' text with the 'vidaRestante' variable value using the 'juntar' (join) block. The 'juntar' block is highlighted with a red arrow, indicating its use in combining the text 'Vida: ' with the variable value.

Blocos

- Internos
 - Controle
 - Lógica
 - Matemática
 - Texto
 - Listas
 - Cores
 - Variáveis
 - Procedimentos
- Screen1
 - Legenda1
 - Pintura1

Visualizador

```
inicializar global vidaRestante para 3

quando Mosquito .Tocou
  x y
  fazer
    ajustar global vidaRestante para obter global vida

para AtualizarVidaRestante
  fazer
    ajustar Legenda1 . Texto para 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.

The image shows a Scratch code editor with the 'Visualizador' (Viewer) tab selected. The 'Blocos' (Blocks) palette on the left shows the 'Procedimentos' (Procedures) category highlighted. The 'Visualizador' area displays the following code:

- quando BotãoIniciar .Clique**
 - fazer **ajustar Temporizador1 . Ativado para verdadeiro**
- inicializar global vidaRestante para 3**
- quando Mosquito .Tocou**
 - fazer**
 - ajustar global vidaRestante para obter global vidaRestante - 1**
 - chamar AtualizarVidaRestante**
- para AtualizarVidaRestante**
 - fazer **ajustar Legenda1 . Texto para juntar (" Vida: " obter global vidaRestante)**

The 'AtualizarVidaRestante' procedure is called from the 'quando Mosquito .Tocou' event block and the 'para AtualizarVidaRestante' loop block.

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!

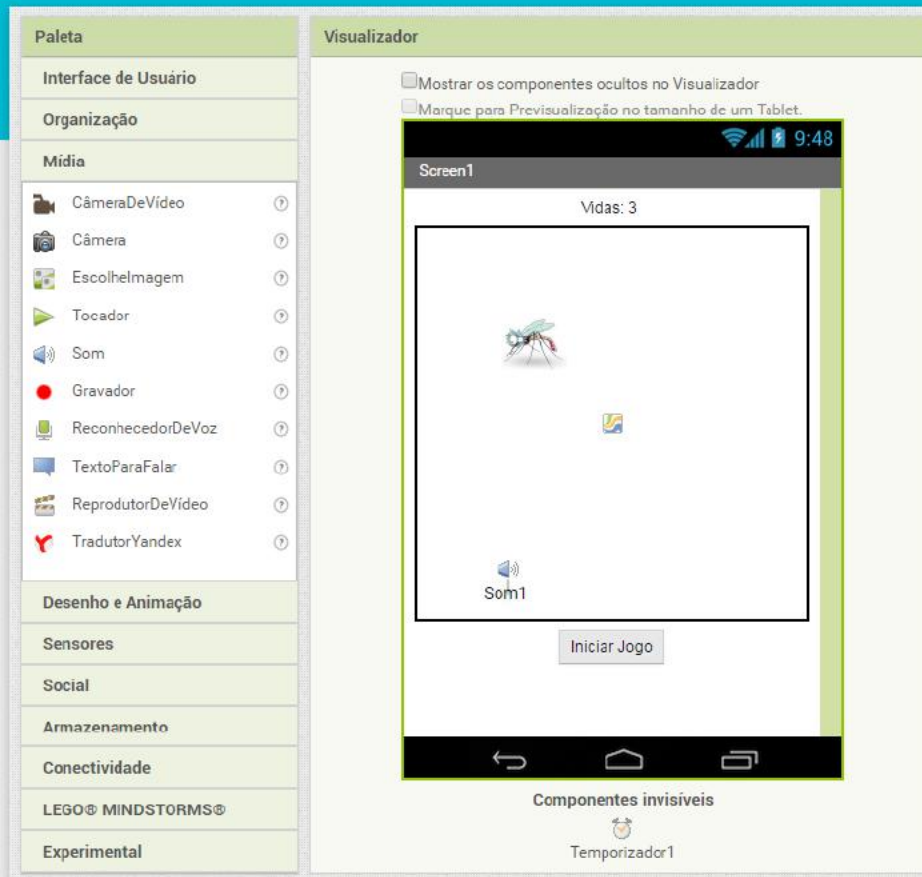
VIDAS DO MOSQUITO

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



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 the Scratch interface with two panels: 'Blocos' (Blocks) and 'Visualizador' (Viewer).

Blocos (Blocks) Panel:

- Internos
 - Controle
 - Lógica
 - Matemática
 - Texto
 - Listas
 - Cores
 - Variáveis
 - Procedimentos
- Screen1
 - Legenda1
- Pintura1
 - Mosquito
- BotãoIniciar
- Temporizador1
- Som1** (highlighted with a red arrow)
- Qualquer componente

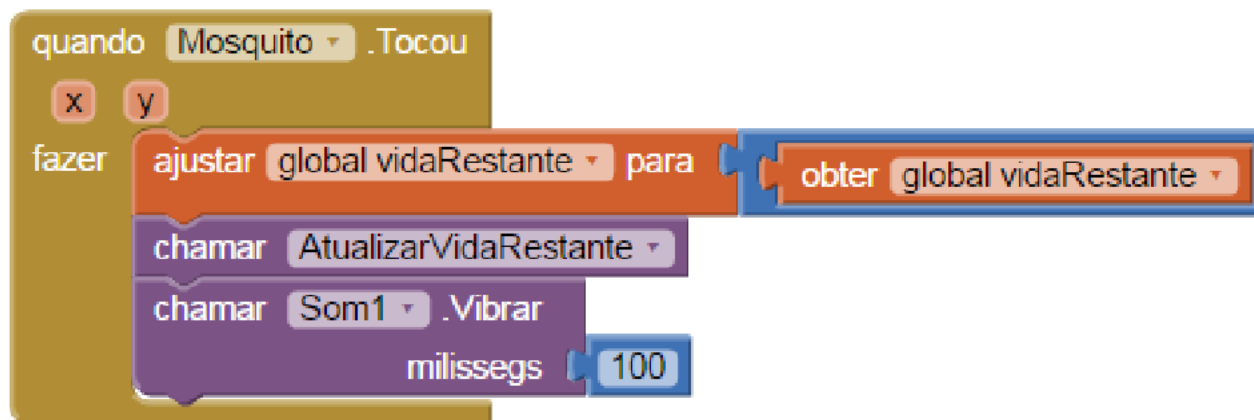
Visualizador (Viewer) Panel:

The script in the Visualizador panel is as follows:

```
quando Som1 .undefined
  mensagem
  fazer
    chamar Som1 .Parar
    chamar Som1 .Tocar
    chamar Som1 .Retomar
    chamar Som1 .Parar
    chamar Som1 .Vibrar milisegs (highlighted with a red arrow)
    Som1 . IntervaloMinimo
    ajustar Som1 . IntervaloMinimo para
```

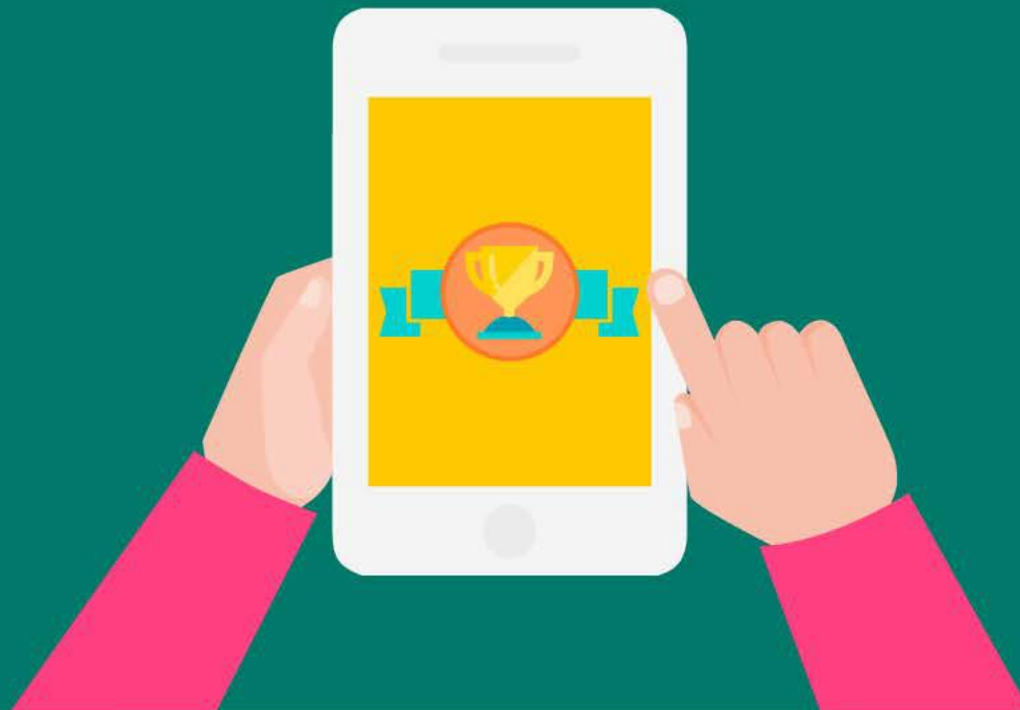
VIDAS DO MOSQUITO

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



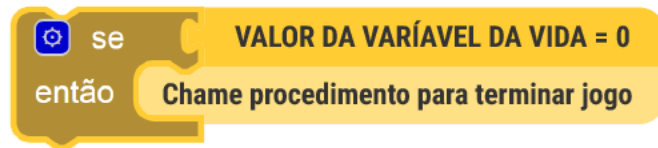
FALSO

CONTINUAR O JOGO



VERDADEIRO

TERMINAR O 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”.

The screenshot displays the Scratch IDE interface. On the left, the 'Blocos' (Blocks) palette is visible, with the 'Lógica' (Logic) category highlighted by a red circle and a white arrow pointing to the 'igual' (equal) block. The 'Visualizador' (Viewer) area on the right shows the following code:

```
inicializar global vidaRestante para 3

quando Mosquito .Tocou
  fazer
    ajustar global vidaRestante para obter global vidaRestante - 1
    chamar AtualizarVidaRestante
    chamar Som1 .Vibrar milisegs 100
    se obter global vidaRestante = 0
      então
        para AtualizarVidaRestante
          fazer
            ajustar Legenda1 .Texto para juntar " Vida: " obter global vidaRestante
```

The code implements a health system where the global variable 'vidaRestante' is initialized to 3. When the mosquito touches the player, the health is decreased by 1, a sound is played, and a conditional check is performed. If the health reaches 0, the 'para AtualizarVidaRestante' loop is triggered, which updates the 'Legenda1' text to show the remaining health.

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 Temporizador1.Ativado” informando o valor “falso”.

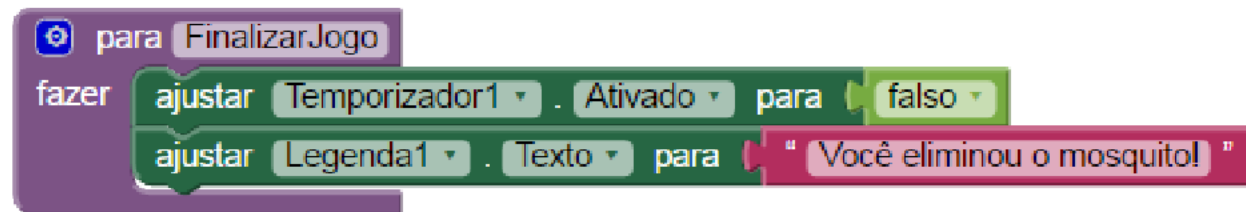


FINALIZAR O JOGO

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.

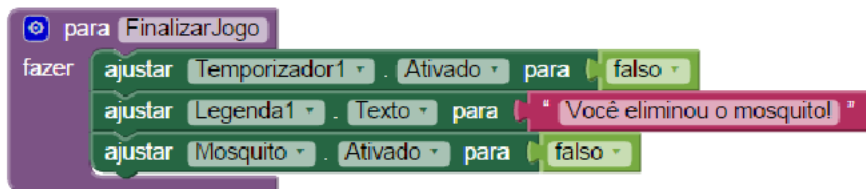


FINALIZAR O JOGO

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”.

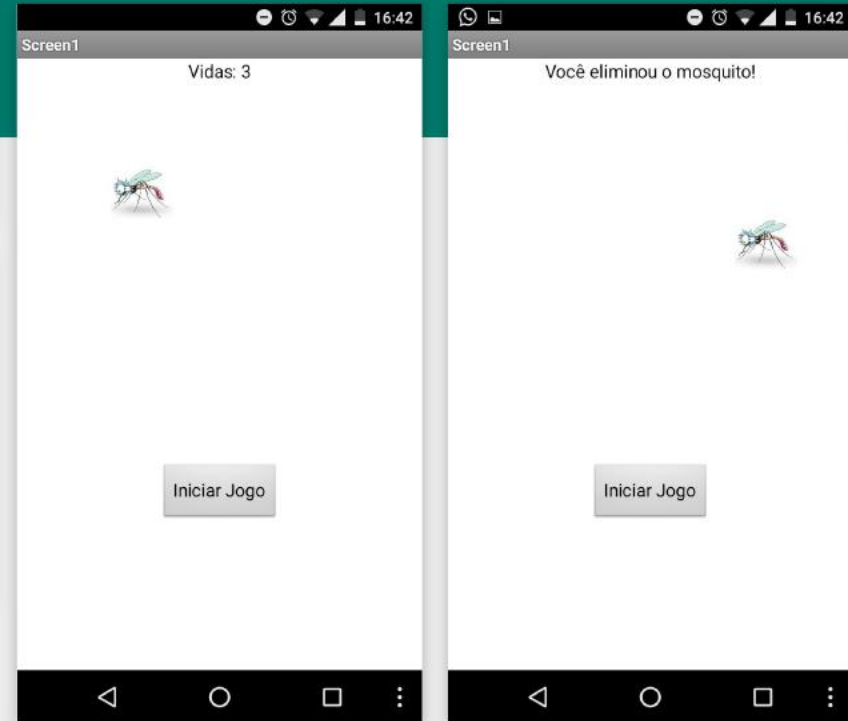
The image shows the Scratch development environment. On the left, the 'Blocos' (Blocks) palette is visible, with the 'Procedimentos' (Behaviors) category selected. A red circle with a white arrow points from the 'FinalizarJogo' block in the 'Procedimentos' category to the 'chamar FinalizarJogo' block in the 'Visualizador' (Stage) area. The 'Visualizador' area shows the 'Mosquito' object's script area. The script starts with 'inicializar global vidaRestante para 3'. Then, a 'quando Mosquito . Tocou' (when clicked) event triggers a 'fazer' (do) loop. Inside the loop, there are blocks for 'ajustar global vidaRestante para obter global vidaRestante' (set global life remaining to current value), 'chamar AtualizarVidaRestante' (call update life remaining), 'chamar Som1 . Vibrar milisegs 100' (play sound 1 for 100 ms), and a conditional block: 'se obter global vidaRestante = 0 então chamar FinalizarJogo' (if life remaining is 0, then call finish game). A red circle with a white arrow points to the 'FinalizarJogo' block within this conditional statement.

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 recomendar 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”.

The image shows the Scratch development environment. On the left, the 'Blocos' (Blocks) panel is open, displaying a tree view of the project's internal structure. Under the 'Internos' (Internal) category, the 'Procedimentos' (Procedures) block is highlighted. A red circle with a white arrow points from this block to the 'Visualizador' (Viewer) panel on the right. The 'Visualizador' panel shows the code for the 'AtualizarVidaRestante' procedure. The code consists of three blocks: a 'quando' (when) block, an 'ajustar' (set) block, and a 'chamar' (call) block. The 'quando' block is triggered by 'Temporizador1' (Timer1) and the 'Disparo' (Fire) event. The 'ajustar' block sets the 'global vidaRestante' (global remaining life) to the value 3. The 'chamar' block calls the 'AtualizarVidaRestante' procedure. The 'Visualizador' panel also shows the code for the 'BotãoIniciar' (Start Button) click event, which sets the 'Temporizador1' (Timer1) to 'Ativado' (Active) and calls the 'AtualizarVidaRestante' procedure. The 'inicializar global vidaRestante' (initialize global remaining life) block is also visible, setting the value to 3.

Blocos

- Internos
 - Controle
 - Lógica
 - Matemática
 - Texto
 - Listas
 - Cores
 - Variáveis
 - Procedimentos**
- Screen1
 - Legenda1
- Pintura1

Visualizador

quando Temporizador1 .Disparo
fazer chamar MoverMosquito

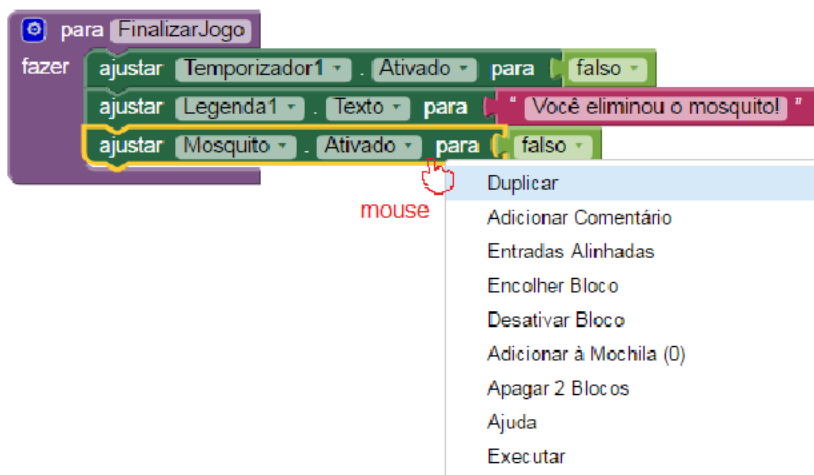
quando BotãoIniciar .Clique
fazer ajustar Temporizador1 .Ativado para verdadeiro
ajustar global vidaRestante para 3
chamar AtualizarVidaRestante

inicializar global vidaRestante para 3

REINICIANDO O JOGO

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

- a) Para ativar o mosquito, podemos usar o bloco "ajustar Mosquito.Ativado" com o valor "Verdadeiro".
- b) 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".
- c) 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!

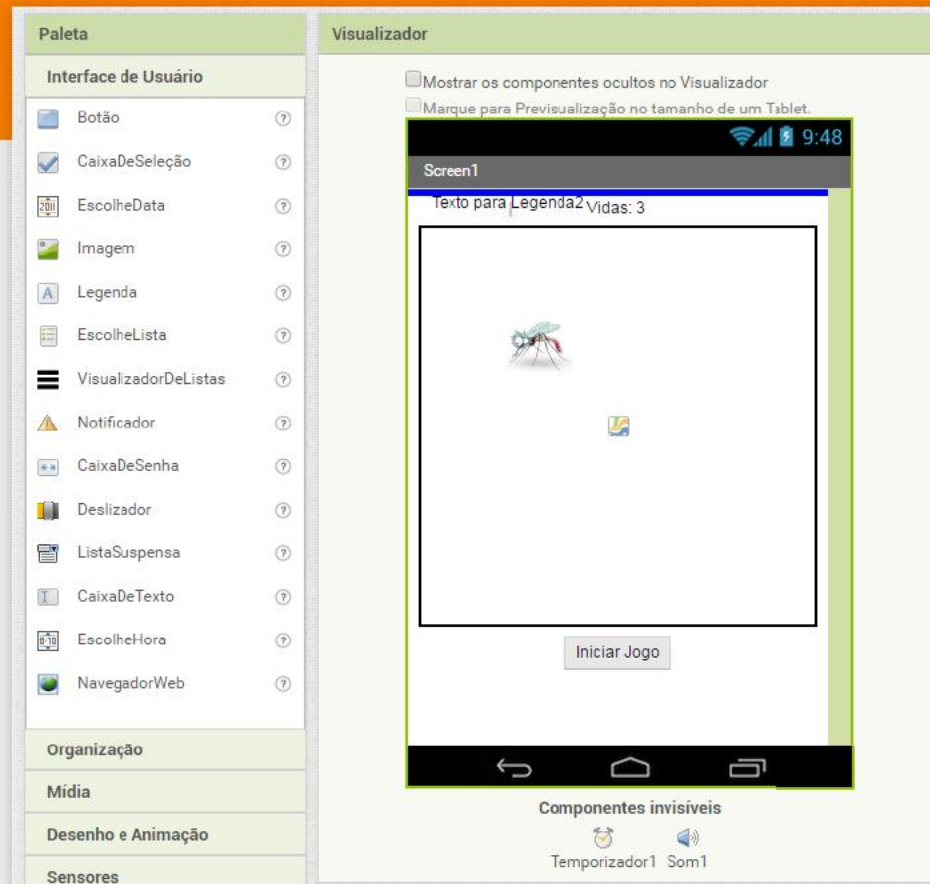
Você pode notar que o jogador ainda não possui nenhuma maneira de ser derrotado.

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”.

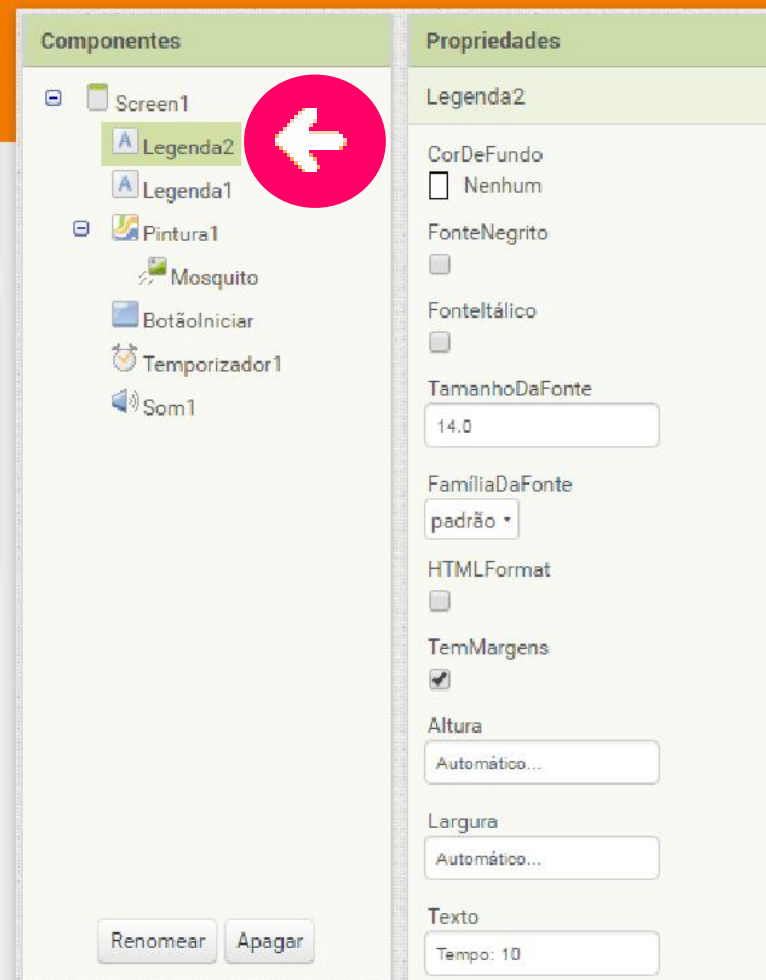
Vamos adicionar um novo componente Legenda para mostrar o tempo restante, preferencialmente acima da **Legenda1** (da vida).



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 milissegundos), vamos deixá-lo desativado por enquanto.

The screenshot displays the Scratch IDE interface with four main panels: Paleta, Visualizador, Componentes, and Propriedades.

- Paleta:** The 'Sensores' (Sensors) category is selected. The 'Temporizador' (Timer) component is highlighted with a black arrow.
- Visualizador:** The stage area shows a game interface with a mosquito icon, a score of 3, and a timer showing 'Tempo: 10'. A black arrow points from the 'Temporizador' component in the Paleta to the 'Temporizador2' component on the stage.
- Componentes:** The 'Componentes' panel lists various components. 'Temporizador2' is highlighted with a green box. A red arrow points from 'Temporizador2' to the 'Propriedades' panel.
- Propriedades:** The 'Propriedades' panel shows the configuration for 'Temporizador2'. The 'DisparosContínuos' (Continuous Shots) checkbox is checked. The 'Intervalo' (Interval) is set to 1000 milliseconds. The 'Ativado' (Activated) checkbox is unchecked.

At the bottom of the interface, the 'Componentes invisíveis' (Invisible Components) area shows 'Temporizador1', 'Som1', and 'Temporizador2'.

ADICIONANDO TEMPO

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



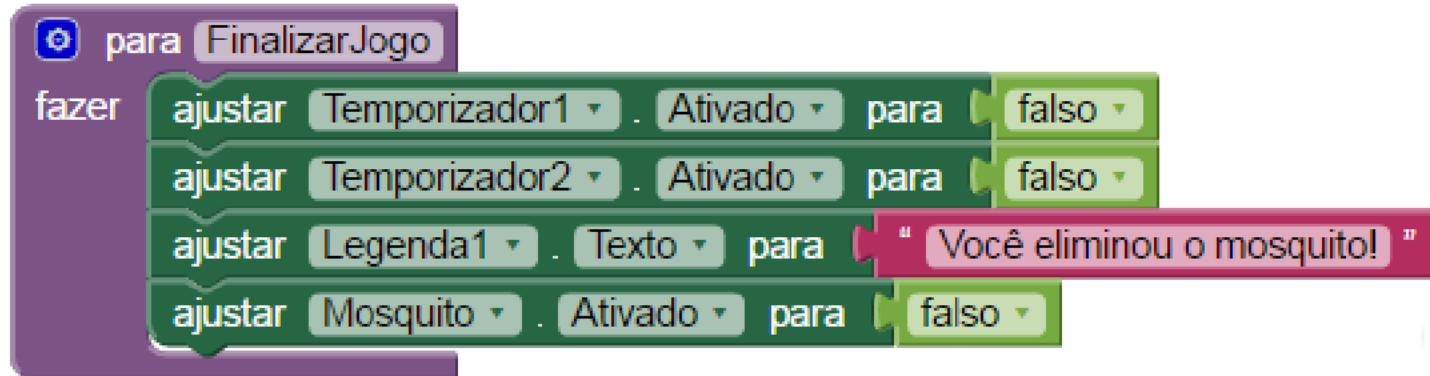
Primeiramente, vamos ativar o nosso **Temporizador2** quando o jogo for iniciado, incluindo o bloco **ajustar Temporizador2.Ativado** com valor Verdadeiro dentro do procedimento quando **Botaoiniciar.Clique**.

DICA

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

ADICIONANDO **TEMPO**

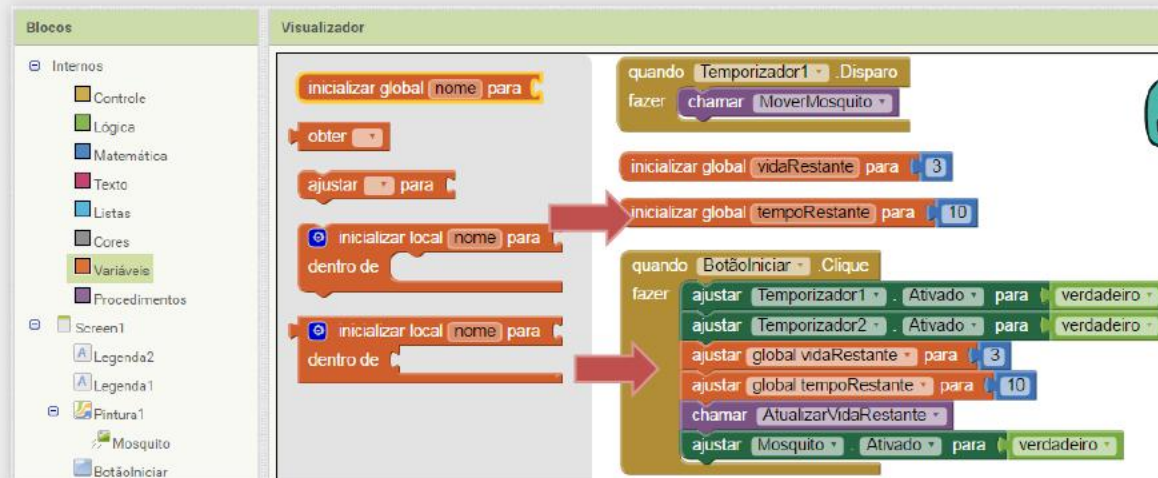
Também precisamos lembrar de desativar o **Temporizador2** quando o jogo for finalizado.



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

ADICIONANDO TEMPO

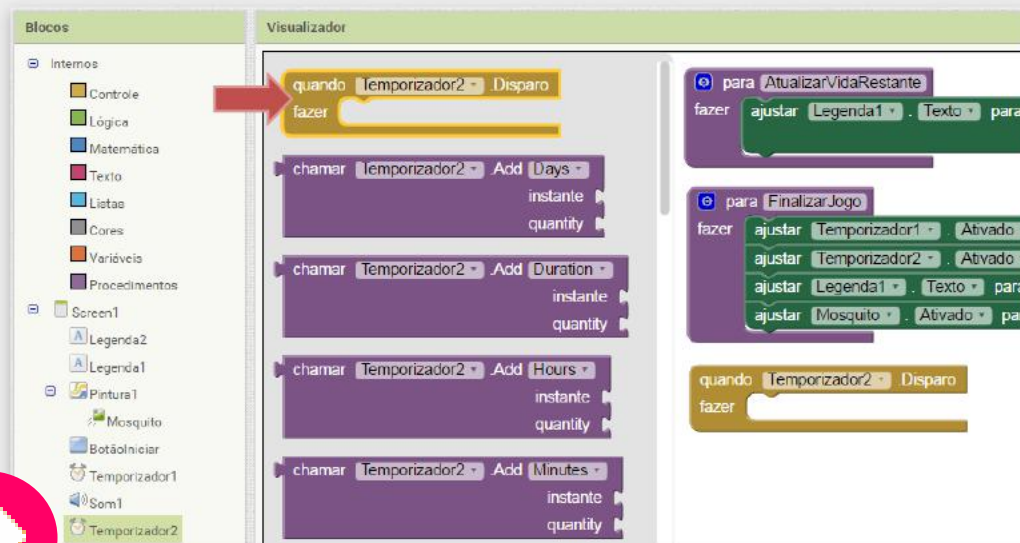
Agora precisamos criar uma variável global, responsável por armazenar o tempo restante de jogo. Vamos iniciá-la com o valor "10".



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.Disparar”.

ADICIONANDO TEMPO

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



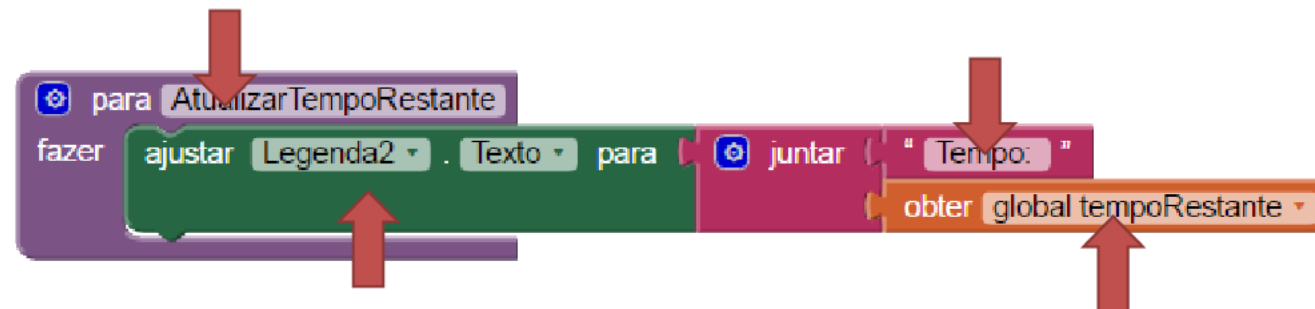
Usamos o bloco “ajustar global tempoRestante”, os blocos Matemáticos de Subtração e número inteiro “1” e o bloco “obter globalRestante”.

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 Botaolniciar.Clique”, para mostrar o tempo correto quando o jogo iniciar.

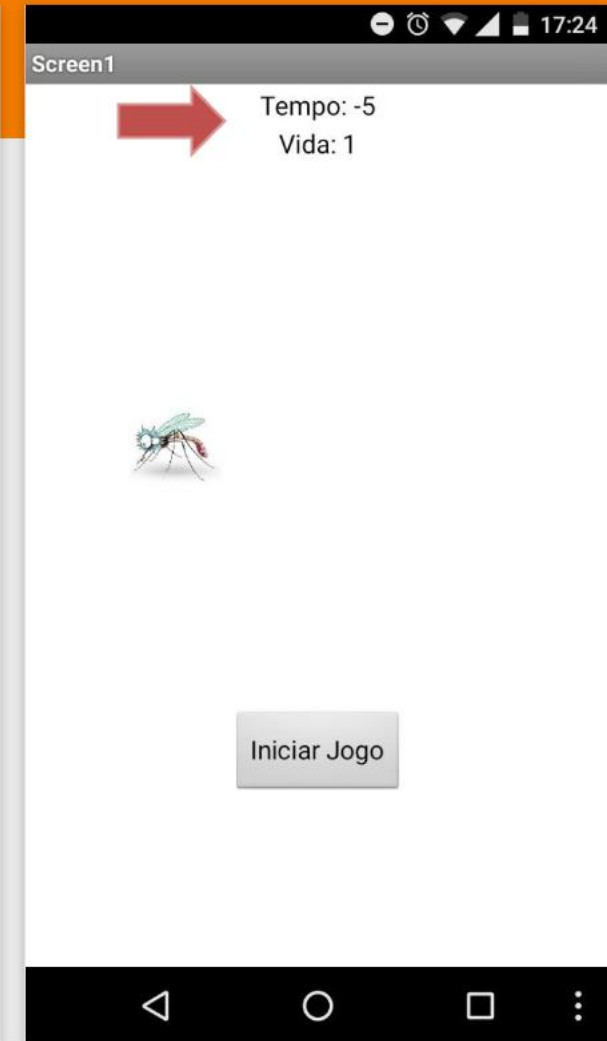
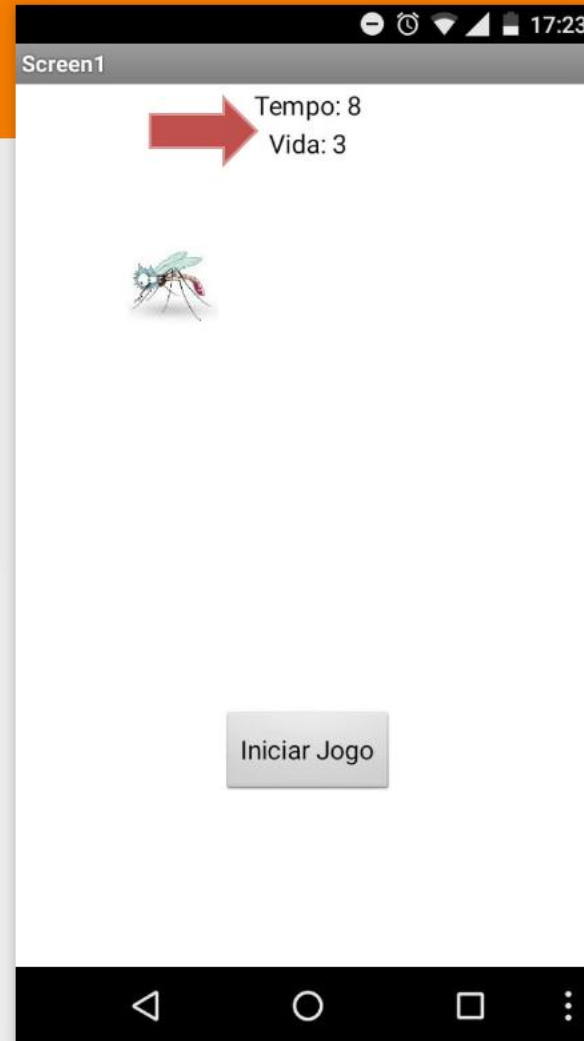


ADICIONANDO **TEMPO**

VAMOS TESTAR!

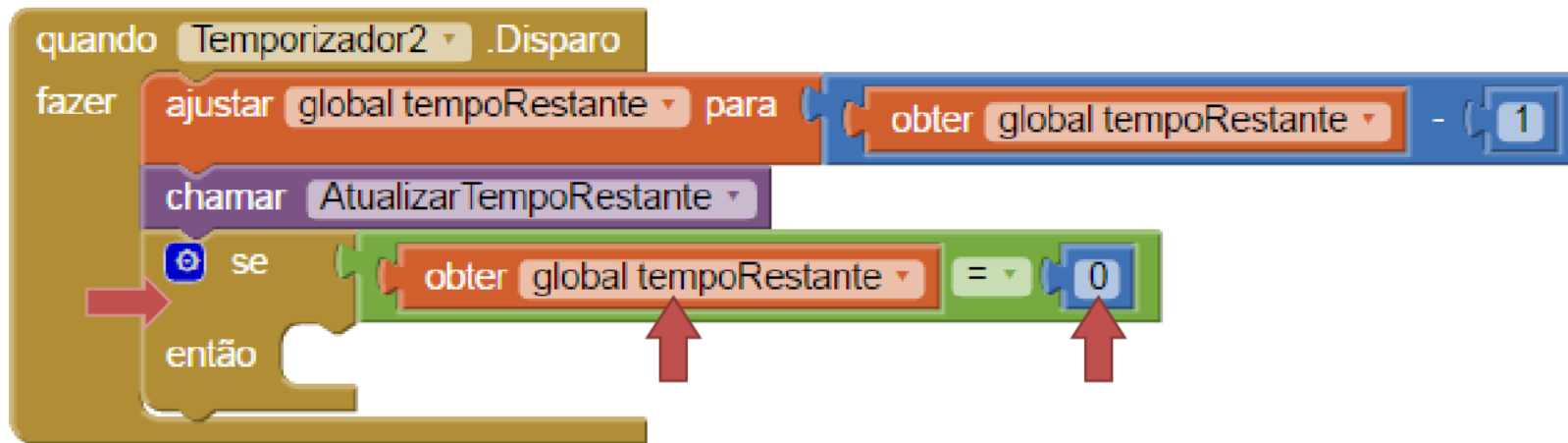
Verifique se o tempo está diminuindo na tela e se está se comportando corretamente.

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

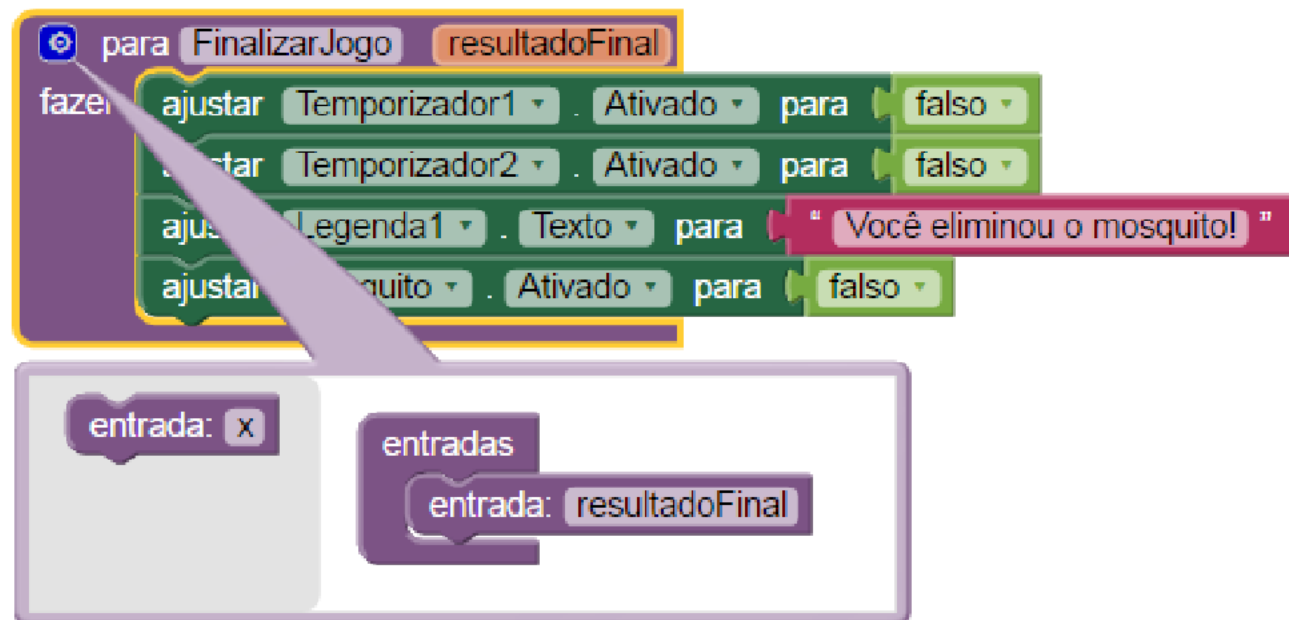
Temos que modificar o procedimento “FinalizarJogo” para receber a informação se o jogador ganhou ou perdeu, e mudar o texto da **Legenda1** corretamente.

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

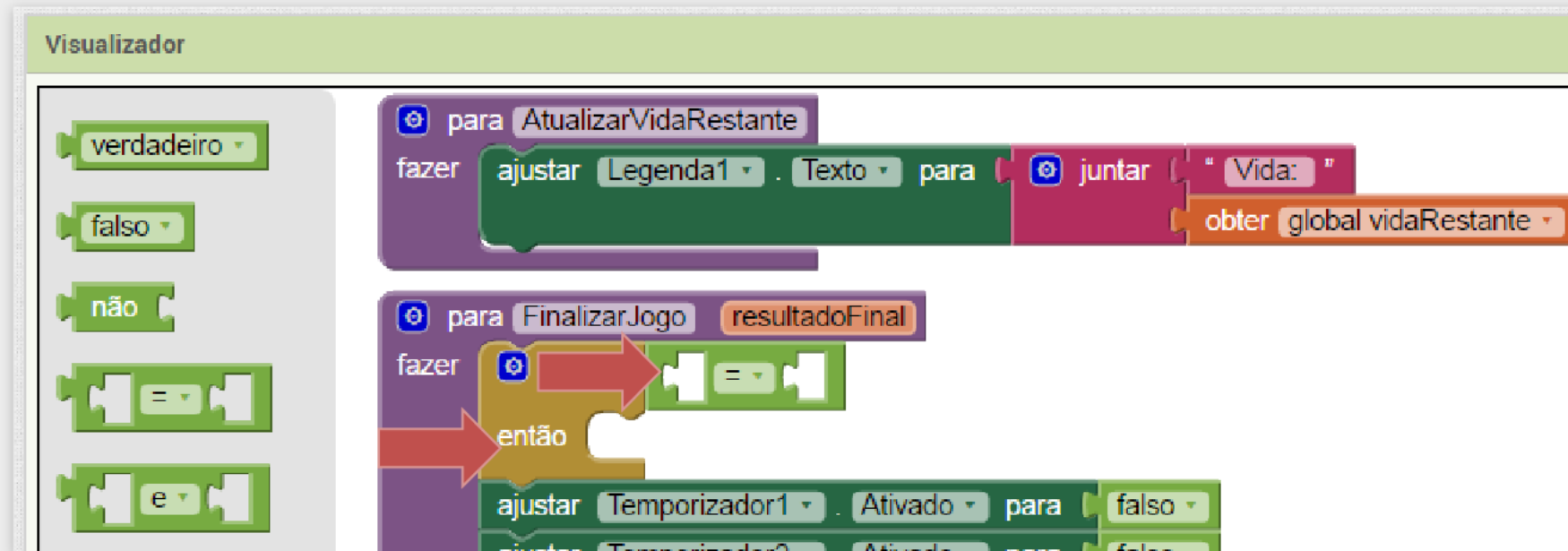
DEVE FICAR ASSIM:



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.

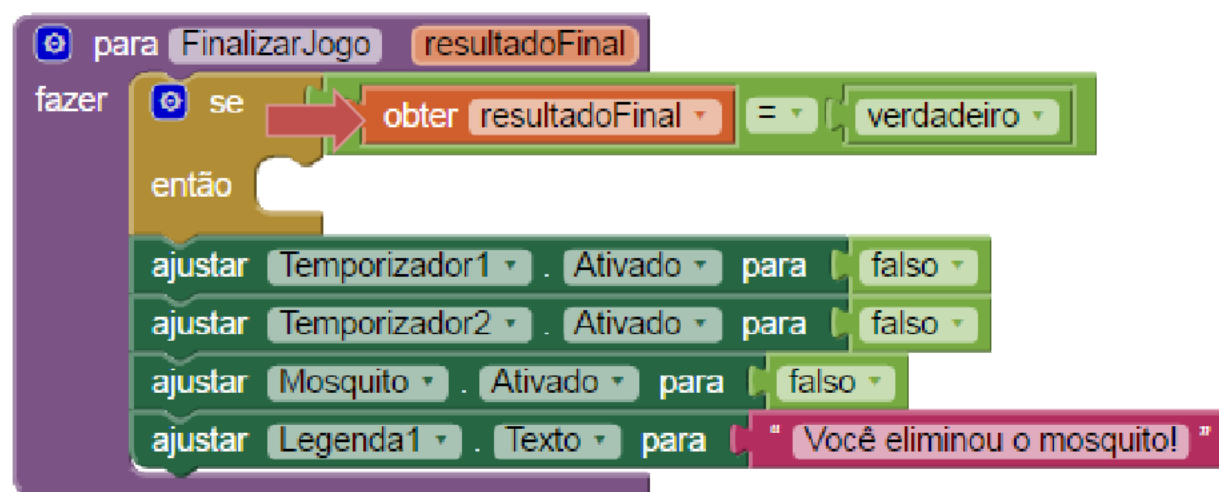
Primeiro adicione o bloco "se então" ao "FinalizarJogo", e coloque uma bloco Lógico de comparação (=).



ADICIONANDO TEMPO

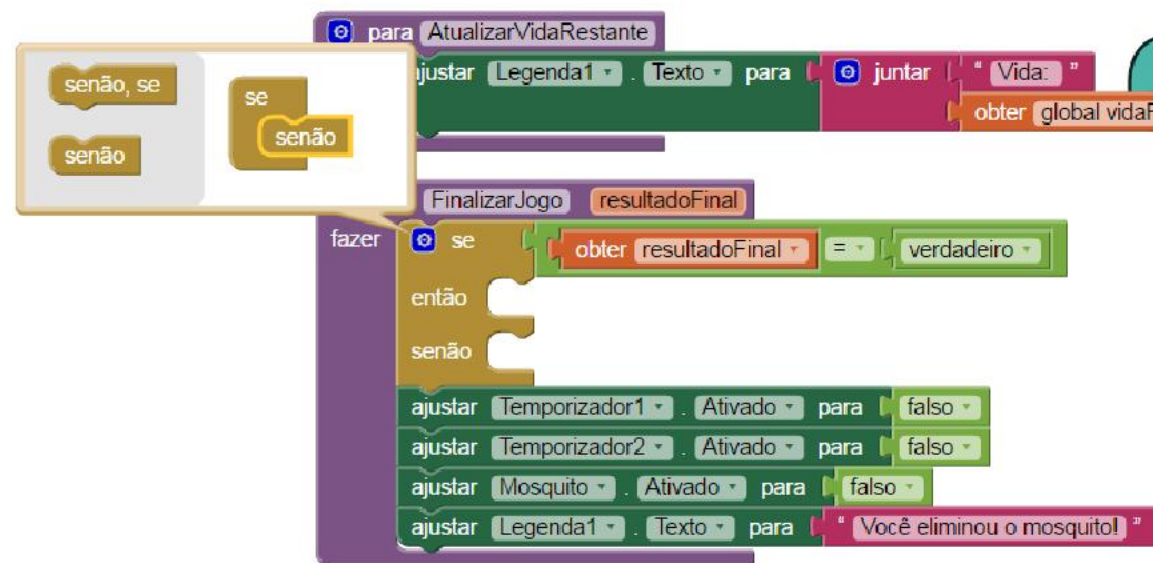
Adicione na comparação a variável **resultadoFinal** e o valor Verdadeiro.

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”.



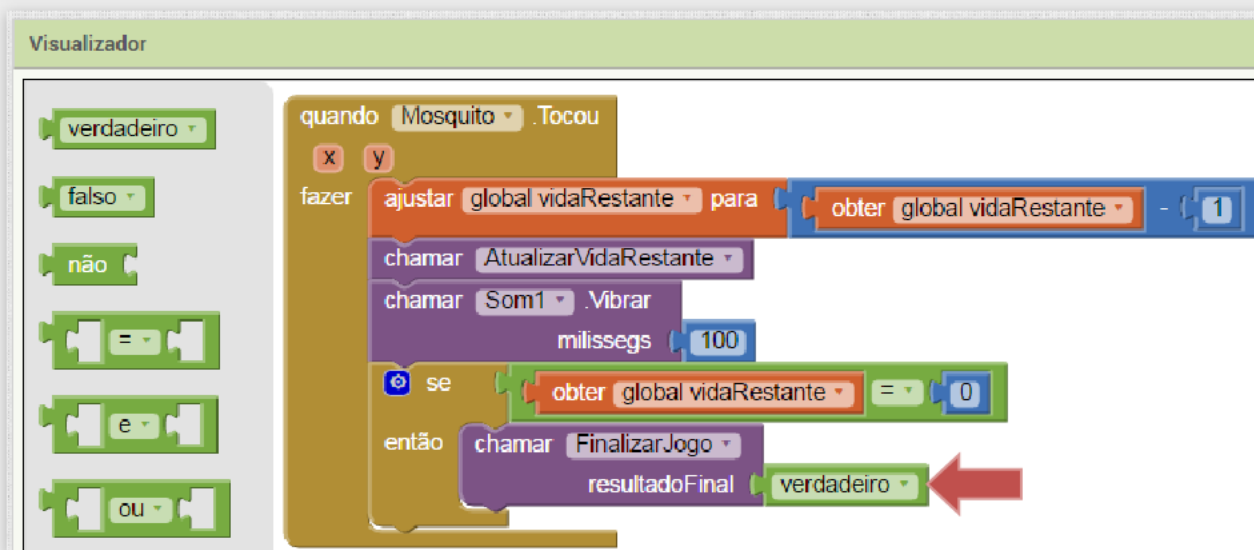
ADICIONANDO TEMPO

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.



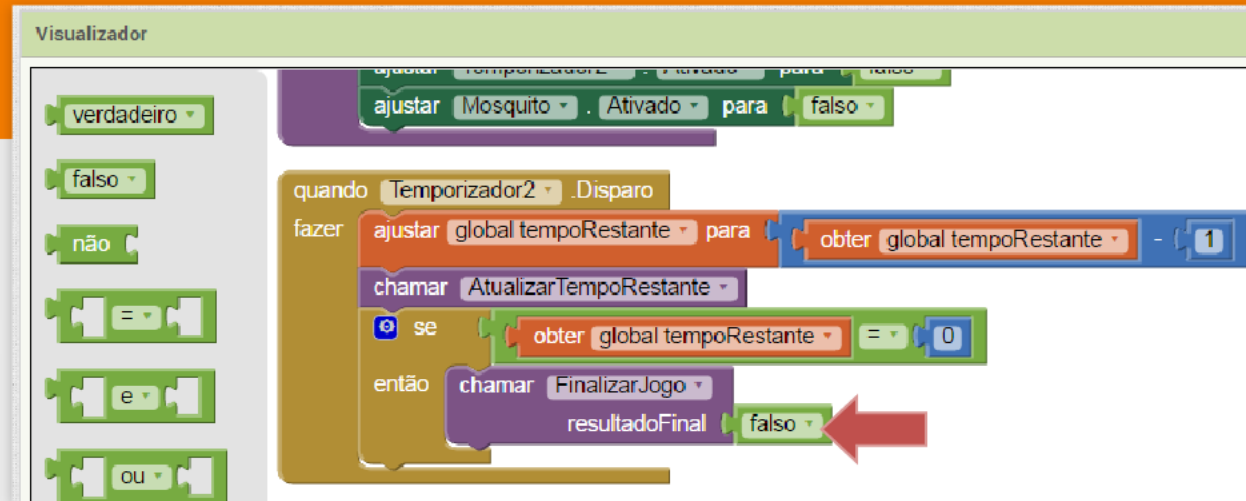
Vamos atualizar o bloco “Mosquito.Tocou” passando o valor “Verdadeiro” como parâmetro, pois neste procedimento o jogador foi vencedor.

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.

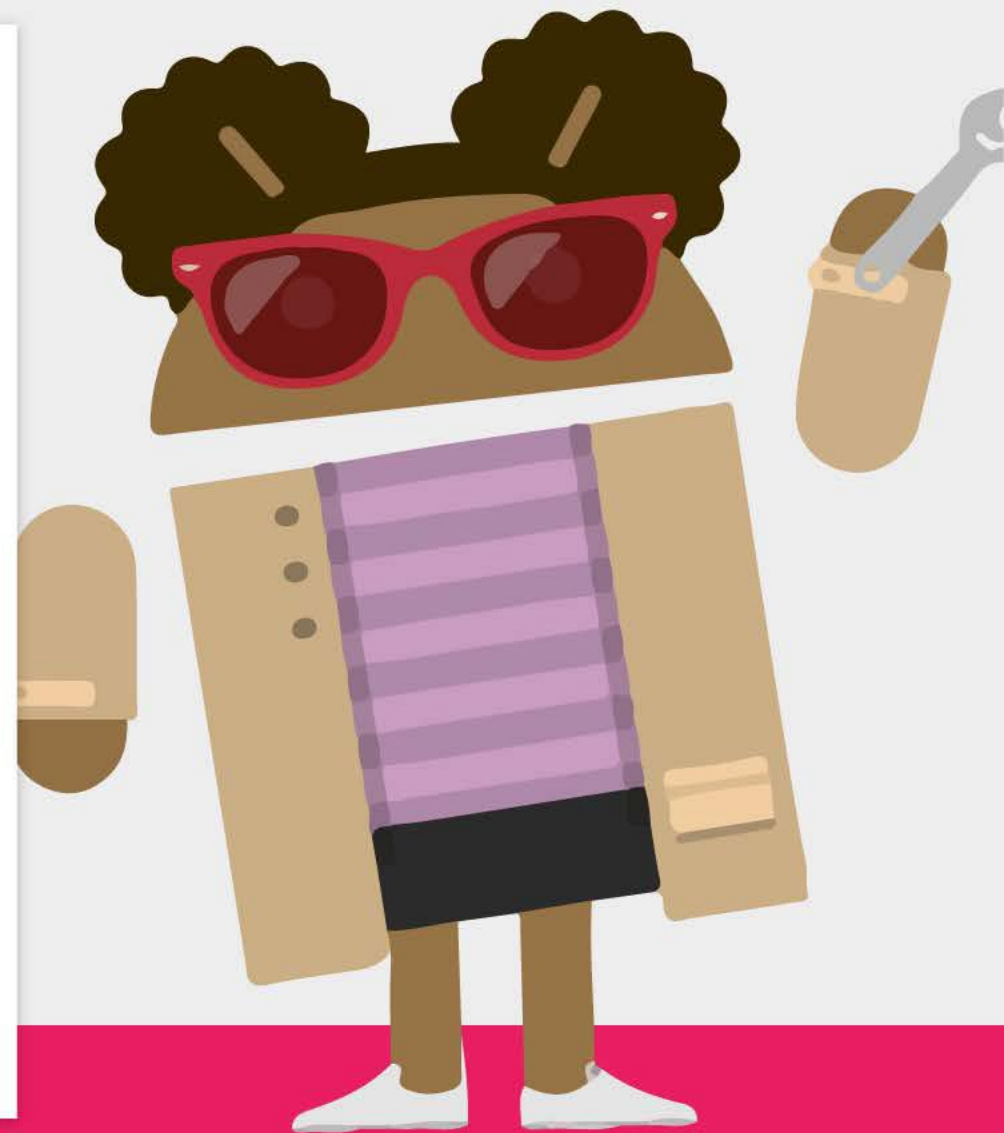
Agora verifique se o seu jogo está funcionando perfeitamente com o tempo para matar o mosquito!

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



AGORA É COM
VOCÊ!

SEJA CRIATIVO



COMPUTAÇÃO NA ESCOLA

