

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.

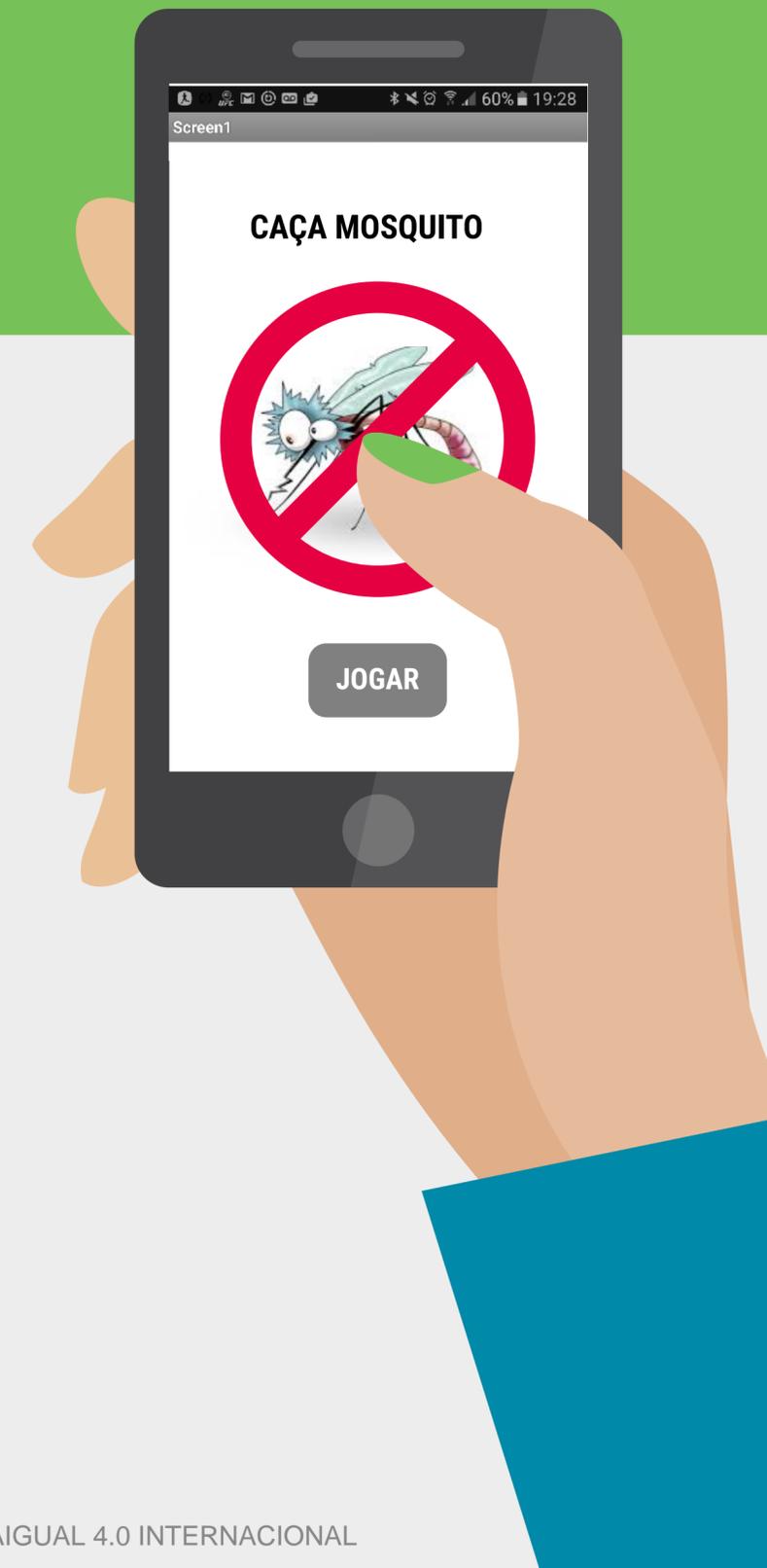


Fazer um mosquito se mover pela tela.

Adicionar vida ao mosquito.

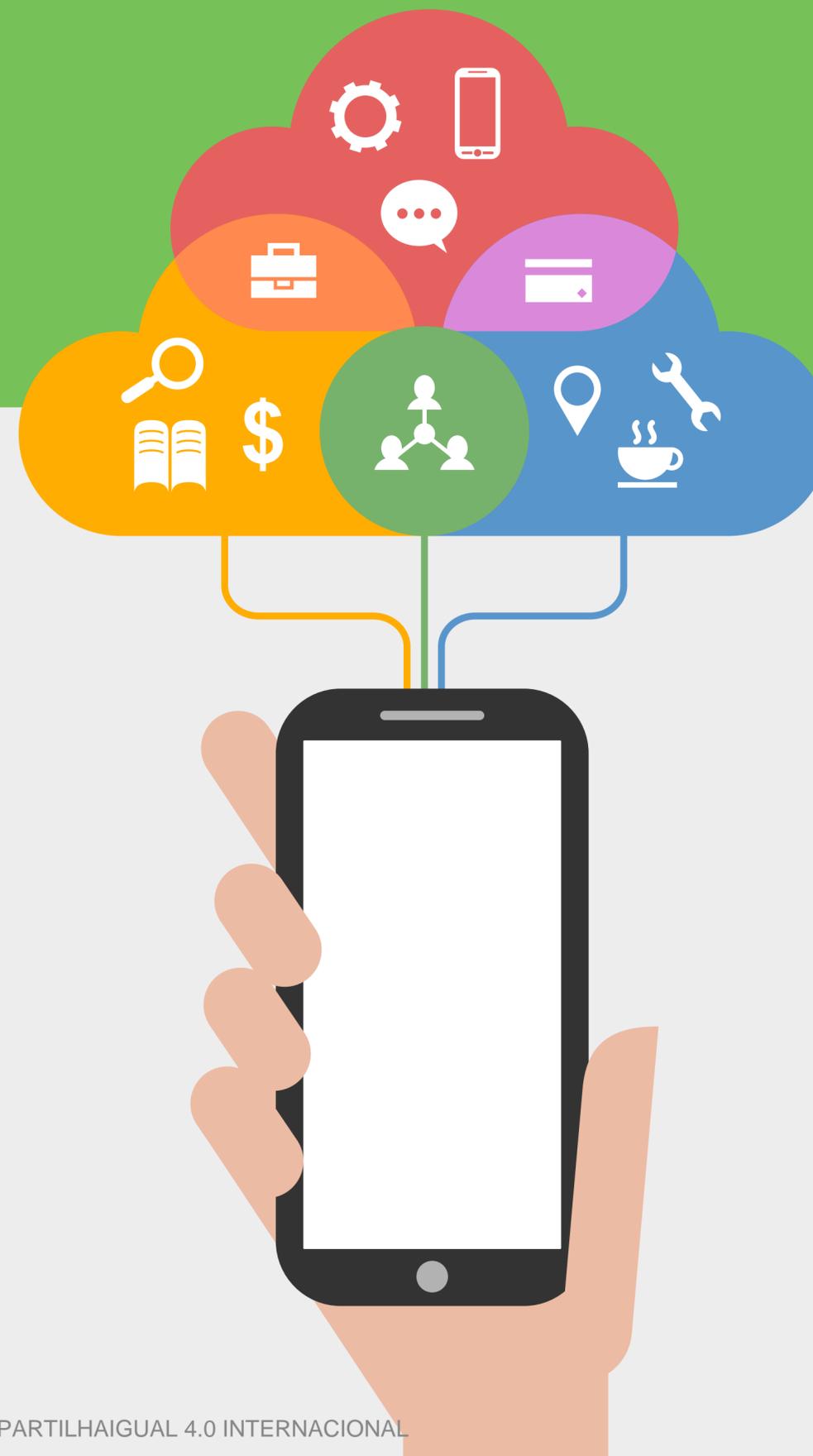
Adicionar um tempo para matar o mosquito.

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

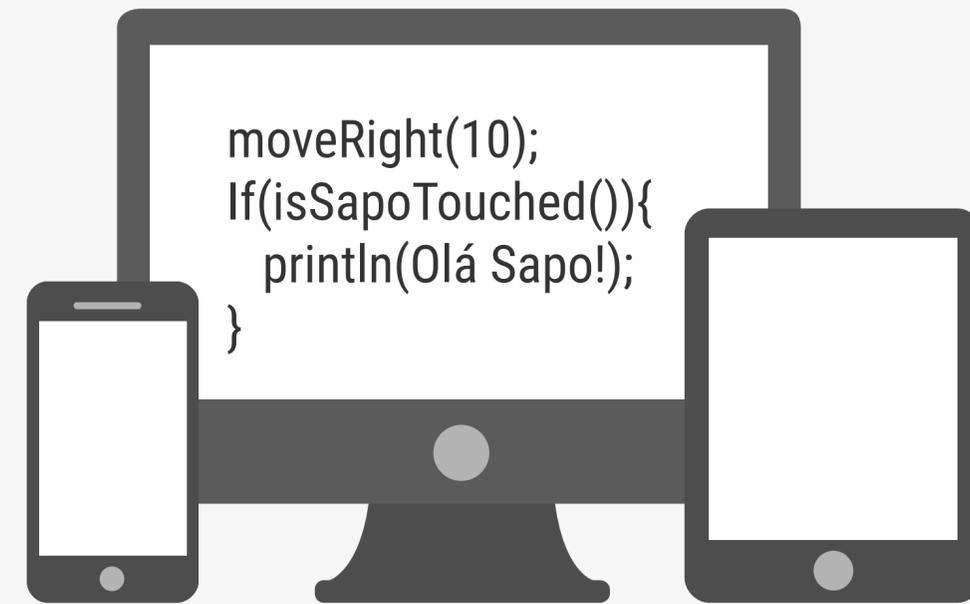
Um algoritmo é um conjunto de instruções para realizar uma determinada tarefa.

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

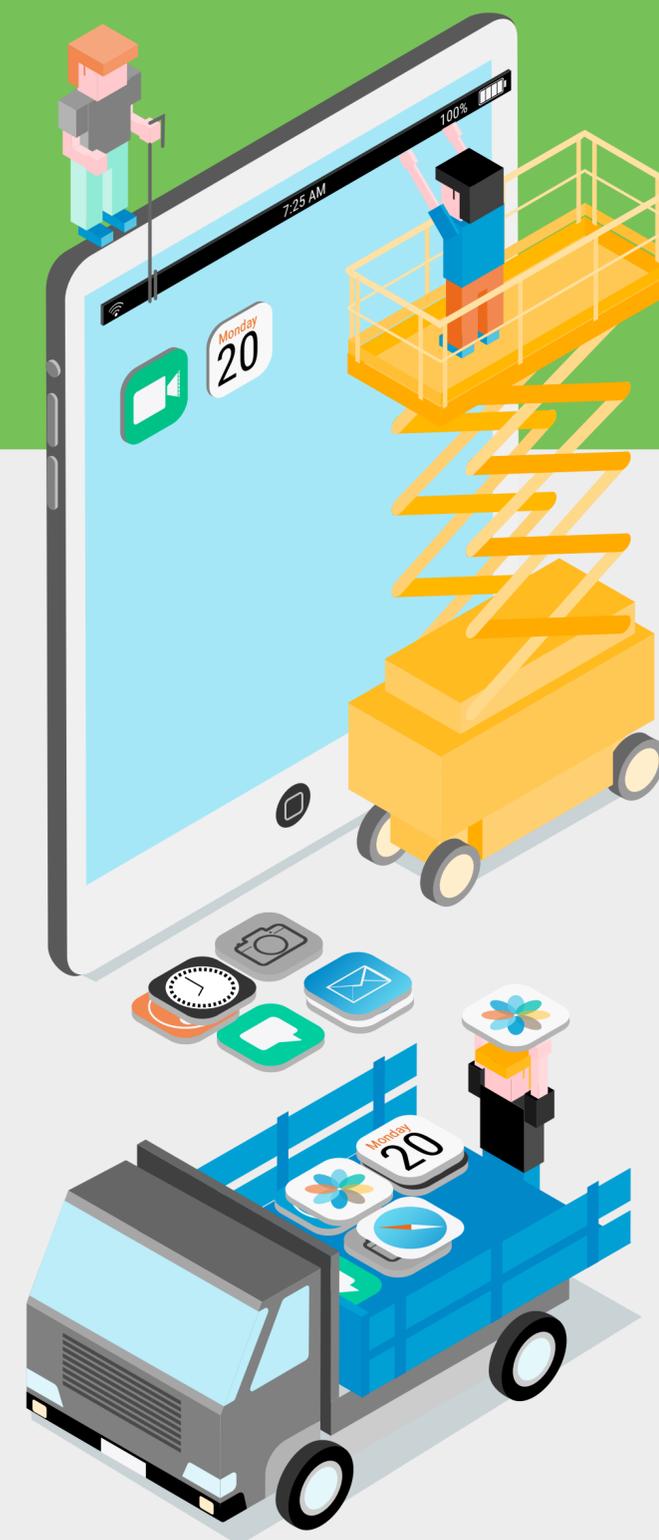
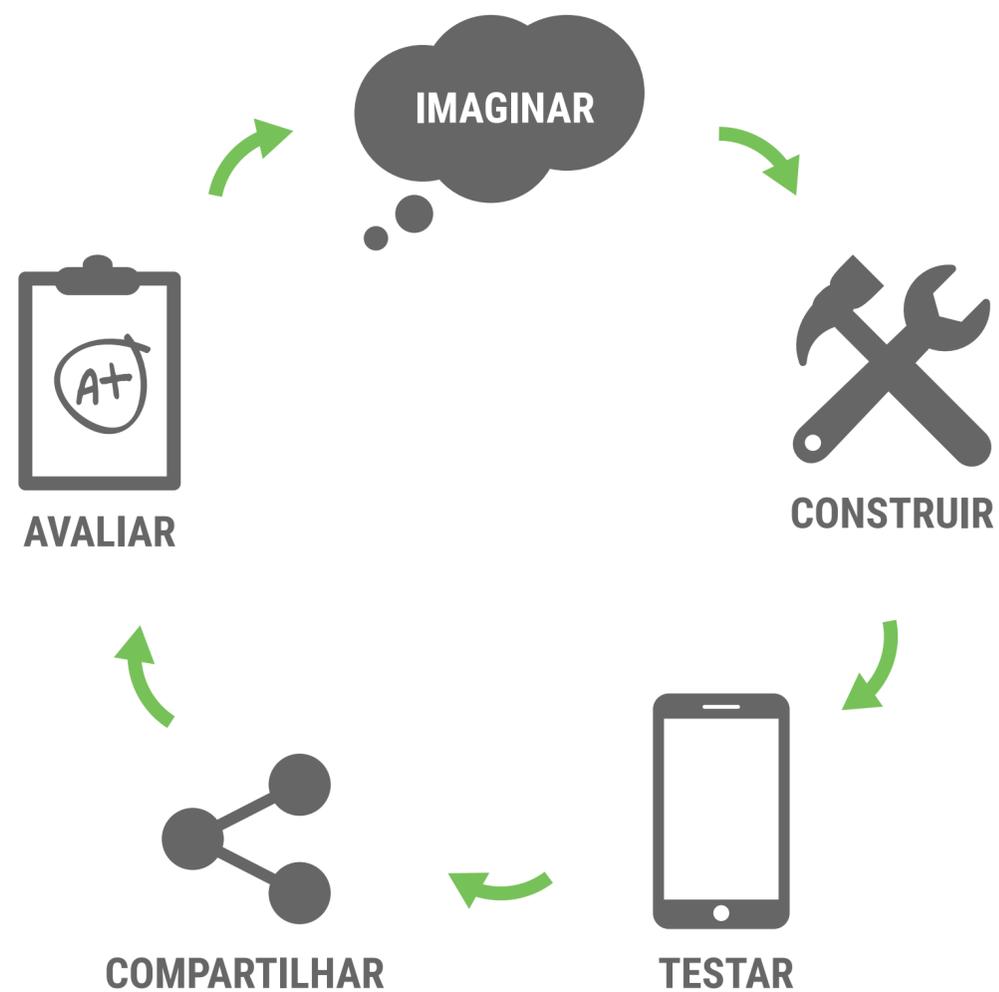


SOFTWARE

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



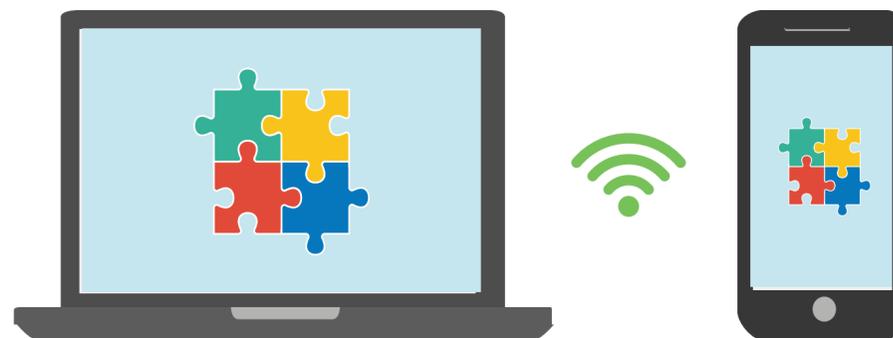
COMO FAZER APPS DE CELULAR



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

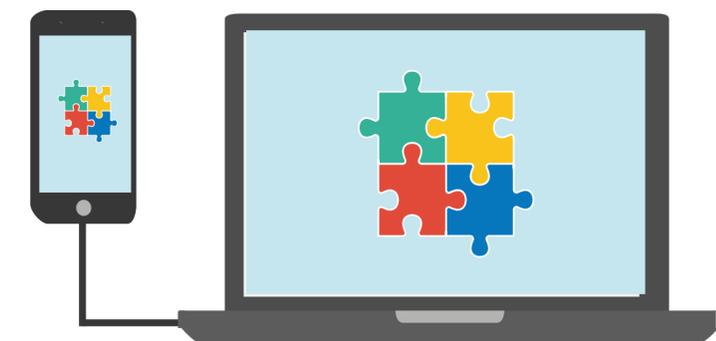
OPÇÃO 2

Não tem um dispositivo Android? Use um emulador



OPÇÃO 3

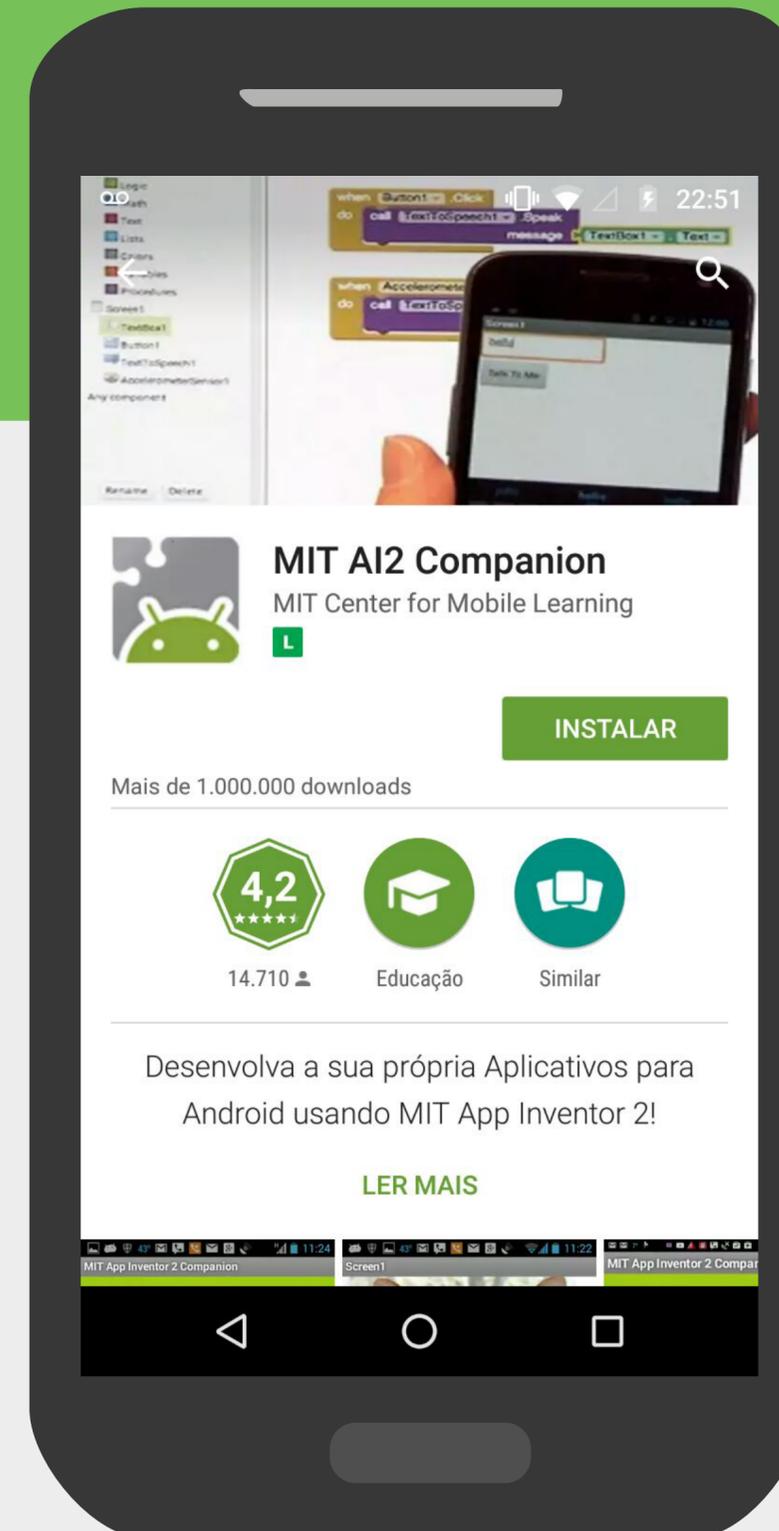
Não tem conexão wi-fi? Construa apps com um dispositivo Android e um cabo USB



INSTALANDO O APLICATIVO

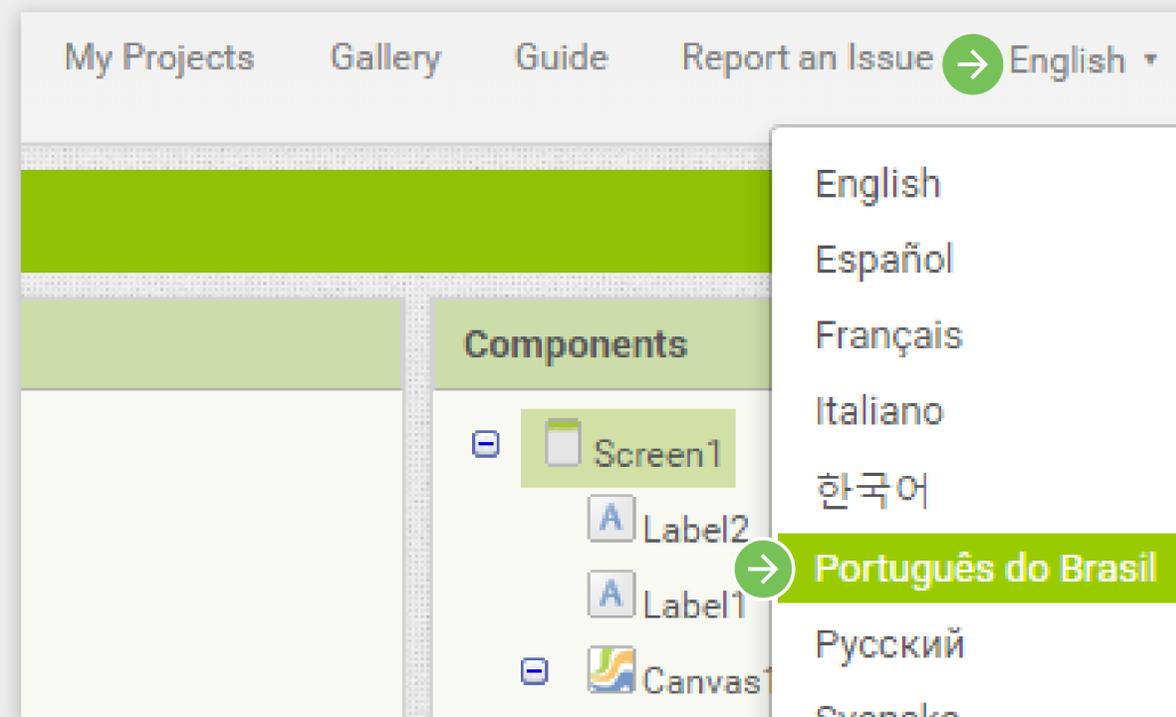
Para poder testar o seu jogo do mosquito em seu celular, primeiramente precisamos instalar o aplicativo do MIT.

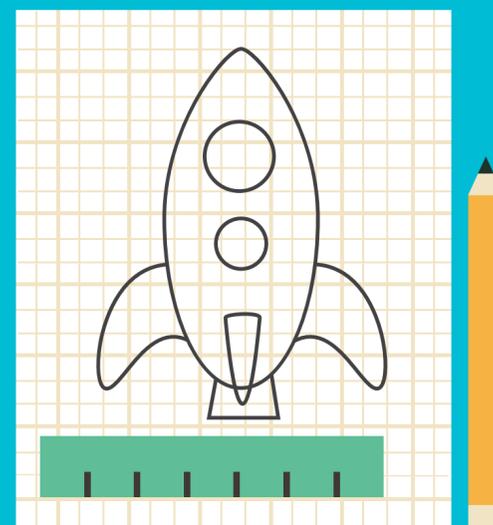
1. Vá na Google Play e procure por 'MIT AI2 Companion'.
2. Escolha o app de mesmo nome
3. Clique em 'Instalar' e aguarde o download e instalação do aplicativo



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.

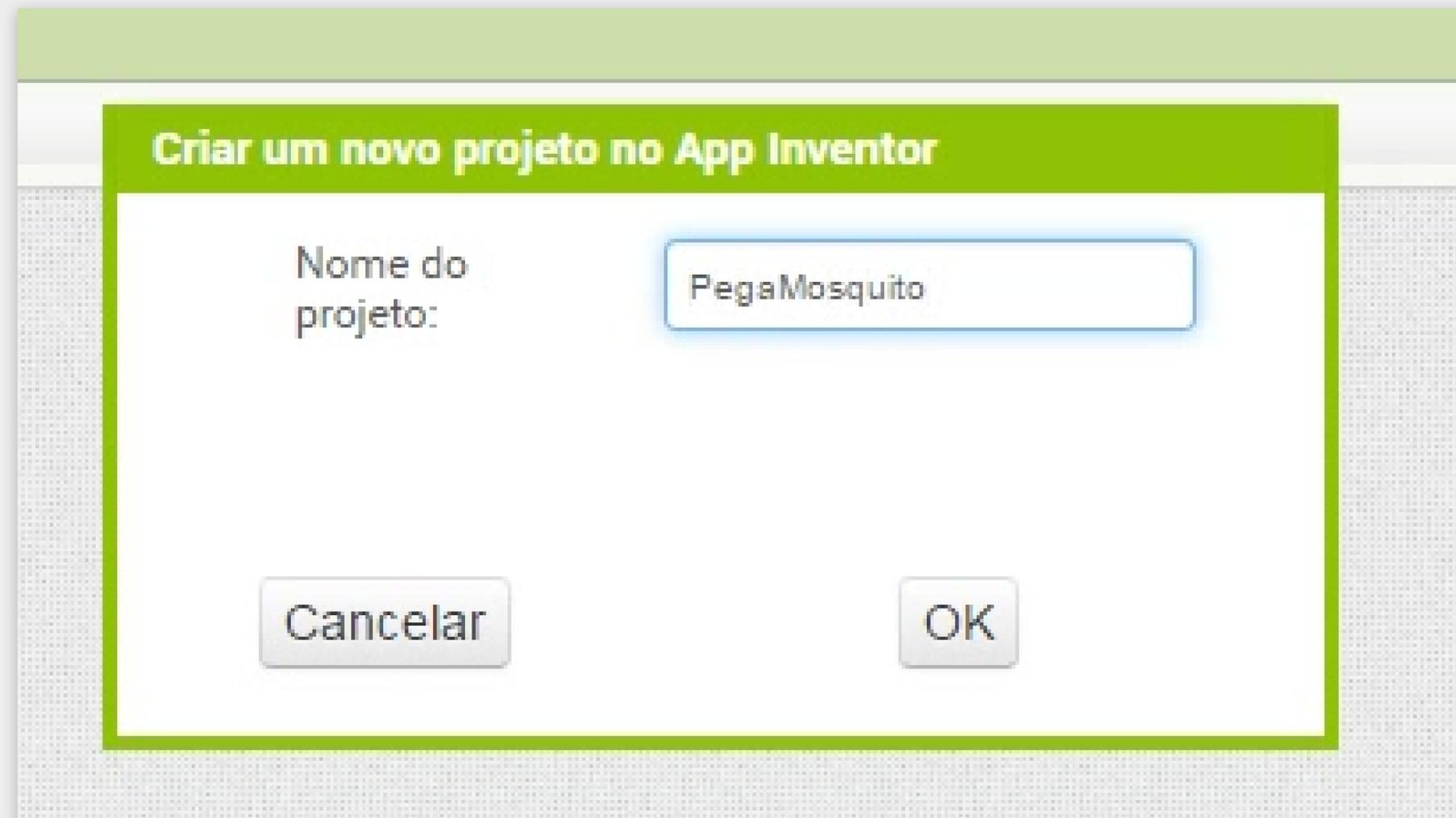


The screenshot shows the MIT App Inventor 2 Beta interface. At the top left is the MIT App Inventor logo (a green Android robot) and the text "MIT App Inventor 2 Beta". To the right are three menu items: "Projetos", "Conectar", and "Compilar", each with a downward arrow. Below the header is a green bar containing three buttons: "Iniciar novo projeto ..." (highlighted in white), "Apagar Projeto", and "Publicar na Galeria". Below this bar is a section titled "Meus Projetos" in a light green header. Underneath is a table with two columns: "Nome" and "Data de Criação".

Nome	Data de Criação
------	-----------------

CRIANDO UM NOVO PROJETO

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



Criar um novo projeto no App Inventor

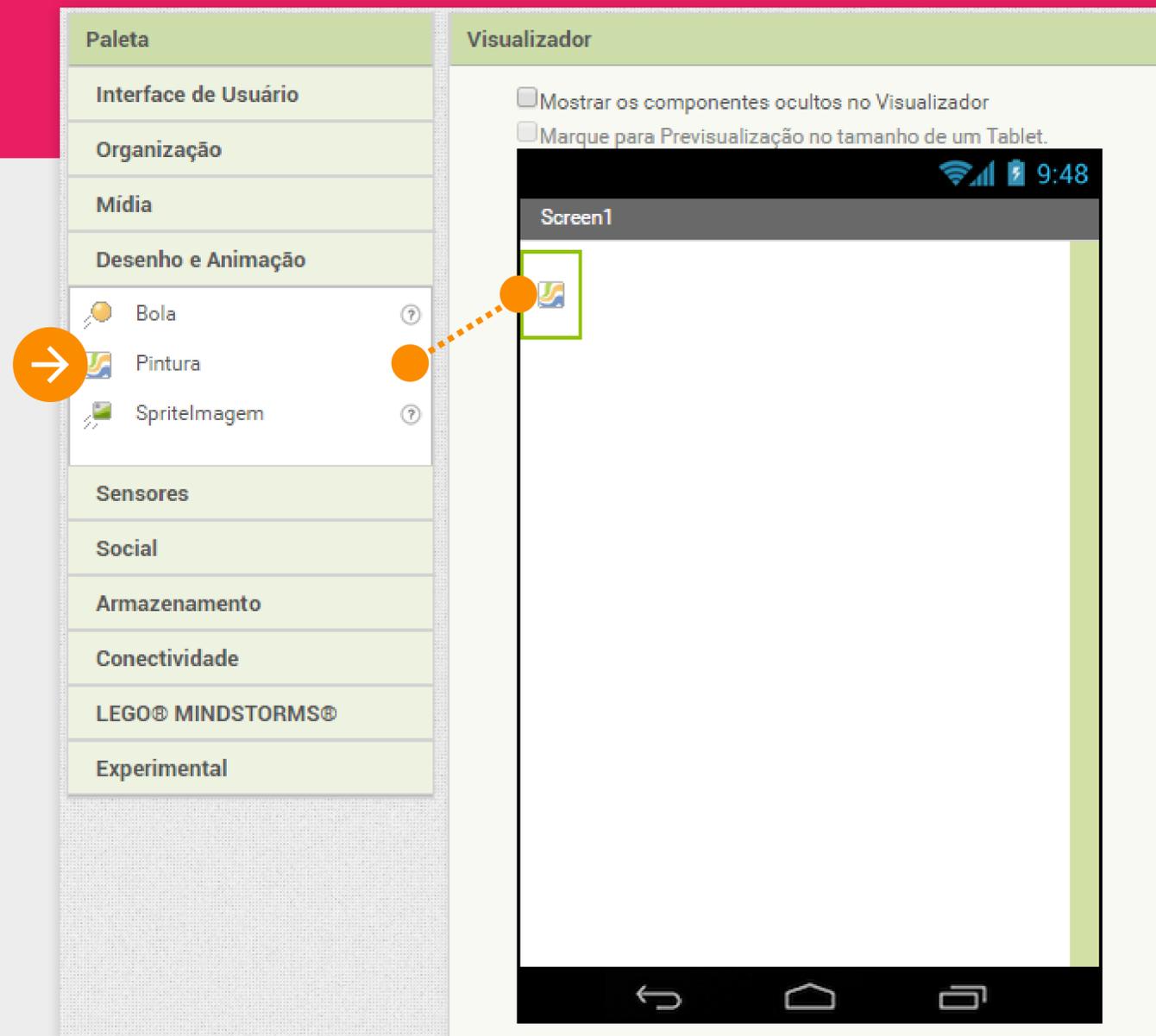
Nome do projeto:

Cancelar OK

DEFININDO A INTERFACE



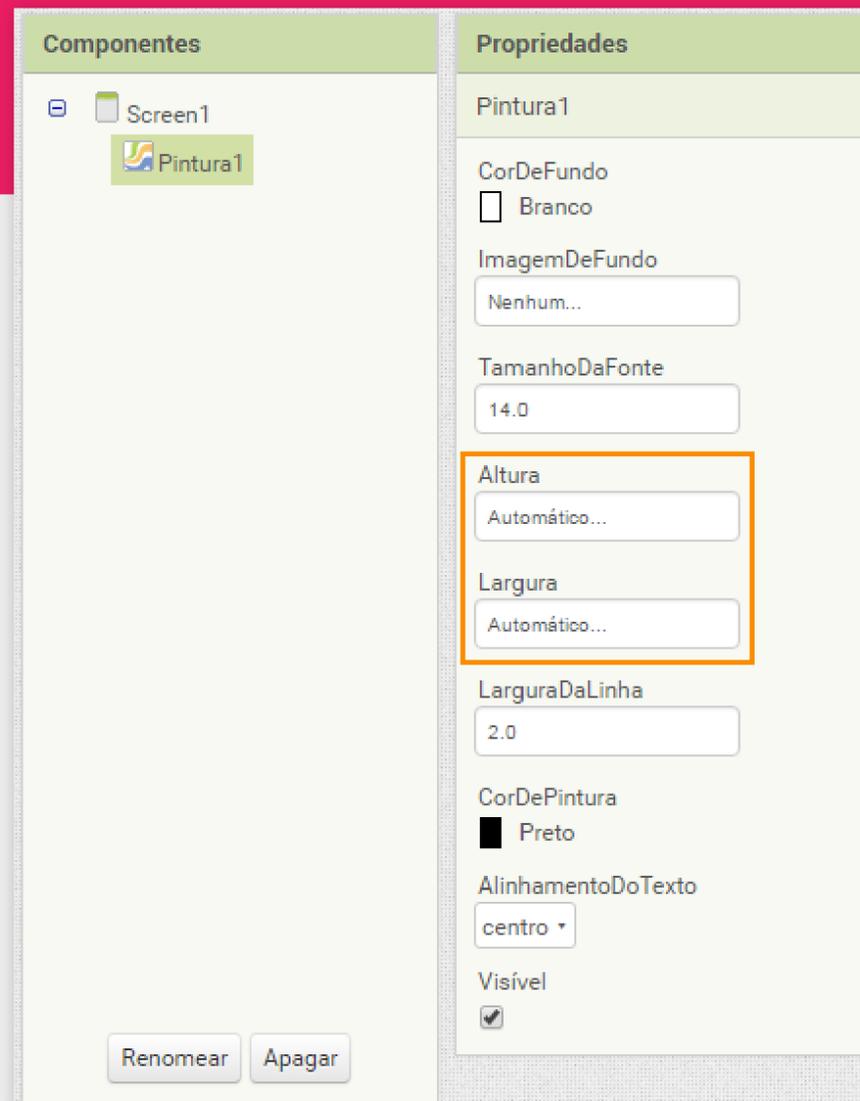
DEFININDO A INTERFACE



Vamos começar definindo a nossa interface.

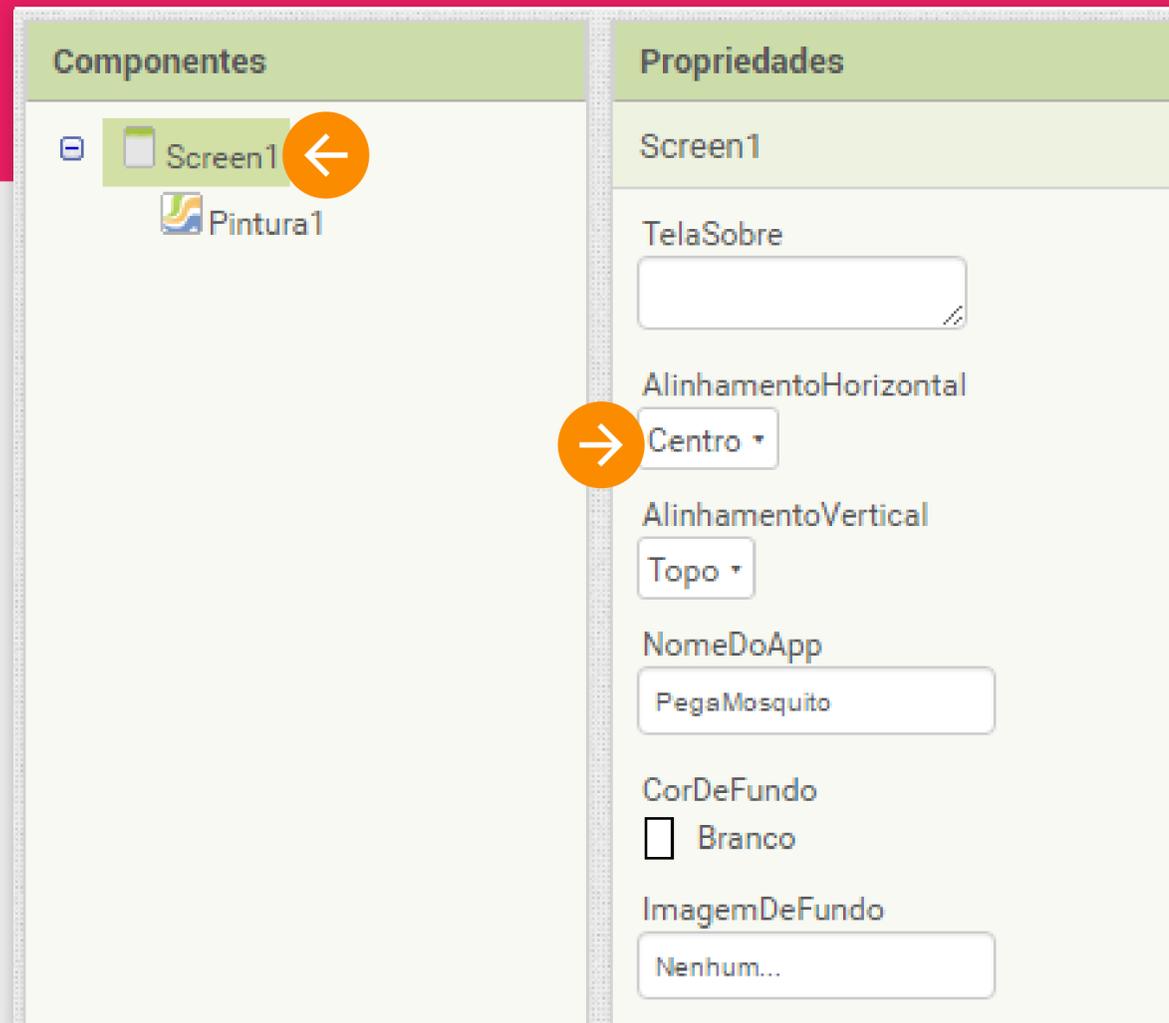
Primeiro vamos clicar na paleta **“Desenho e Animação”** e arrastar o componente **“Pintura”** para dentro da nossa tela (no visualizador).

DEFININDO A INTERFACE

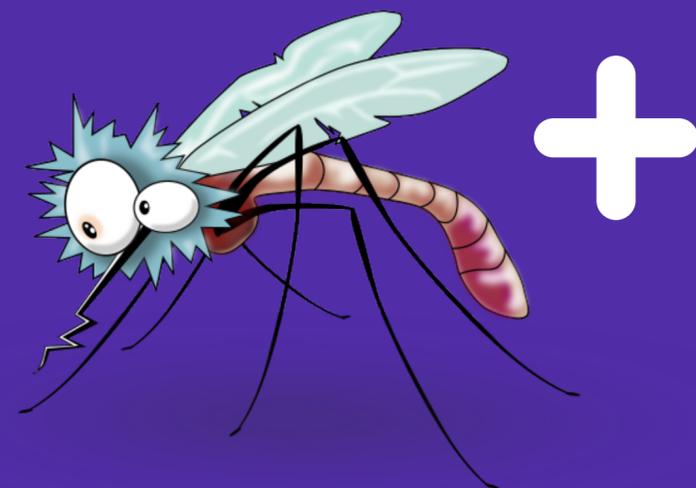


Vamos alterar as propriedades da pintura para que ela possua altura de 300 pontos e largura de 300 pontos.

DEFININDO A INTERFACE



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

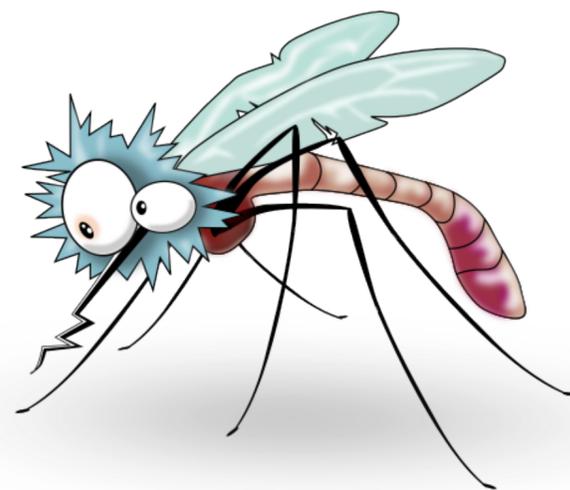


ADICIONANDO O **MOSQUITO**

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.

The screenshot displays the Scratch IDE interface with four main panels: Paleta (left), Visualizador (center), Componentes (right), and Propriedades (far right). The Paleta panel is expanded to the 'Mídia' section, where the 'Enviar Arquivo ...' button is highlighted with a red circle. The Visualizador panel shows a mobile device simulation with a small image of a mosquito on the screen. The Componentes panel shows a tree view with 'Screen1' and 'Pintura1'. The Propriedades panel shows various settings for the selected object, including 'NomeDoApp' set to 'PegaMosquito'.

ADICIONANDO O MOSQUITO

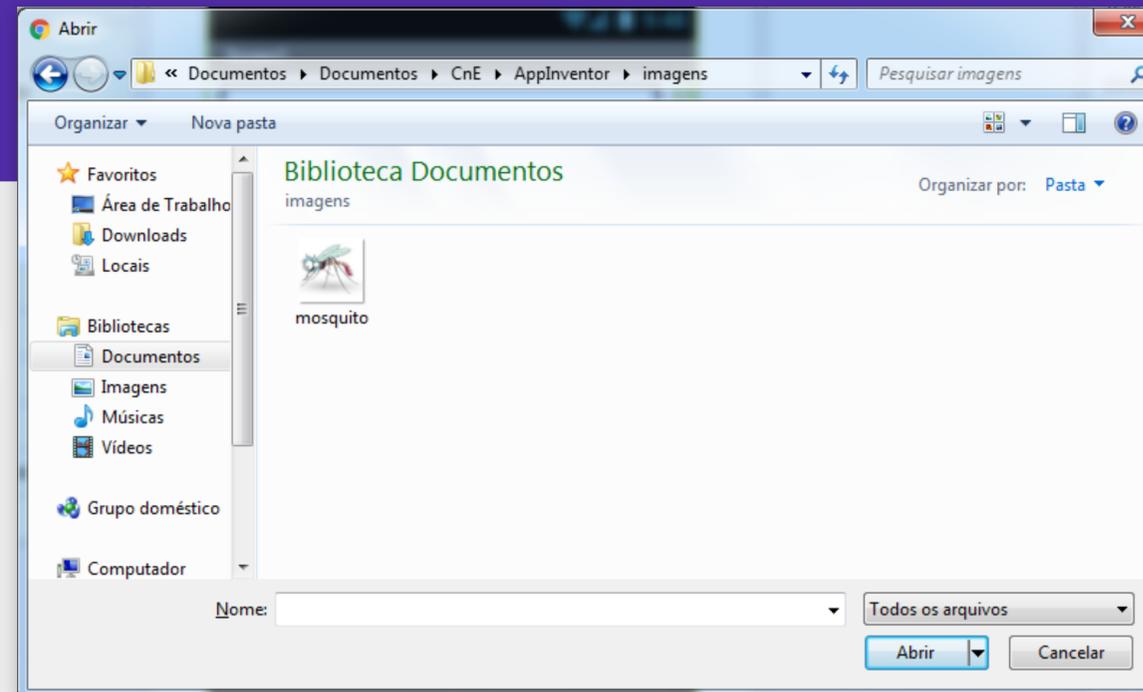
Clique no botão **“Escolher arquivo”** e selecione o arquivo com nome **“mosquito.png”** que você baixou.

Enviar Arquivo ...

Escolher arquivo Nenhum arquivo selecionado

Cancelar

OK



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"**.

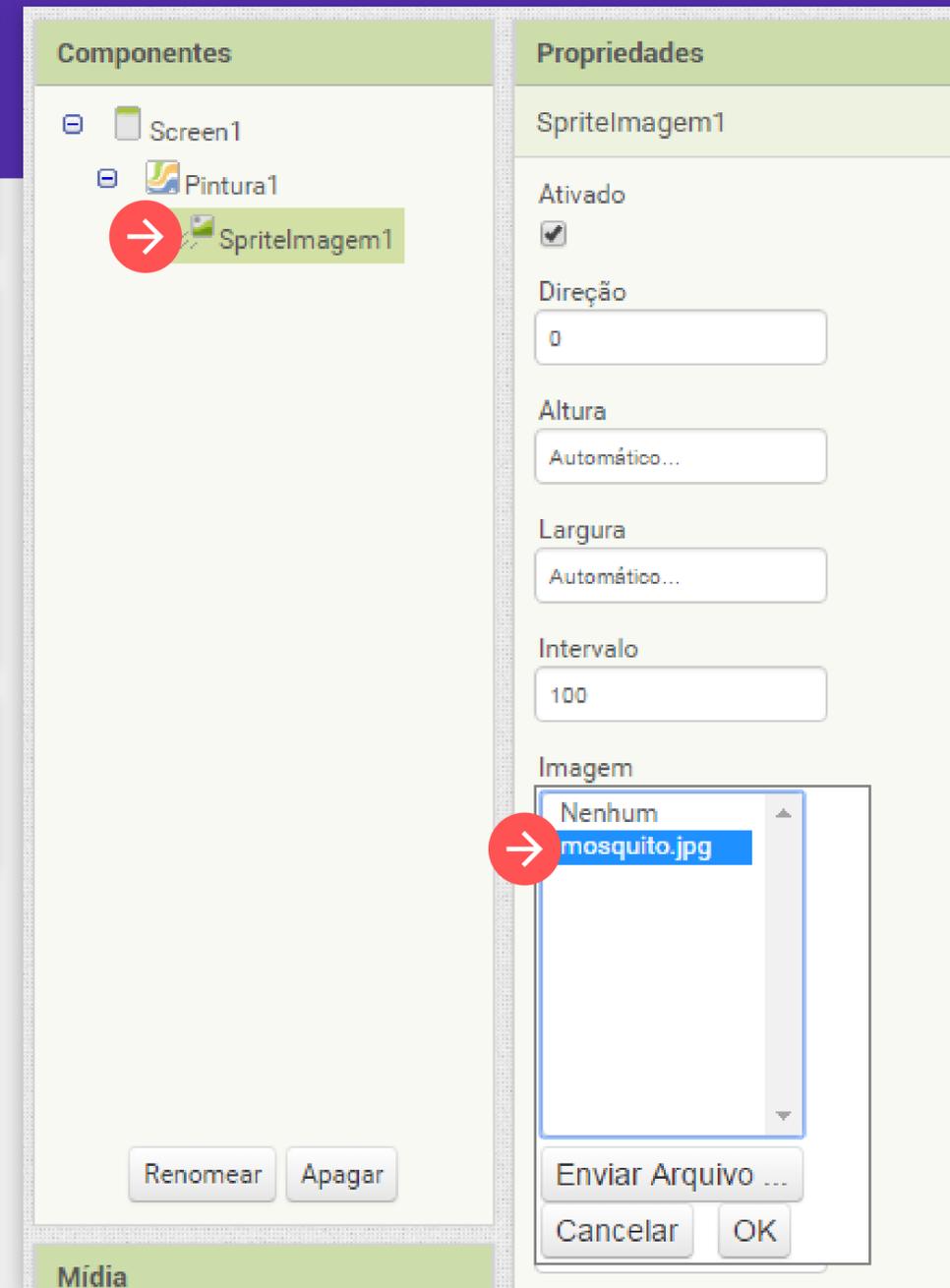
The screenshot displays the Scratch IDE interface with four main panels: Paleta, Visualizador, Componentes, and Propriedades. The Paleta panel on the left is expanded to the 'Desenho e Animação' category, where the 'SpriteImagem' component is highlighted with a red circle. A red dashed line indicates the drag action from this component to the 'Pintura' component in the Componentes panel. The Visualizador panel shows a stage labeled 'Screen1' with a drawing area containing a small mosquito sprite. The Propriedades panel on the right shows the settings for the selected 'SpriteImagem1' component, including options for 'Ativado', 'Direção', 'Altura', 'Largura', 'Intervalo', 'Imagem', 'Rodar', 'Velocidade', and 'Visível'.

ADICIONANDO O MOSQUITO

Agora devemos definir a imagem dessa Sprite.

Para fazer isso, nas Propriedades do componente **"SpriteImagem1"**, clique no campo **"Imagem"** e selecione o arquivo **"mosquito.png"**.

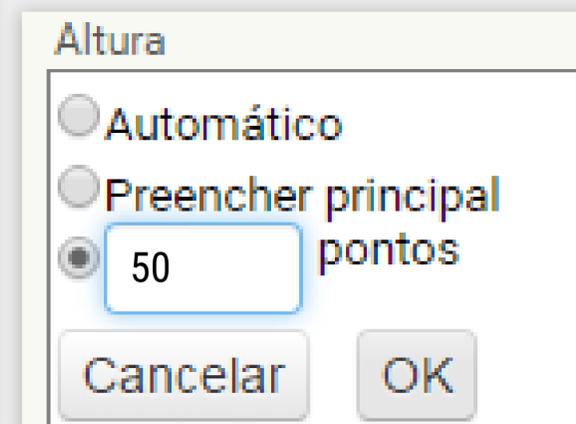
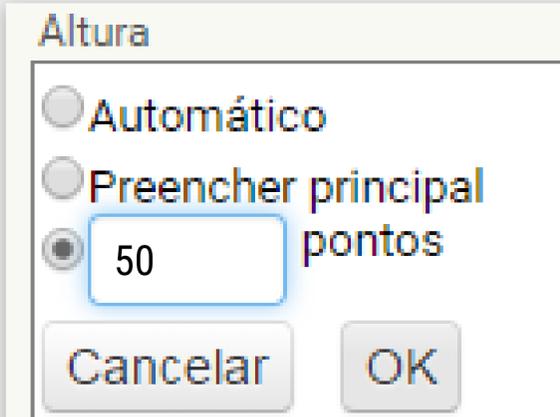
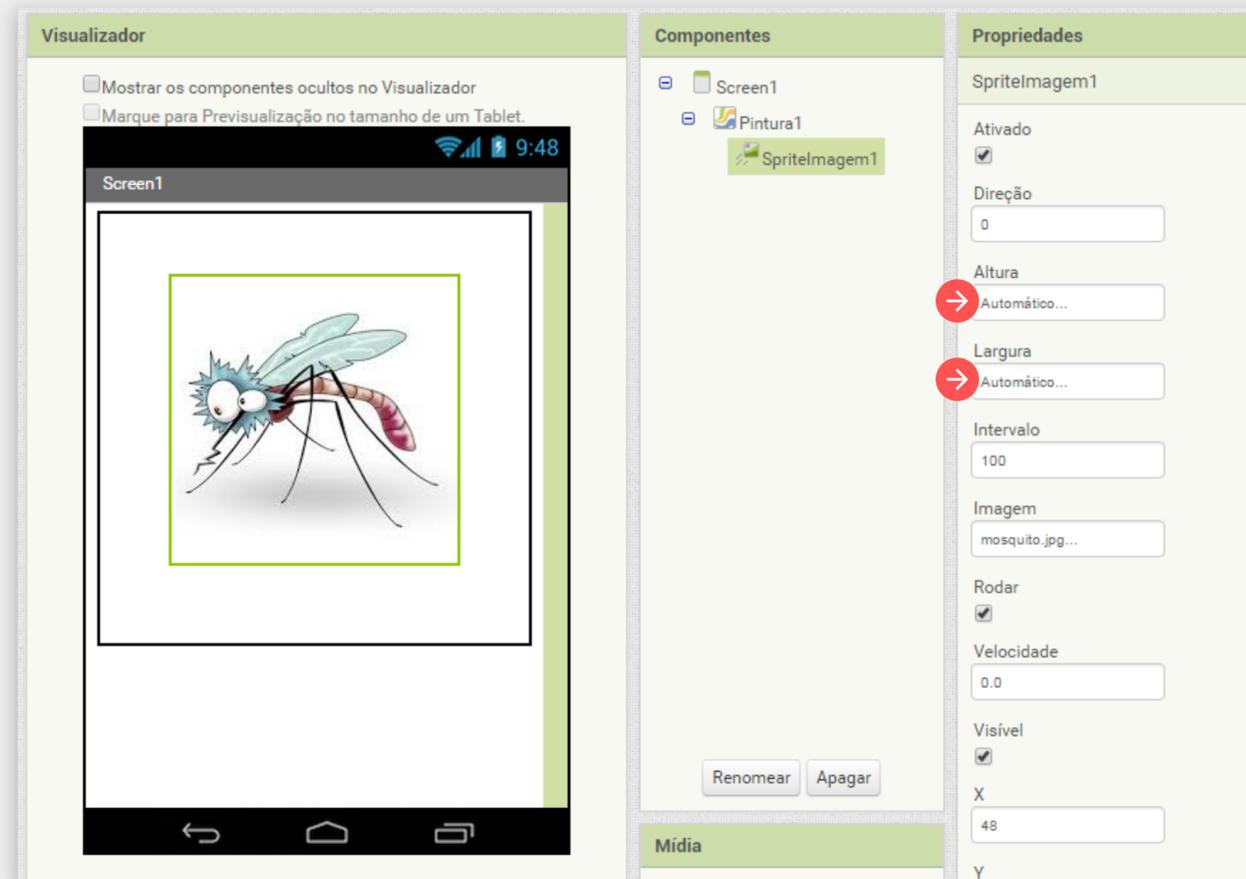
Clique em OK



ADICIONANDO O MOSQUITO

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

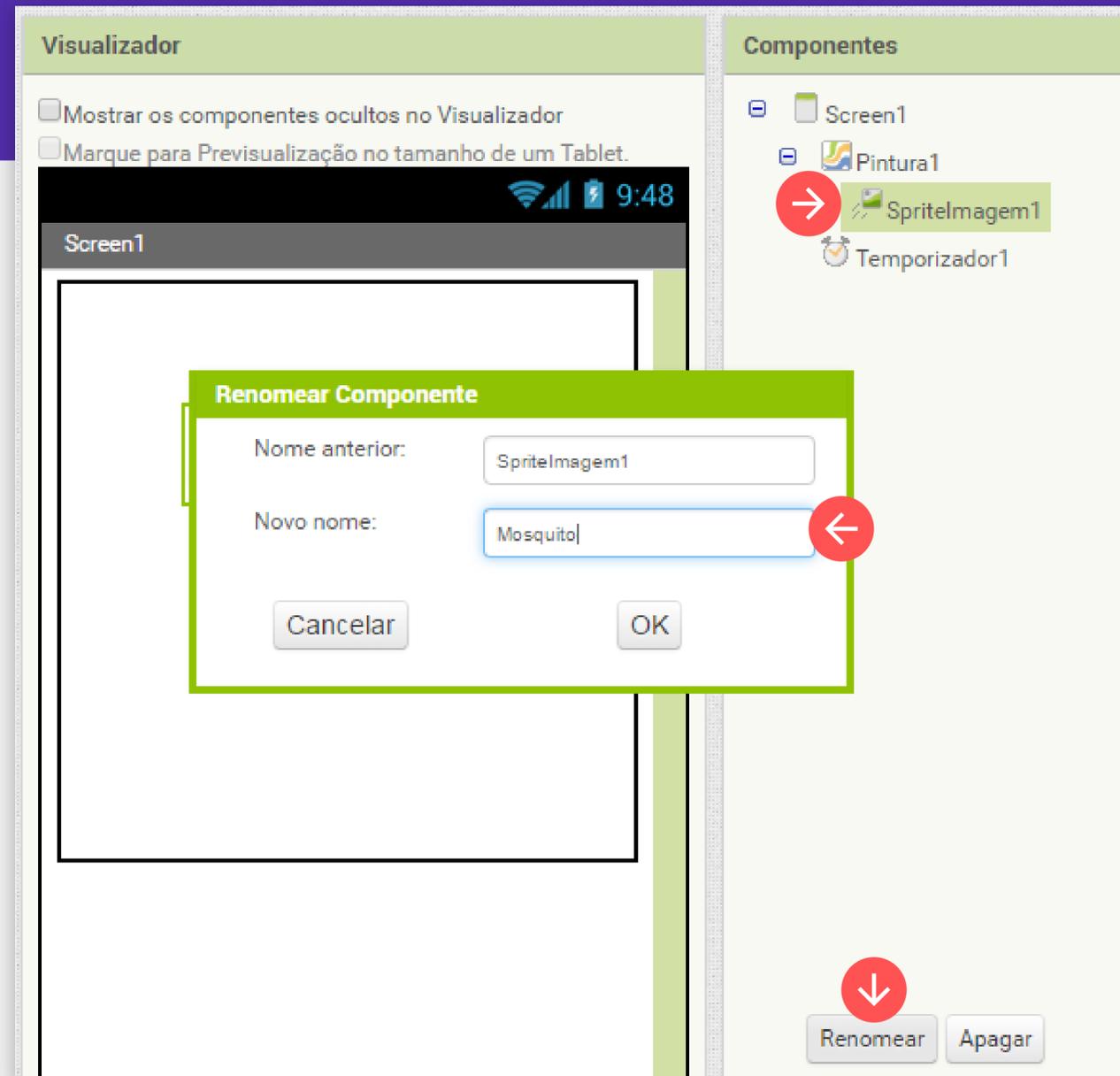
Mude a altura e largura do mosquito para 50.

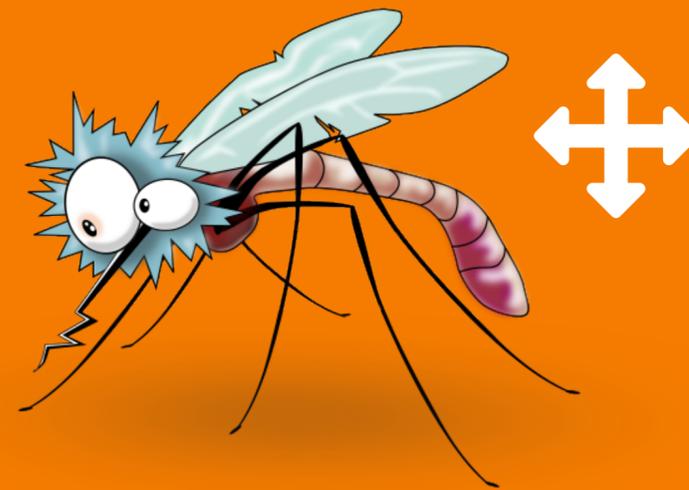


ADICIONANDO O MOSQUITO

Para ficar mais fácil reconhecer que o componente SpritImagem1 é a imagem do nosso mosquito, vamos mudar o nome dele para **“Mosquito”**.

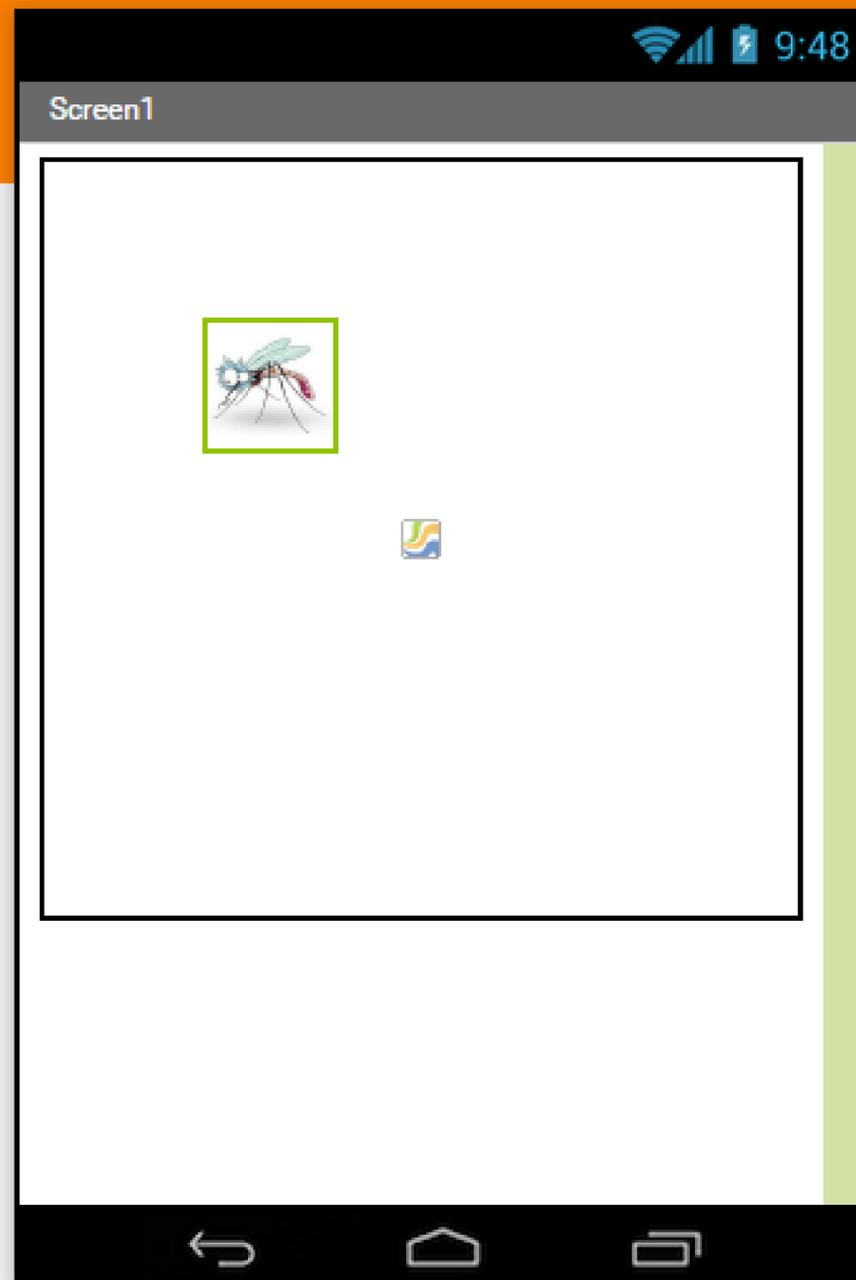
Selecione o componente e clique em Renomear. Digite **“Mosquito”** e clique em OK.





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

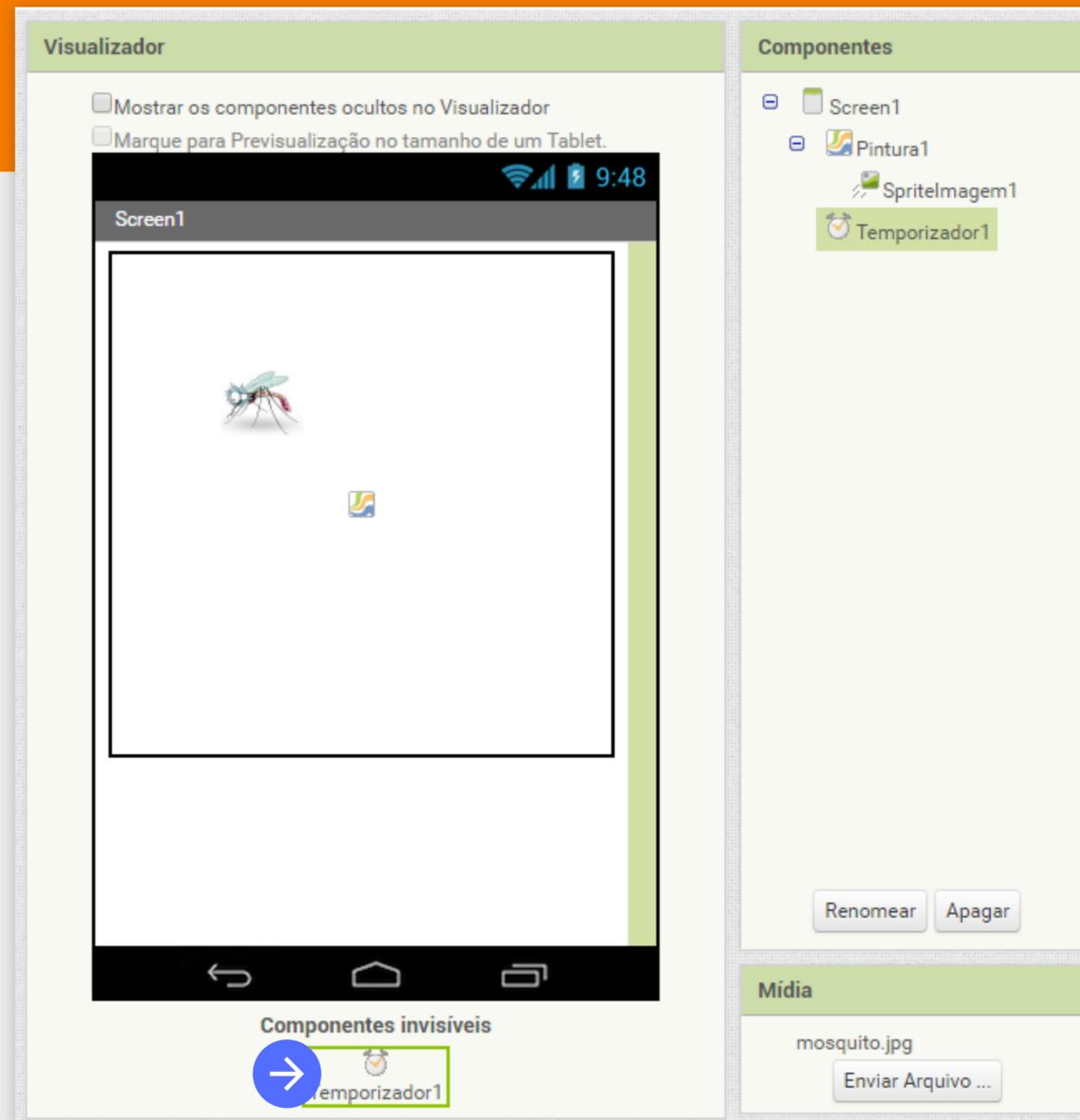
Para o mosquito se movimentar iremos utilizar o componente **Temporizador**. Este será responsável por mandar o mosquito se movimentar a cada meio segundo.

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

The image shows the Scratch IDE interface. On the left is the 'Paleta' (Palette) with categories: Interface de Usuário, Organização, Mídia, Desenho e Animação, and Sensores. Under 'Sensores', the 'Temporizador' component is highlighted with a blue circle. A dashed blue line connects this circle to a 'Temporizador2' component on the 'Visualizador' (Stage) area. The stage area shows a mosquito icon and a 'Temporizador2' component. Above the stage, there are two checkboxes: 'Mostrar os componentes ocultos no Visualizador' and 'Marque para Previsualização no tamanho de um Tablet'. The stage area also shows a status bar with 'Screen1', a Wi-Fi icon, a battery icon, and the time '9:48'.

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.

The screenshot shows the Scratch IDE interface. On the left, the 'Componentes' panel lists 'Screen1', 'Pintura1', 'SpriteImagem1', and 'Temporizador1'. 'Temporizador1' is highlighted with a blue arrow icon. On the right, the 'Propriedades' panel shows the properties for 'Temporizador1'. The 'DisparosContínuos' property is checked with a checkbox and has a blue arrow icon to its right. The 'Ativado' property is also checked with a checkbox. The 'Intervalo' property is set to '500' in a text input field, with a blue arrow icon to its right.

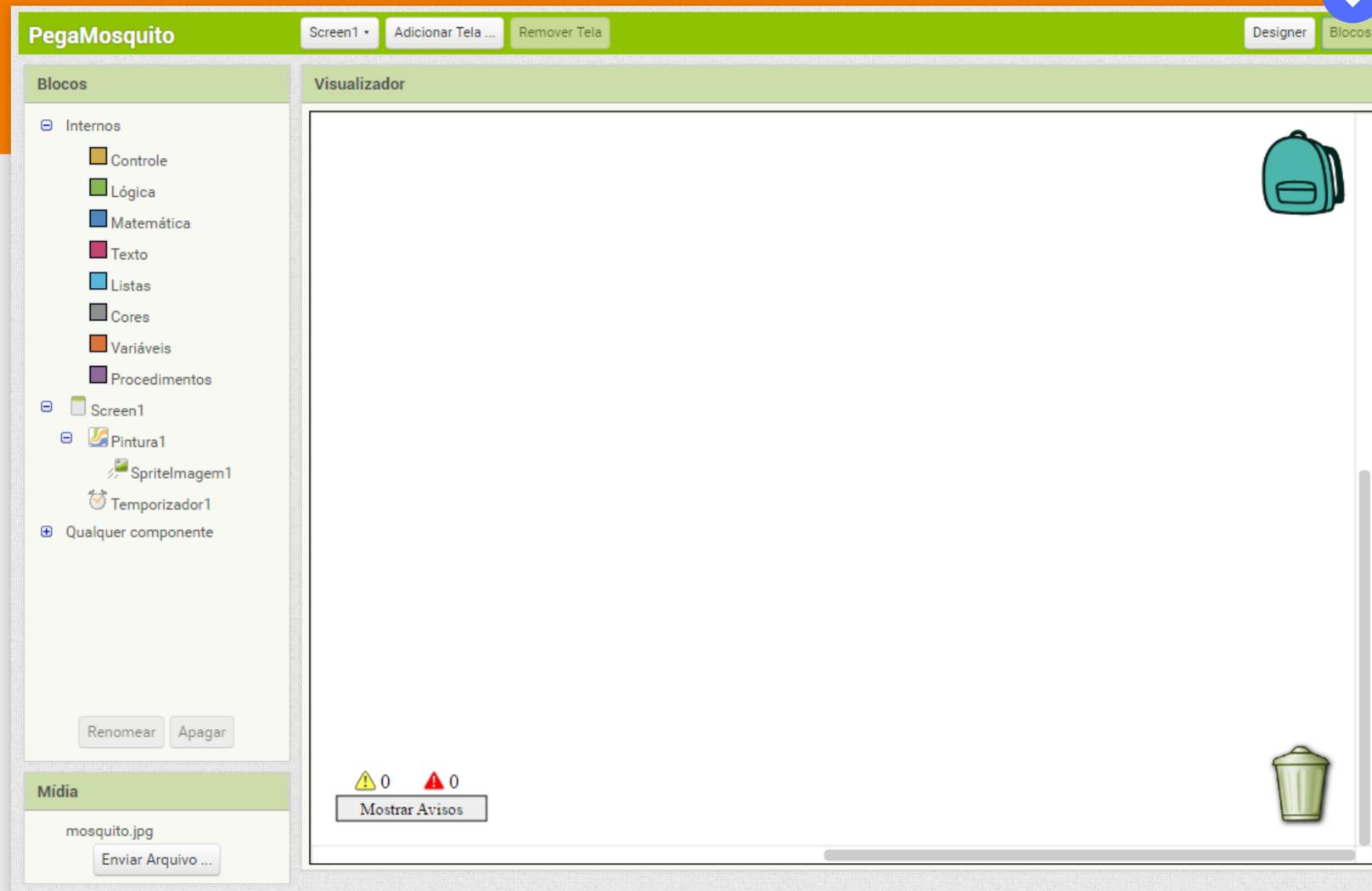
DICA

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.

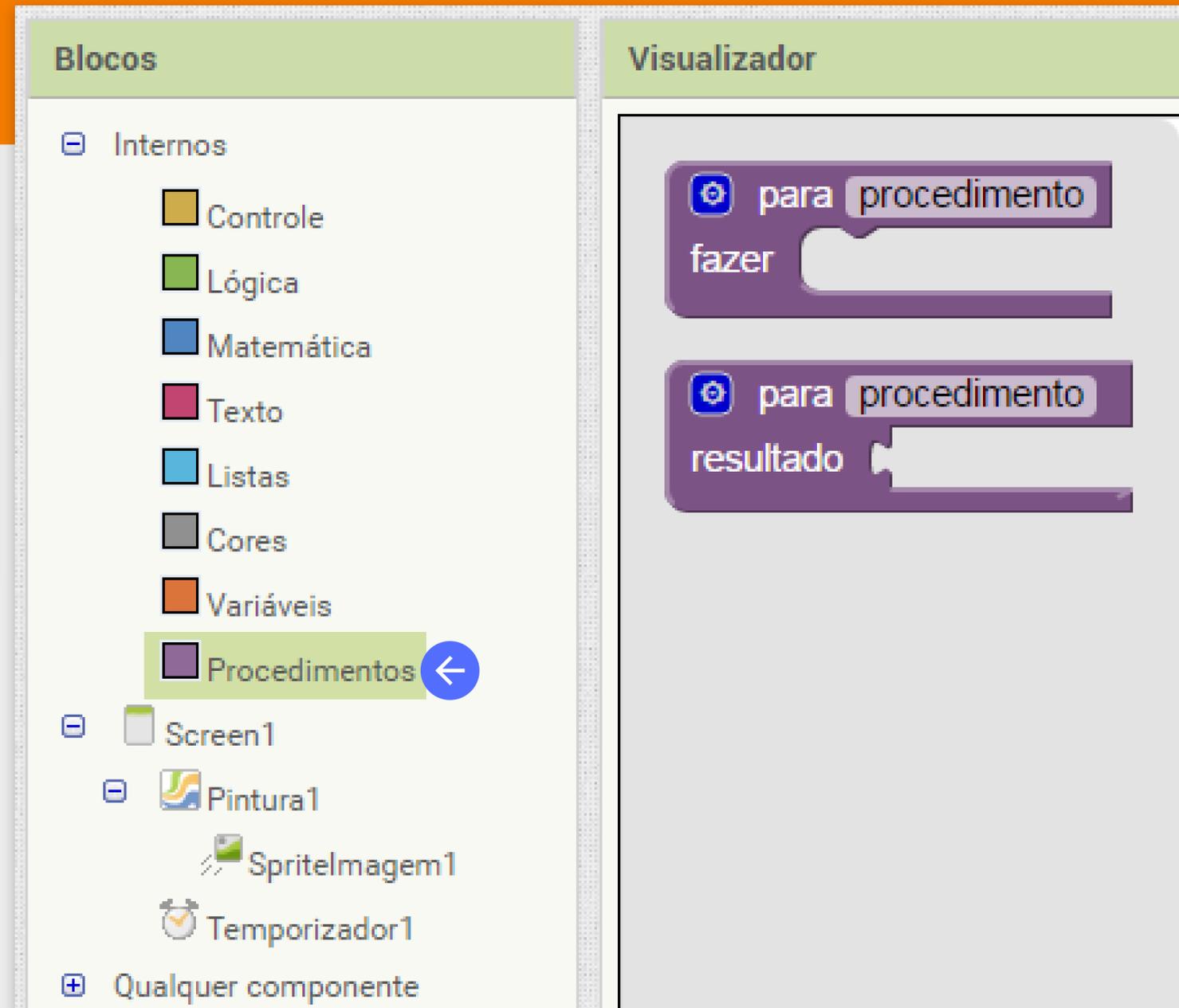
Até então estávamos trabalhando na tela de Design do nosso aplicativo, agora vamos passar para a tela de programação, ou **Blocos**.



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

Blocos: A list of block categories is shown. The 'Procedimentos' (Procedures) category is highlighted in green and has a blue arrow pointing to it. Other categories include Internos, Controle, Lógica, Matemática, Texto, Listas, Cores, Variáveis, Screen1, Pintura1, Spritemagem1, Temporizador1, and Qualquer componente.

Visualizador: Two 'para procedimento' (for procedure) blocks are visible. The first block is labeled 'fazer' (do) and the second is labeled 'resultado' (result).

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

The screenshot shows a visual programming environment. On the left, a palette titled 'Visualizador' contains three blocks: a purple 'para procedimento fazer' block, a purple 'para procedimento resultado' block, and a purple 'chamar MoverMosquito' block. On the right, a workspace contains a purple 'para MoverMosquito fazer' block. A dashed blue line connects the 'procedimento' label of the first block in the palette to the 'procedimento' label of the block in the workspace. A mosquito icon is visible in the top right of the workspace.

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

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

The screenshot shows the Scratch IDE interface for a project named 'PegaMosquito'. The 'Blocos' panel on the left lists various components, with 'Mosquito' selected. The 'Visualizador' panel on the right shows the 'Mosquito' component on the stage, represented by a backpack icon. The 'Scripts' area contains four event-driven code blocks for the 'Mosquito' component:

- quando Mosquito .ColidiuCom**: Triggered when the mosquito collides with an object. It has a 'fazer' block attached.
- quando Mosquito .Arrastado**: Triggered when the mosquito is dragged. It has variables for `xInicial`, `yInicial`, `xAnterior`, `yAnterior`, `xAtual`, and `yAtual`. It has a 'fazer' block attached.
- quando Mosquito .AlcançouBorda**: Triggered when the mosquito reaches the edge of the stage. It has a 'borda' block and a 'fazer' block attached.
- quando Mosquito .Arremessado**: Triggered when the mosquito is thrown. It has variables for `x`, `y`, `velocidade`, `direção`, `xvel`, and `yvel`. It has a 'fazer' block attached.
- quando Mosquito .ToqueParaBaixo**: Triggered when the mosquito is touched from below. It has variables for `x` and `y`. It has a 'fazer' block attached.

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.

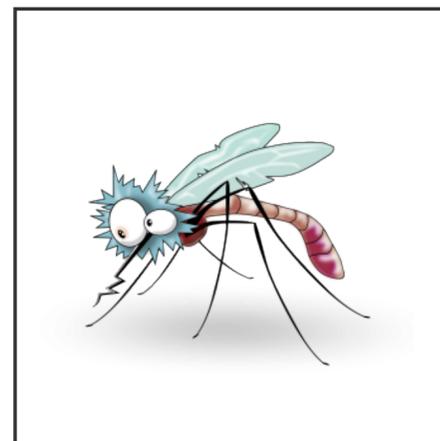
The screenshot shows the PegaMosquito software interface. The top bar includes the title "PegaMosquito" and buttons for "Screen1", "Adicionar Tela ...", and "Remover Tela". On the right, there are "Designer" and "Blocos" tabs. The left pane, labeled "Blocos", shows a tree view with categories: Internos (Controle, Lógica, Matemática, Texto, Listas, Cores, Variáveis, Procedimentos), Screen1, Pintura1 (Mosquito), and Qualquer componente. The main area is split into "Visualizador" and "Editor". The Visualizador pane displays a sequence of commands: "ajustar Mosquito . Visivel para", "Mosquito . Largura", "ajustar Mosquito . Largura para", "Mosquito . X", "ajustar Mosquito . X para", "Mosquito . Y", "ajustar Mosquito . Y para", "Mosquito . Z", "ajustar Mosquito . Z para", "Mostrar Avisos", and "Mosquito". The Editor pane shows a procedure call "para MoverMosquito" with a "fazer" block containing "ajustar Mosquito . X para". A blue dashed line connects the "ajustar Mosquito . X para" block in the Visualizador to the "ajustar Mosquito . X para" block in the Editor. A vertical scrollbar is visible on the right side of the Visualizador pane.

MOVER O MOSQUITO PELA TELA

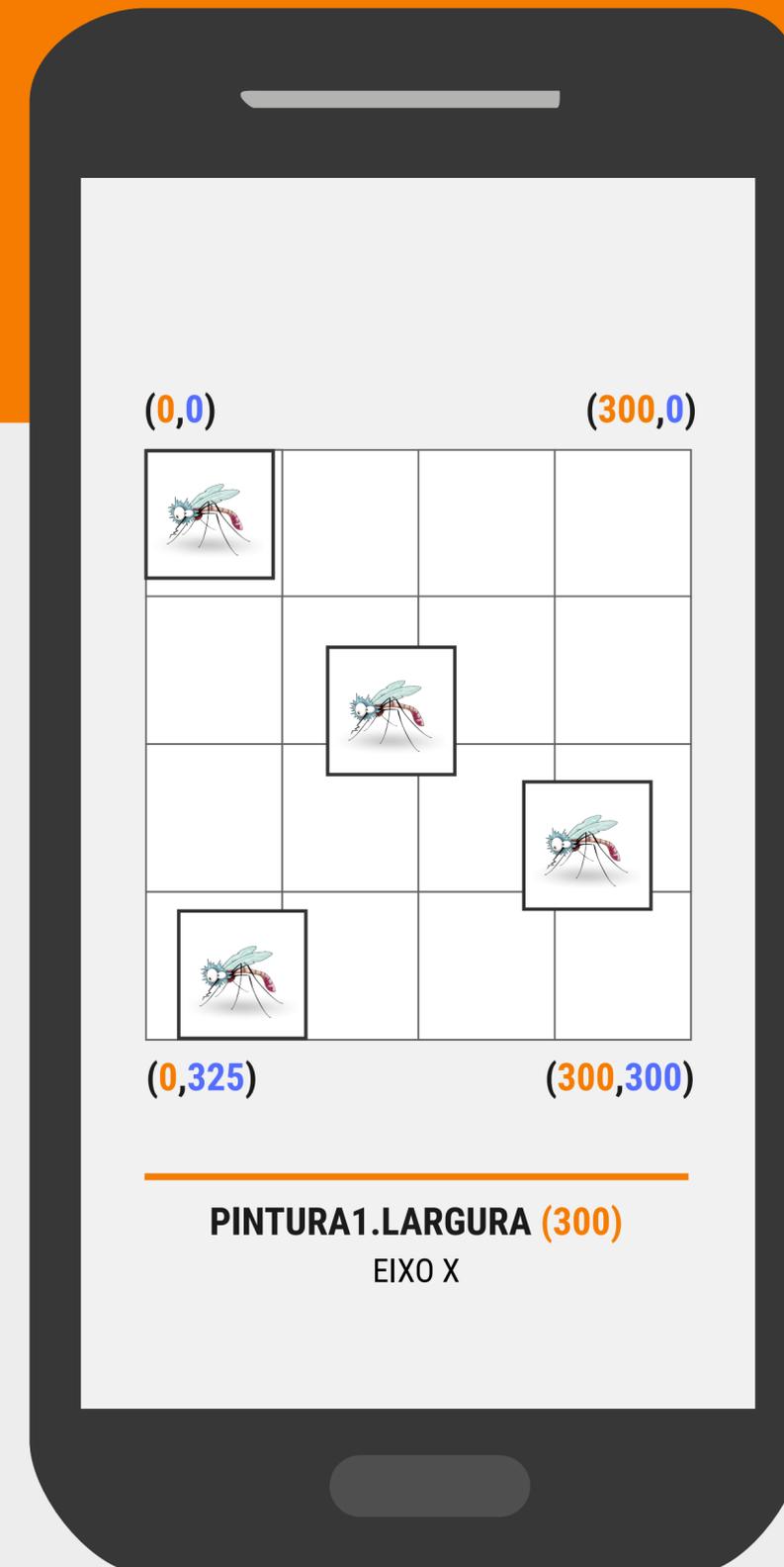
Para que posição queremos mover o mosquito?

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

MOSQUITO.ALTURA (30)



MOSQUITO.LARGURA (30)



PINTURA1.ALTURA (300)
EIXO Y

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”.

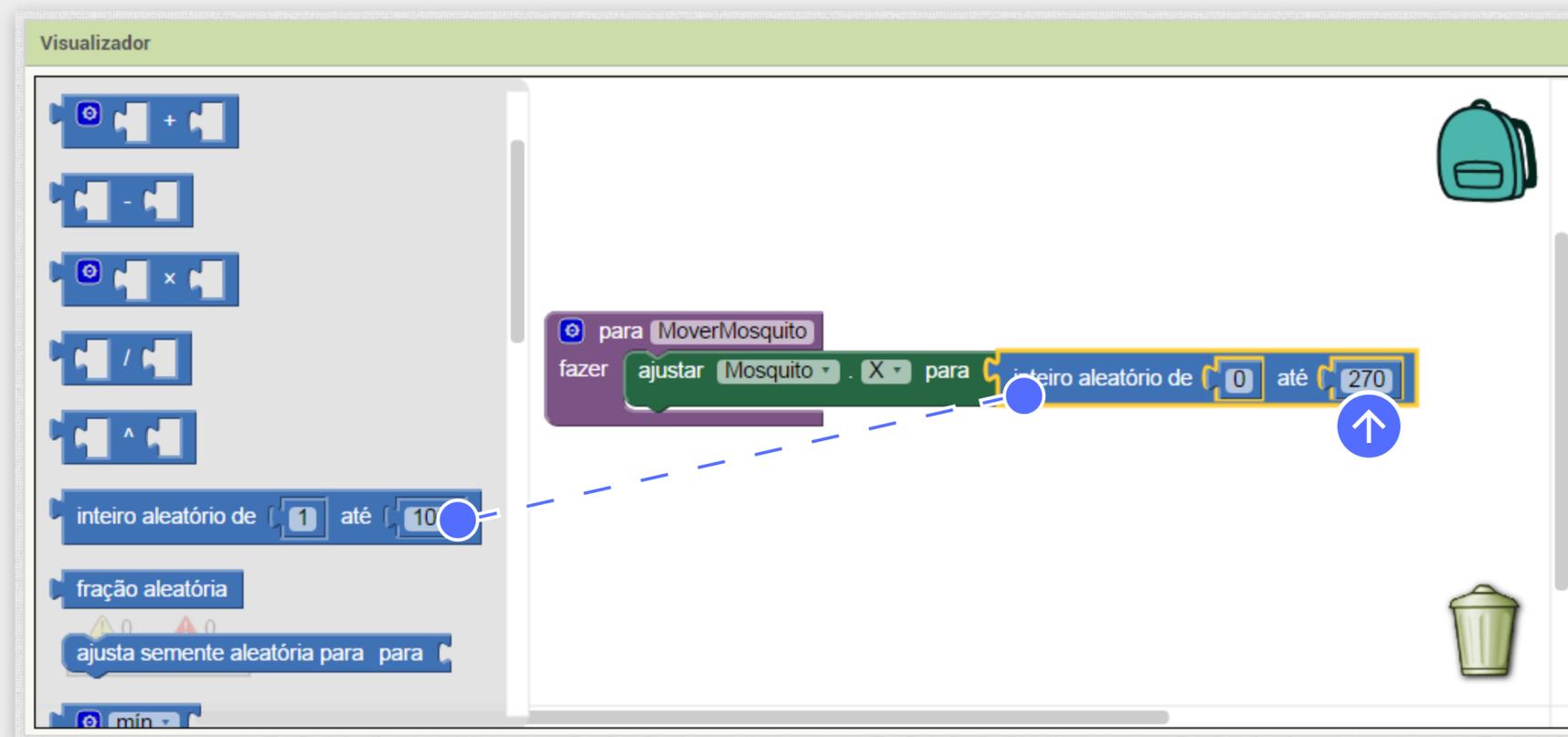
The image shows a block editor interface with two main panels: 'Blocos' (Blocks) on the left and 'Visualizador' (Viewer) on the right.

Blocos Panel: A list of block categories is shown under 'Internos'. The 'Matemática' (Mathematics) category is highlighted in green. Other categories include Controle, Lógica, Texto, Listas, Cores, Variáveis, and Procedimentos. Below this, a project structure is visible with 'Screen1', 'Pintura1' (containing a 'Mosquito' sprite), and 'Temporizador1'.

Visualizador Panel: A script is being built. The first block is a 'quando clicado' (when clicked) block. It contains a 'mover para' (move to) block with 'Mosquito' selected in the dropdown. Below it is a 'fazer ajustar' (do a turn) block with 'Mosquito' selected in the dropdown and 'X' selected in the dropdown. Further down are blocks for multiplication (x), division (/), and exponentiation (^). At the bottom, a 'inteiro aleatório de' (random integer) block is highlighted with a yellow border, with '1' in the first input field and '100' in the second input field.

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 a programming environment with two main panels: 'Blocos' (Blocks) on the left and 'Visualizador' (Viewer) on the right.

Blocos Panel: A tree view under 'Internos' (Internal) showing categories: Controle (Control), Lógica (Logic), Matemática (Mathematics), Texto (Text), Listas (Lists), Cores (Colors), Variáveis (Variables), and Procedimentos (Procedures). Below this, a project structure shows 'Screen1', 'Pintura1' (Paint) containing 'Mosquito', and 'Temporizador1' (Timer) highlighted in green. At the bottom is 'Qualquer componente' (Any component).

Visualizador Panel: A script area showing a sequence of blocks:

- A yellow 'quando Temporizador1 .Disparo' (when Timer1 ticks) block.
- A 'fazer' (do) block containing:
 - A purple 'chamar Temporizador1 .Add Days' (call Timer1 Add Days) block with 'instante' (immediately) and 'quantity'.
 - A purple 'chamar Temporizador1 .Add Duration' (call Timer1 Add Duration) block with 'instante' and 'quantity'.
 - A purple 'chamar Temporizador1 .Add Hours' (call Timer1 Add Hours) block with 'instante' and 'quantity'.
 - A purple 'chamar Temporizador1 .Add Minutes' (call Timer1 Add Minutes) block with 'instante'.

At the bottom of the script area, there are two warning icons (yellow and red triangles) with a '0' next to each.

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.

The image shows the Scratch IDE interface. On the left is the 'Blocos' (Blocks) panel, and on the right is the 'Visualizador' (Scripts) panel.

Blocos Panel:

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

Visualizador Panel:

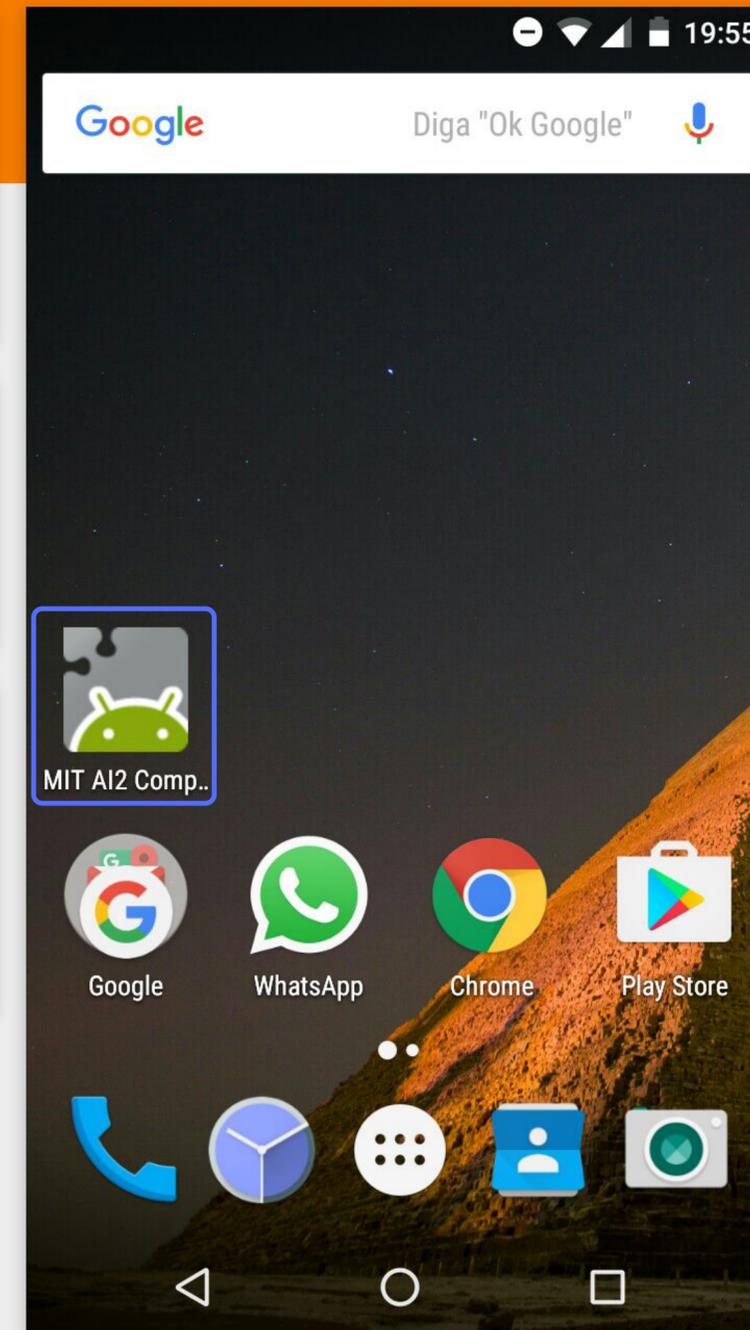
- Script 1: 'para MoverMosquito' loop containing 'ajustar Mosquito . X'.
- Script 2: 'quando Temporizador1 .Disparo' event block containing 'chamar MoverMosquito'.

MOVER O MOSQUITO PELA TELA

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

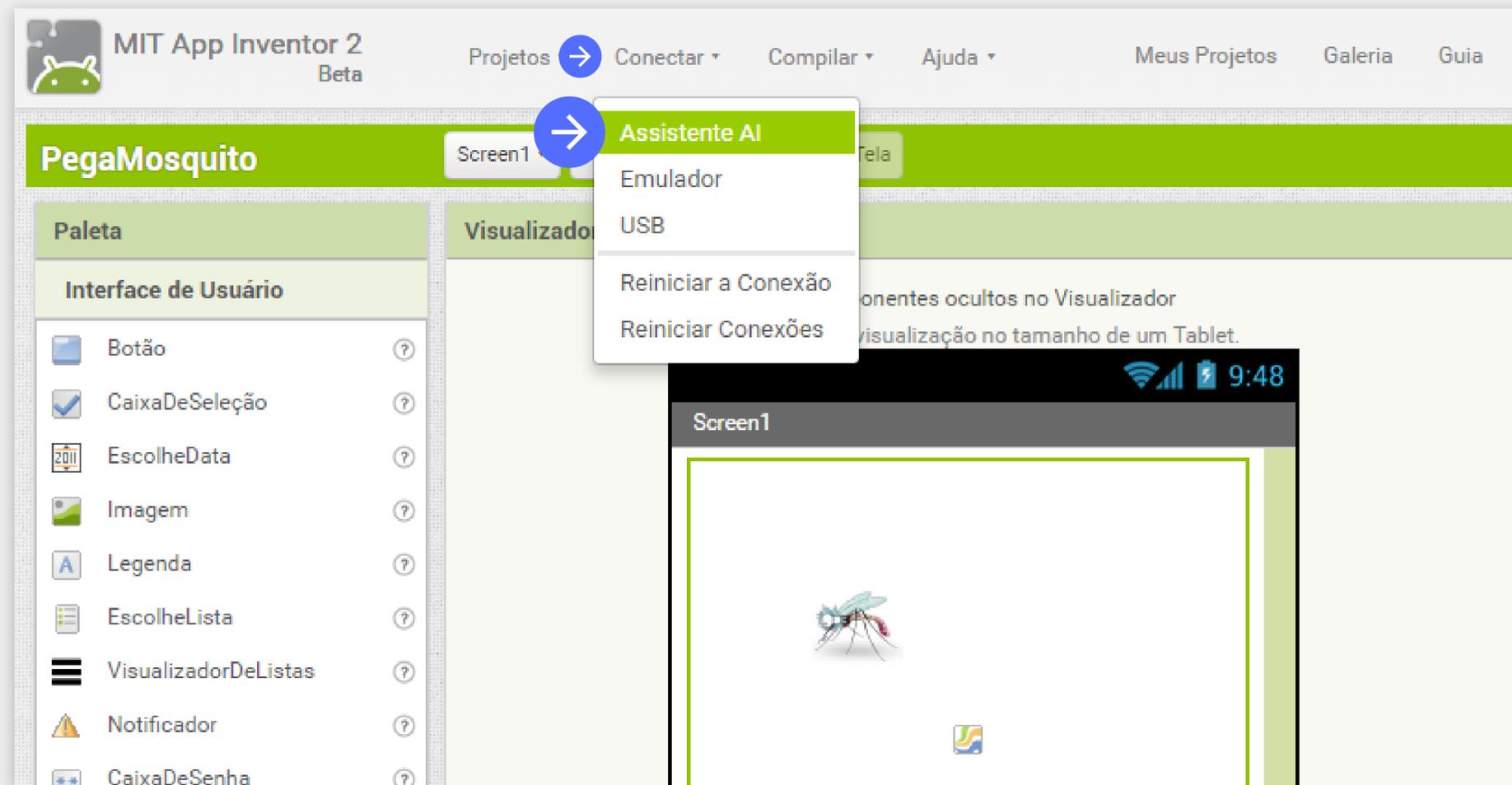
Vamos testar no celular como está ficando o jogo!

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.

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

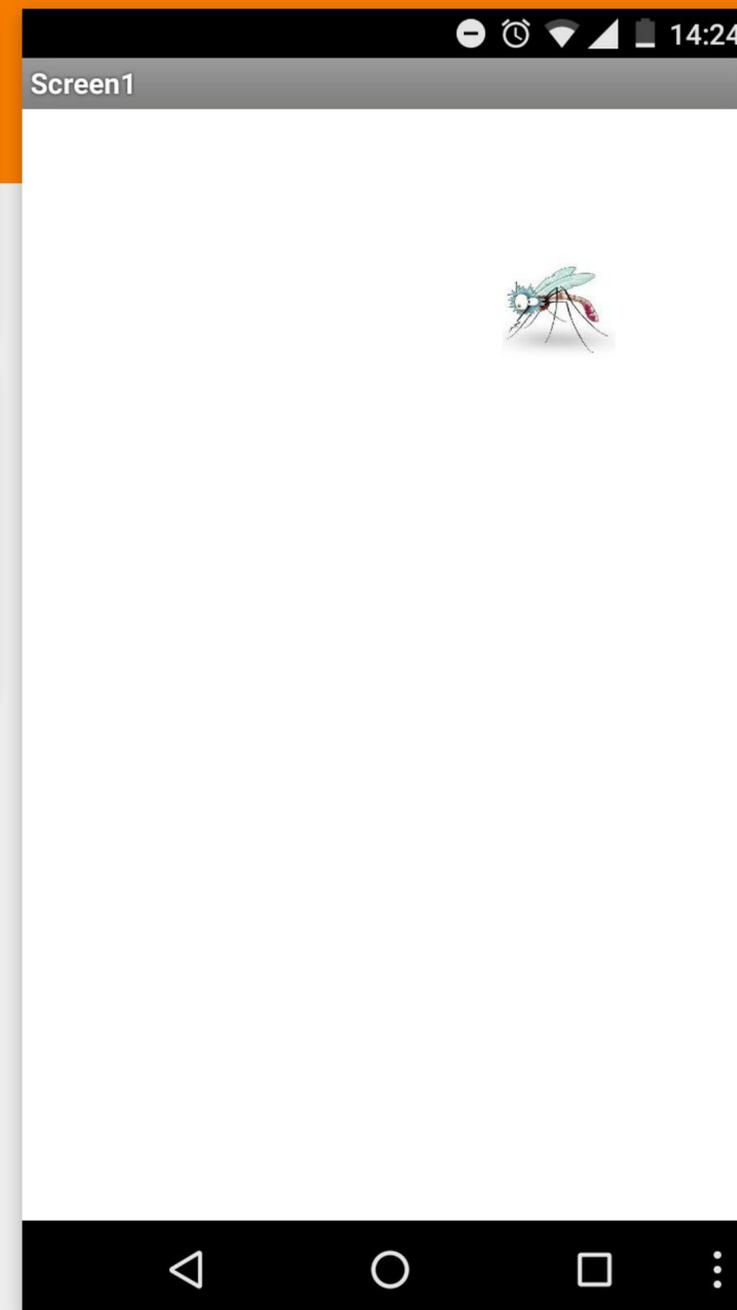
DICA

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.

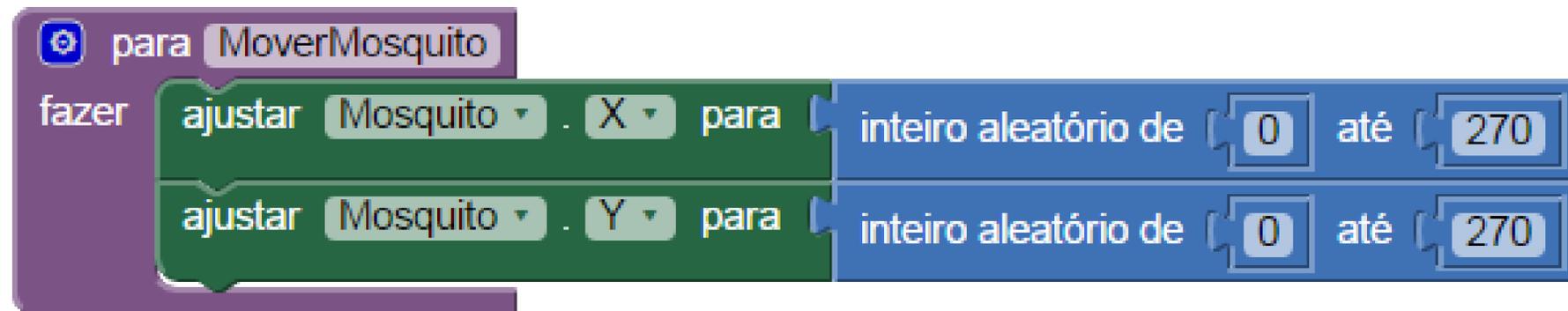


MOVER O MOSQUITO PELA TELA

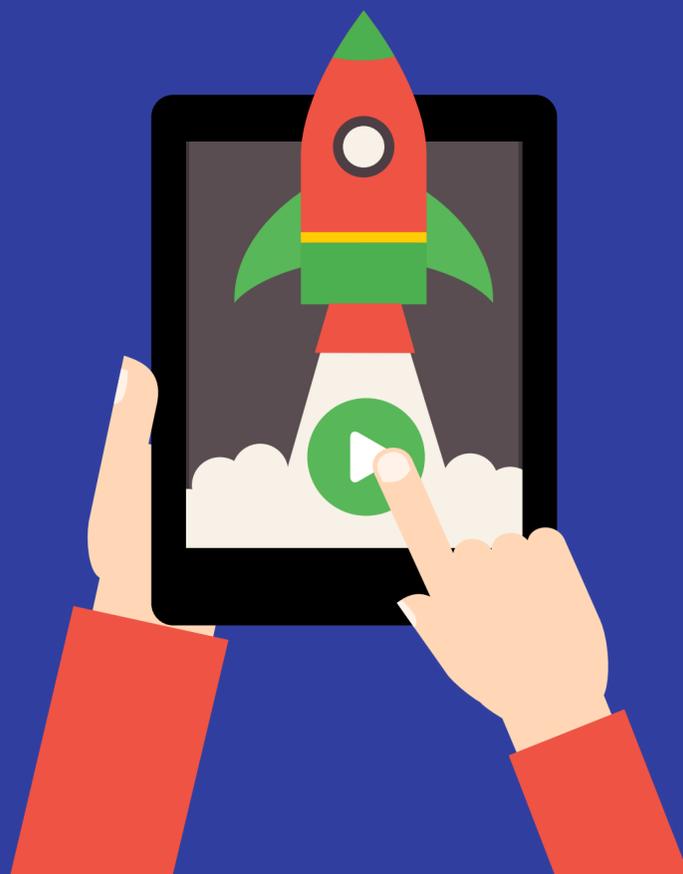
Vamos seguir os mesmos passos anteriores para alterar também a coordenada Y do **Mosquito**.

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.

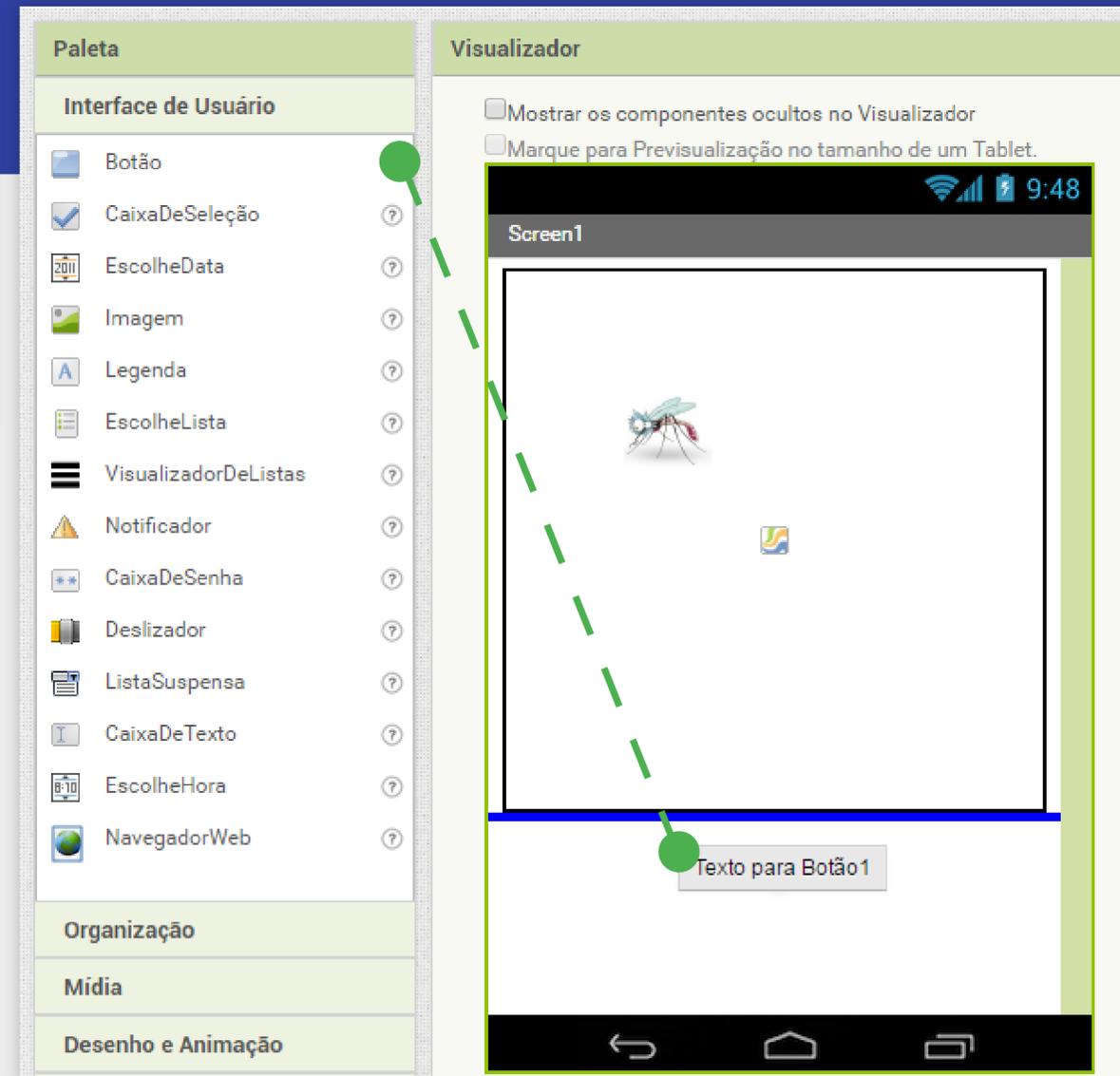
O mosquito está se movimentando a cada disparo do **Temporizador1**. Contudo precisamos que isso seja realizado somente quando formos jogar. Para isto, vamos inserir um botão de **Iniciar Jogo**.

The screenshot shows the Visual Studio Designer interface with two panels: Componentes and Propriedades. The Componentes panel displays a tree view with the following items: Screen1, Pintura1, Mosquito, BotãoIniciar, and Temporizador1. The Propriedades panel shows the properties for Temporizador1, including DisparosContínuos (checked), Ativado (unchecked), and Intervalo (500).

Componentes	Propriedades
Screen1	Temporizador1
Pintura1	DisparosContínuos
Mosquito	<input checked="" type="checkbox"/>
BotãoIniciar	Ativado
Temporizador1	<input type="checkbox"/>
	Intervalo
	500

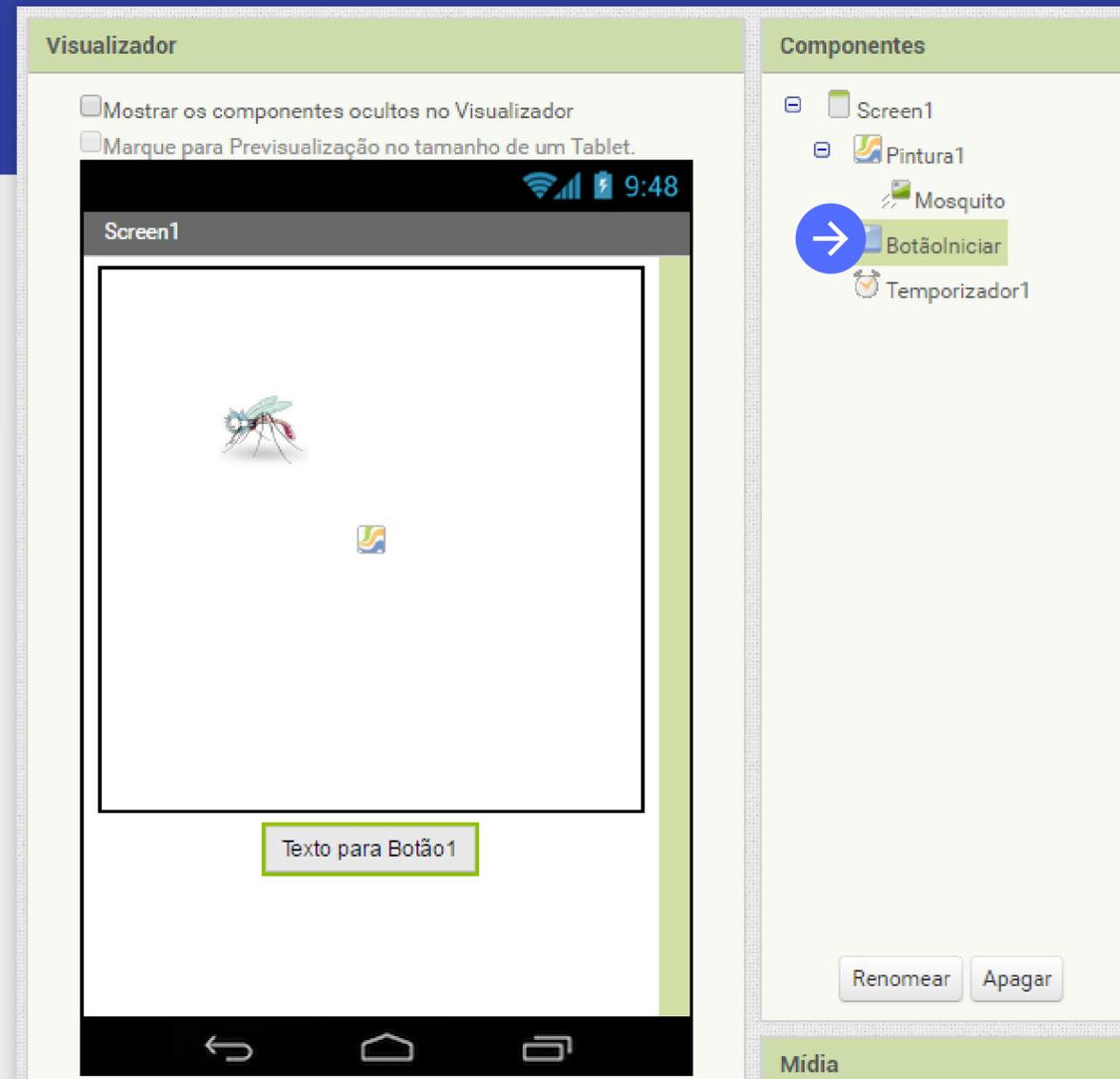
INICIAR O JOGO

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**.



The screenshot displays the Android Studio interface. On the left, the 'Visualizador' (Visualizer) panel shows a preview of the application screen. The screen contains a mosquito icon, a small Android logo, and a button labeled 'Texto para Botão1'. On the right, the 'Componentes' (Components) panel lists the UI elements: 'Screen1', 'Pintura1', 'Mosquito', 'BotãoIniciar' (highlighted with a blue arrow), and 'Temporizador1'. At the bottom of the components panel are 'Renomear' (Rename) and 'Apagar' (Delete) buttons. The 'Mídia' (Media) panel is visible at the very bottom.

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!

The screenshot displays the Xcode IDE interface for developing a mobile application. It is divided into three main panels:

- Visualizador (Preview):** Shows a mobile app preview with a mosquito icon and a button labeled "Iniciar Jogo".
- Componentes (Component Inspector):** Shows a hierarchy of components: Screen1, Pintura1, Mosquito, BotãoIniciar, and Temporizador1.
- Propriedades (Property Inspector):** Shows the properties for the selected "BotãoIniciar" component, including "CorDeFundo", "Ativado", "FonteNegrito", "Fonteltálico", "TamanhoDaFonte", "FamíliaDaFonte", "Altura", "Largura", "Imagem", "Forma", "MostrarFeedback", and "Texto". The "Texto" property is set to "Iniciar Jogo".

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**.

The screenshot displays the Scratch programming environment. On the left, the 'Blocos' (Blocks) panel is open, showing a tree view of the project's components. Under 'Internos', there are categories like 'Controle', 'Lógica', 'Matemática', 'Texto', 'Listas', 'Cores', 'Variáveis', and 'Procedimentos'. Under 'Screen1', there are 'Pintura1', 'Mosquito', 'BotãoIniciar', and 'Temporizador1'. The 'BotãoIniciar' block is highlighted. At the bottom of the 'Blocos' panel are 'Renomear' and 'Apagar' buttons. On the right, the 'Visualizador' (Inspector) panel shows the event-driven programming blocks for the 'BotãoIniciar' object. The blocks are: 'quando BotãoIniciar .Clique' (when clicked), 'quando BotãoIniciar .RecebeuFoco' (when focus is received), 'quando BotãoIniciar .CliqueLongo' (when clicked and held), 'quando BotãoIniciar .PerdeuFoco' (when focus is lost), 'quando BotãoIniciar .ToqueParaBaixo' (when touched from below), and 'quando BotãoIniciar .ToqueParaCima' (when touched from above). Each block has a 'fazer' (do) field. At the bottom of the 'Visualizador' panel, there is a 'BotãoIniciar . CorDeFundo' block. The status bar at the very bottom shows two warning icons with a '0' next to each.

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. On the left is the 'Blocos' (Blocks) panel, and on the right is the 'Visualizador' (Scripts) panel.

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:

- Script 1: **para** MoverMosquito
 - fazer
 - ajustar Mosquito . X para inteiro aleatório
 - ajustar Mosquito . Y para inteiro aleatório
- Script 2: **quando** Temporizador1 .Disparo
 - fazer
 - chamar MoverMosquito
- Script 3: **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.

The image shows two panels from the Scratch interface. The left panel, titled 'Blocos', contains a list of block categories: Internos, Controle, Lógica (highlighted), Matemática, Texto, Listas, Cores, Variáveis, Procedimentos, and Screen1. Below these is the option 'Qualquer componente'. The right panel, titled 'Visualizador', shows a vertical stack of logic blocks: 'verdadeiro', 'falso', 'não', an equals sign (=), 'e', and 'ou'.

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 a software development environment with two main panels: 'Blocos' (Blocks) and 'Visualizador' (Stage).

Blocos (Blocks): A list of block categories is shown, including Internos (Internal), Screen1, Pintura1 (Paint), Mosquito, BotãoIniciar (Start Button), Temporizador1 (Timer), and Qualquer componente (Any Component). The 'Variáveis' (Variables) category is highlighted in green.

Visualizador (Stage): A sequence of blocks is visible on the stage:

- A yellow block: 'inicializar global nome para' (Initialize global variable).
- An orange block: 'obter' (Get).
- An orange block: 'ajustar para' (Adjust).
- An orange block: 'inicializar local nome para dentro de' (Initialize local variable).
- Another orange block: 'inicializar local nome para dentro de'.

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.

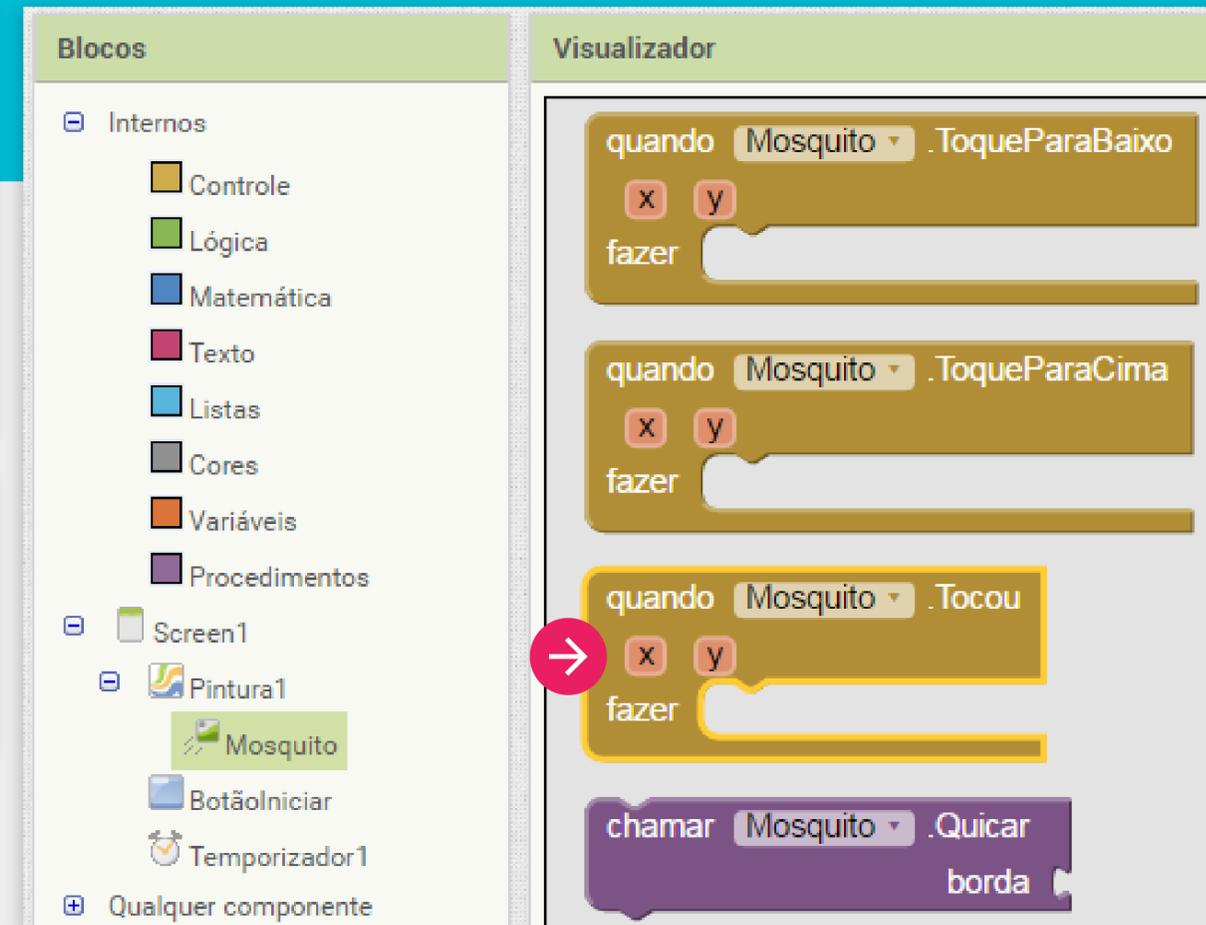
The screenshot displays the Scratch IDE interface. On the left, the 'Blocos' (Blocks) panel is visible, with the 'Matemática' (Math) category highlighted. On the right, the 'Visualizador' (Visualizer) panel shows a script area with the following blocks:

- A 'para' loop for 'MoverMosquito' containing two 'ajustar' (set) blocks for 'Mosquito . X' and 'Mosquito . Y', both set to 'inteiro al' (integer al).
- A 'quando Temporizador1 .Disparo' event block followed by a 'chamar MoverMosquito' (call MoverMosquito) block.
- A 'quando BotãoIniciar .Clique' event block followed by an 'ajustar Temporizador1 .Ativado' (set Temporizador1 .Ativado) block.
- A global variable block 'iniciar global vidaRestante para 3'.

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 screenshot displays the Scratch IDE interface, divided into two main panels: 'Blocos' (Blocks) on the left and 'Visualizador' (Viewer) on the right. The 'Blocos' panel shows a tree view of components, with 'Mosquito' selected under 'Pintura1'. The 'Visualizador' panel shows the script area for the 'Mosquito' component, containing three event blocks: 'quando Mosquito .ToqueParaBaixo', 'quando Mosquito .ToqueParaCima', and 'quando Mosquito .Tocou'. The 'quando Mosquito .Tocou' block is highlighted with a red arrow, indicating it is the focus of the current step. Below the event blocks is a 'chamar Mosquito .Quicar borda' 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 

The image shows the Scratch IDE interface. On the left, the 'Blocos' (Blocks) panel is open, showing the 'Variáveis' (Variables) category selected. On the right, the 'Visualizador' (Inspector) panel shows the code for the 'MoverMosquito' procedure. The code includes an 'ajustar Mosquito.X para' block and an 'ajustar Mosquito.Y para' block. Below this, there is a 'quando Temporizador1 .Disparo' event block that triggers a 'chamar MoverMosquito' block. Another 'quando BotãoIniciar .Clique' event block triggers an 'ajustar Temporizador1 .Ativado' block. At the bottom, there is an 'inicializar global vidaRestante para 3' block, followed by a 'quando Mosquito .Tocou' event block that triggers an 'ajustar global vidaRestante para' block. Red arrows point to the 'ajustar ... para' block in the 'Variáveis' panel and the 'ajustar global vidaRestante para' block in the script.

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:

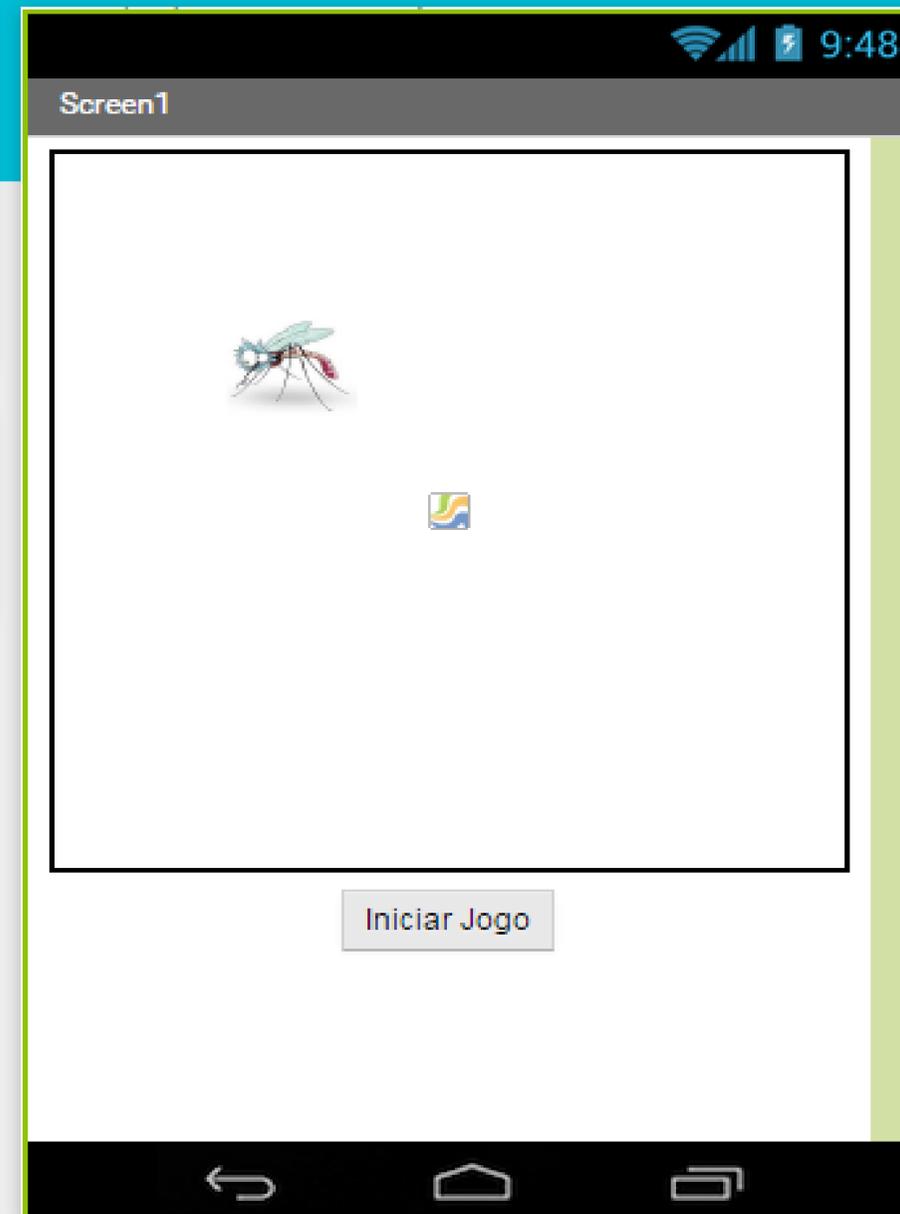
The image shows the Scratch interface. On the left, the 'Blocs' panel is open to the 'Internos' category, where the 'Variáveis' category is highlighted. On the right, the 'Visualizador' panel shows a script with three blocks: 'inicializar global nome para', 'obter', and 'ajustar para'. A red arrow points to the 'obter' block.

The image shows a Scratch script. The script starts with a 'quando Mosquito .Tocou' block, followed by a 'fazer' block containing 'ajustar global vidaRestante para' and 'obter global vidaRestante' blocks, with a '1' in a box indicating the loop count.

VIDAS DO MOSQUITO

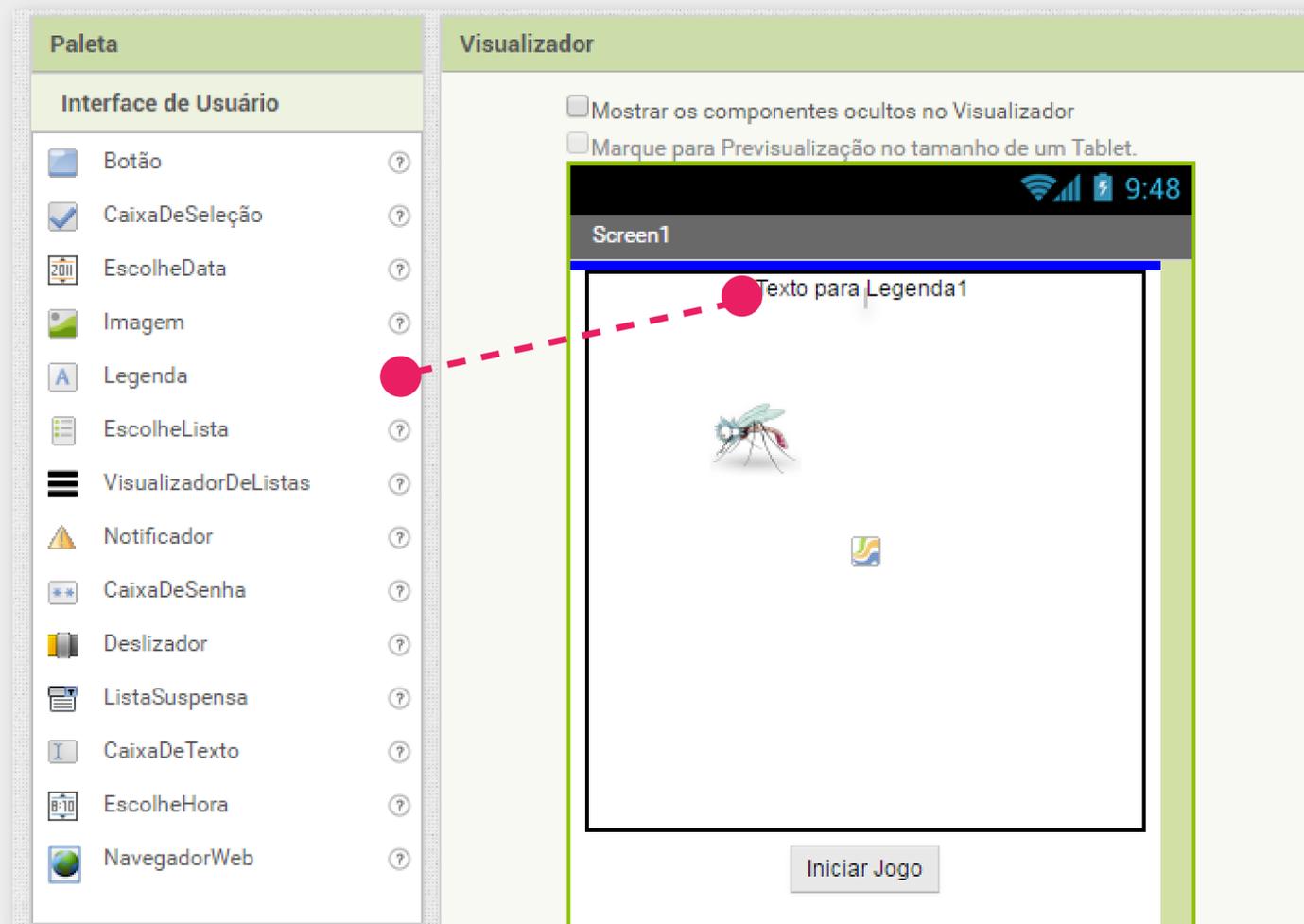
O nosso código de vida já está funcionando. Mas note que na tela do jogo não existe nada indicando para o jogador quanto é a vida do mosquito.

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 image shows a software development environment with two main panels: 'Componentes' and 'Propriedades'.

Componentes: A tree view showing a hierarchy of components. 'Screen1' is the root, containing 'Legenda1', 'Pintura1', 'Mosquito', 'BotãoIniciar', and 'Temporizador1'. A red arrow points to 'Legenda1'.

Propriedades: A panel showing the properties of the selected component, 'Legenda1'. The properties include:

- CorDeFundo: Nenhum
- FonteNegrito:
- Fonteltálico:
- TamanhoDaFonte: 14.0
- FamíliaDaFonte: padrão ▾
- TemMargens:
- Altura: Automático...
- Largura: Automático...
- Texto: Vidas: 3|
- AlinhamentoDoTexto: esquerda ▾

At the bottom of the 'Componentes' panel, there are two buttons: 'Renomear' and 'Apagar'. A red arrow points to the 'Texto' property field in the 'Propriedades' panel.

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”.

The image shows the Scratch IDE interface. On the left, the 'Blocos' (Blocks) palette is visible, with the 'Procedimentos' (Scripts) category selected. On the right, the 'Visualizador' (Inspector) window shows the script for the 'Mosquito' object. The script consists of the following blocks:

- inicializar global** (initialize global) block with the variable 'vidaRestante'.
- quando** (when) block triggered by the event 'Mosquito .Tocou' (Mosquito clicked).
- para** (loop) block containing:
 - ajustar global** (adjust global) block for the variable 'vidaRestante'.
 - chamar** (call) block for the procedure 'AtualizarVidaRestante'.

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 tree view of components. Under the 'Internos' (Internal) category, 'Legenda1' is selected and highlighted with a red arrow. On the right, the 'Visualizador' (Inspector) panel shows the 'ajustar Legenda1.Texto' block highlighted in yellow, indicating it is the active block for editing the text of the 'Legenda1' component. Other visible blocks in the 'Visualizador' panel include 'ajustar Legenda1.TemMargens para', 'Legenda1.Altura', 'ajustar Legenda1.Altura para', 'ajustar Legenda1.PercentualDeAltura para', 'Legenda1.Texto', 'ajustar Legenda1.CorDeTexto para', 'Legenda1.CorDeTexto', 'ajustar Legenda1.CorDeTexto para', and 'Legenda1.Visível'.

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**.

The screenshot displays the Scratch IDE interface. On the left, the 'Blocos' (Blocks) palette is visible, with the 'Texto' (Text) category selected. The 'Visualizador' (Viewer) pane shows a script for the 'Mosquito' object. The script consists of the following blocks:

- inicializar global** `vidaRestante` para `3`
- quando** `Mosquito` `.Tocou`
- fazer** `ajustar global vidaRestante` para `obter global vida`
- para** `AtualizarVidaRestante`
- fazer** `ajustar Legenda1 . Texto` para `juntar`

Red arrows point to the 'Texto' category in the blocks palette and the 'juntar' block in the script, highlighting the focus of the explanation.

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 programming environment with two main panels: 'Blocos' (Blocks) on the left and 'Visualizador' (Visualizer) on the right. The 'Blocos' panel lists categories like 'Internos', 'Controle', 'Lógica', 'Matemática', 'Texto', 'Listas', 'Cores', 'Variáveis', 'Procedimentos', 'Screen1', 'Legenda1', 'Pintura1', 'Mosquito', 'BotãoIniciar', and 'Temporizador1'. The 'Visualizador' panel displays a sequence of code blocks:

- A 'quando BotãoIniciar .Clique' block followed by a 'fazer' block containing 'ajustar Temporizador1 . Ativado para verdadeiro'.
- An 'inicializar global vidaRestante para 3' block.
- A 'quando Mosquito .Tocou' block with 'x' and 'y' inputs, followed by a 'fazer' block containing:
 - 'ajustar global vidaRestante para obter global vidaRestante - 1'
 - 'chamar AtualizarVidaRestante' (highlighted with a red arrow)
- A 'para AtualizarVidaRestante' block followed by a 'fazer' block containing:
 - 'ajustar Legenda1 . Texto para juntar " Vida: " obter global vidaRestante'

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!

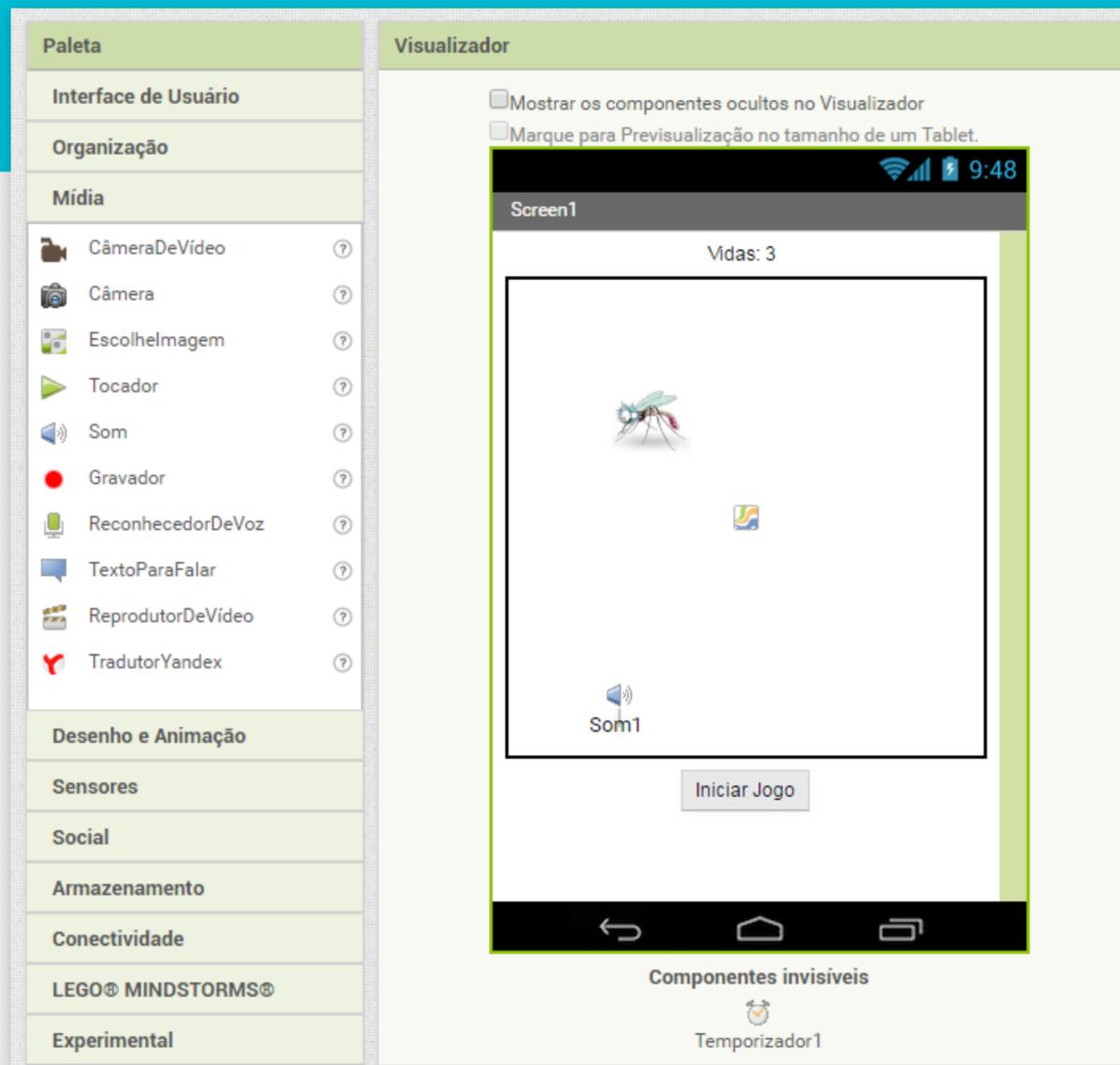
VIDAS DO MOSQUITO

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



Paleta

- Interface de Usuário
- Organização
- Mídia
 - CâmeraDeVídeo
 - Câmera
 - Escolhelimagem
 - Tocador
 - Som
 - Gravador
 - ReconhecedorDeVoz
 - TextoParaFalar
 - ReprodutorDeVídeo
 - TradutorYandex
- Desenho e Animação
- Sensores
- Social
- Armazenamento
- Conectividade
- LEGO® MINDSTORMS®
- Experimental

Visualizador

Mostrar os componentes ocultos no Visualizador

Marque para Previsualização no tamanho de um Tablet.

Screen1

Vidas: 3

Som1

Iniciar Jogo

Componentes invisíveis

Temporizador1

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**.

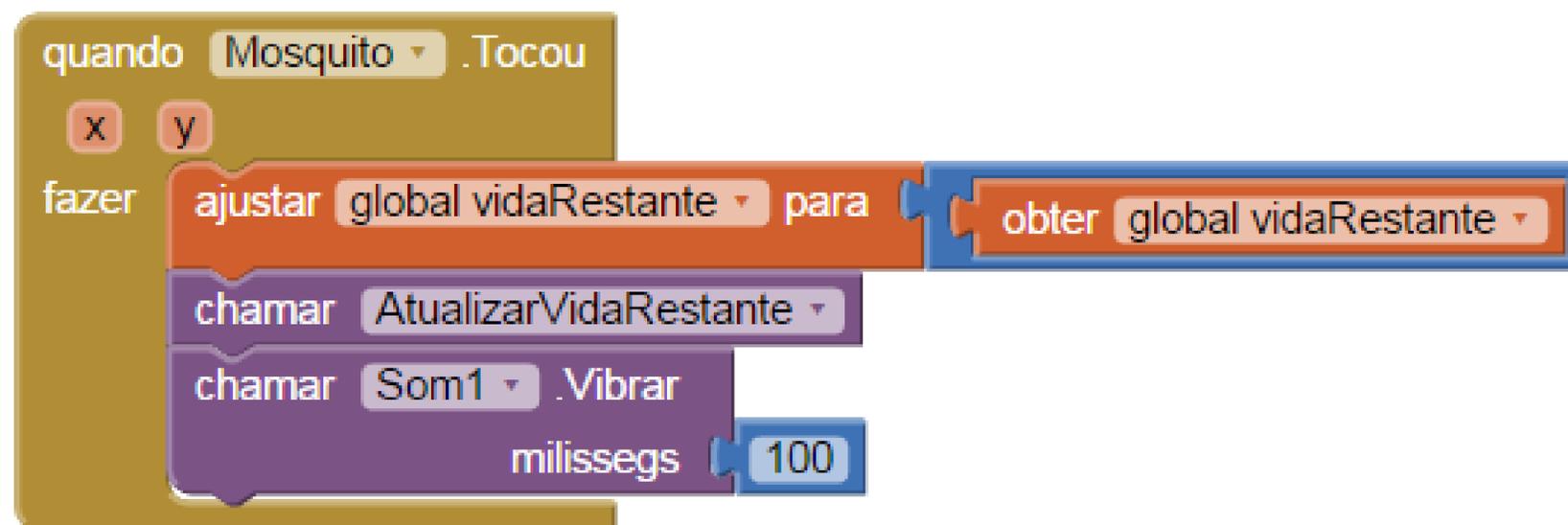
The image shows the Scratch interface with two panels: 'Blocos' (Blocks) and 'Visualizador' (Scripts).

Blocos Panel: A tree view of components. Under 'Qualquer componente' (Any component), the 'Som1' block is selected, indicated by a red arrow. Other categories include Internos, Screen1, and Pintura1.

Visualizador Panel: A script area containing several blocks. The 'chamar Som1 .Vibrar milisegs' block is highlighted with a red arrow. Other blocks include 'quando Som1 .undefined mensagem fazer', 'chamar Som1 .Pausar', 'chamar Som1 .Tocar', 'chamar Som1 .Retomar', 'chamar Som1 .Parar', 'Som1 .IntervaloMínimo', and 'ajustar Som1 .IntervaloMínimo 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

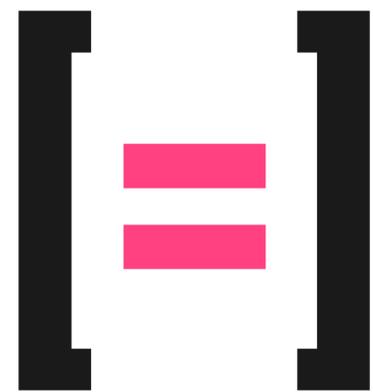
TERMINAR O JOGO



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 a Scratch-like programming environment with two main panels: 'Blocos' (Blocks) on the left and 'Visualizador' (Viewer) on the right.

Blocos (Blocks):

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

Visualizador (Viewer):

The script for the 'Mosquito' character is as follows:

```
inicializar global vidaRestante para 3

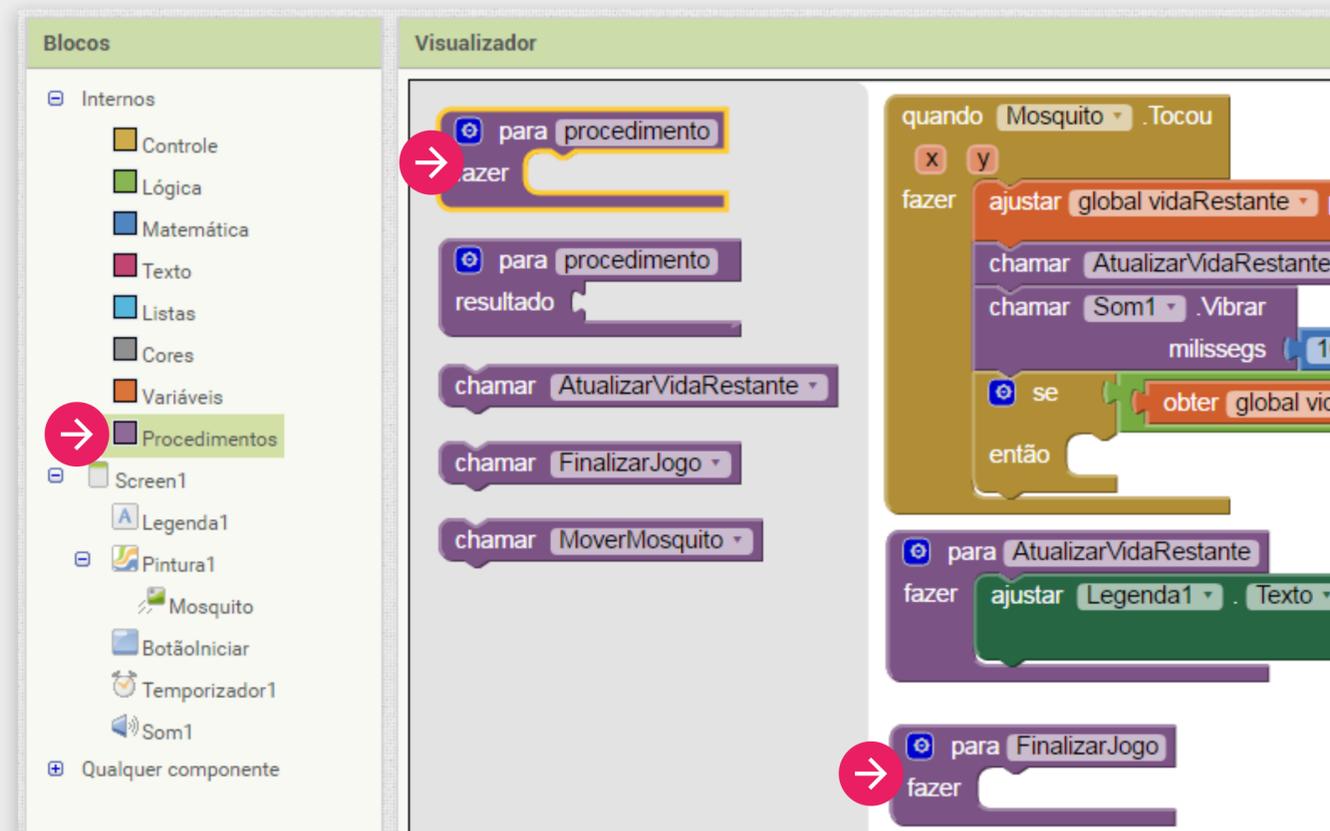
quando Mosquito .Tocou
  fazer
    ajustar global vidaRestante para obter global vidaRestante - 1
    chamar AtualizarVidaRestante
    chamar Som1 .Vibrar milisegs 100
    se obter global vidaRestante = 0
      então
        para AtualizarVidaRestante
          fazer
            ajustar Legenda1 . Texto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

Two red arrows point to the '=' comparison blocks in the script: one in the 'Lógica' category and one in the 'se' block.

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”.

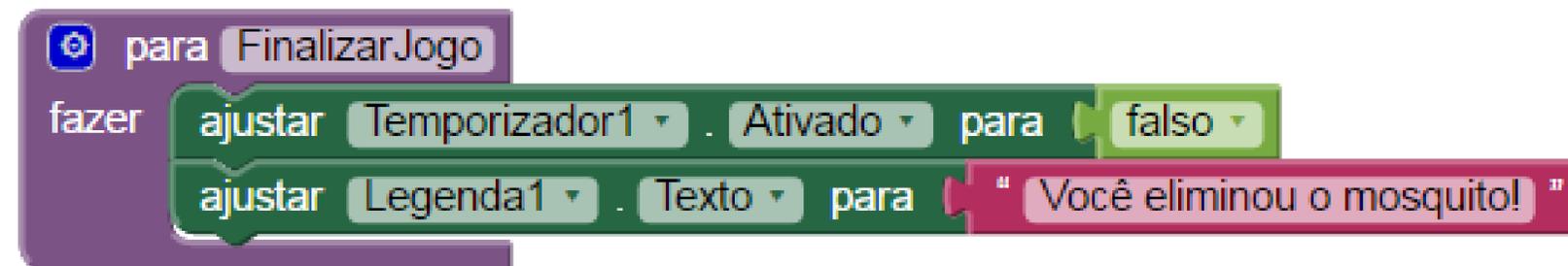


FINALIZAR O JOGO

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.

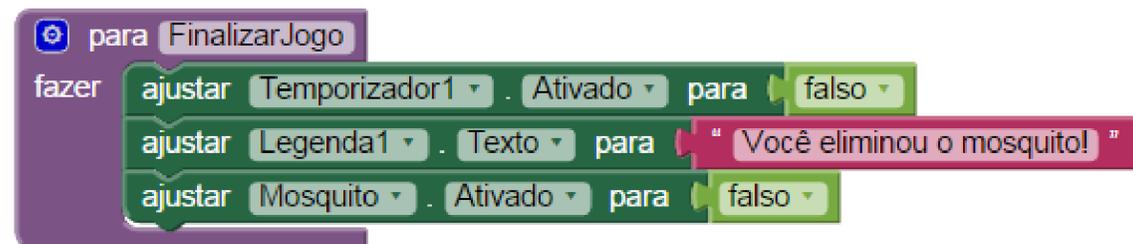


FINALIZAR O JOGO

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.



```
para FinalizarJogo
  fazer
    ajustar Temporizador1 . Ativado para falso
    ajustar Legenda1 . Texto para "Você eliminou o mosquito!"
    ajustar Mosquito . Ativado para falso
```



```
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
```

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”.

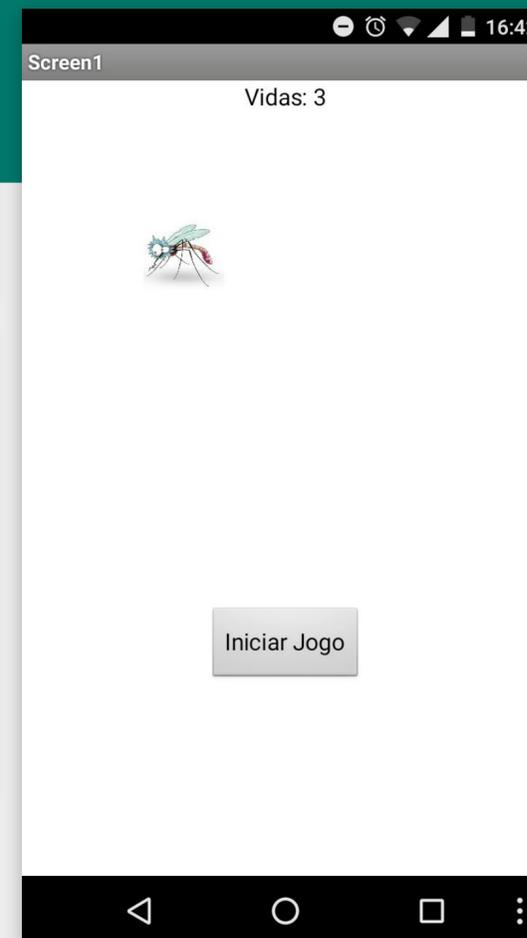
The image shows a programming environment with two main panels: 'Blocos' (Blocks) on the left and 'Visualizador' (Visualizer) on the right. The 'Blocos' panel lists various categories: Internos (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), Screen1, Legenda1, Pintura1, and Mosquito. The 'Visualizador' panel displays a script for a mosquito object. The script starts with 'inicializar global vidaRestante para 3'. It then enters a 'quando Mosquito .Tocou' event block. Inside this event, there is a 'fazer' loop block containing several actions: 'ajustar global vidaRestante para' (with a sub-block 'obter global vidaRestante'), 'chamar AtualizarVidaRestante', 'chamar Som1 .Vibrar' (with a sub-block 'milisegs 100'), and a conditional block 'se' (with a sub-block 'obter global vidaRestante = 0'). The 'então' block of the conditional contains the 'chamar FinalizarJogo' block. A red arrow points to the 'FinalizarJogo' block in the 'Blocos' panel, and another red arrow points to the 'FinalizarJogo' block inside the 'então' block in the 'Visualizador' panel.

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”.

The image shows a Scratch 'Visualizador' (Inspector) window with the following code blocks:

- Global Variable Initialization:**
 - inicializar global nome para
 - obter
 - iniciar para
 - inicializar local nome para dentro de
 - inicializar global vidaRestante para 3
- Timer Event:**
 - quando Temporizador1 .Disparo
 - fazer chamar MoverMosquito
- Button Event:**
 - quando BotãoIniciar .Clique
 - fazer ajustar Temporizador1 . Ativado para verdadeiro
 - ajustar global vidaRestante para 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 screenshot shows the Scratch programming environment. On the left is the 'Blocos' (Blocks) palette, and on the right is the 'Visualizador' (Inspector) panel. The 'Visualizador' panel displays the code for the 'Legenda1' text object. The code consists of several blocks:

- A 'quando Temporizador1 .Disparo' block.
- A 'fazer' block containing a 'chamar MoverMosquito' block.
- A 'quando BotãoIniciar .Clique' block.
- A 'fazer' block containing three sub-blocks:
 - 'ajustar Temporizador1 . Ativado para verdadeiro'.
 - 'ajustar global vidaRestante para 3'.
 - 'chamar AtualizarVidaRestante'.
- An 'inicializar global vidaRestante para 3' block.

In the 'Blocos' palette, the 'Procedimentos' (Procedures) category is selected, and the 'AtualizarVidaRestante' block is highlighted in the 'Visualizador' panel.

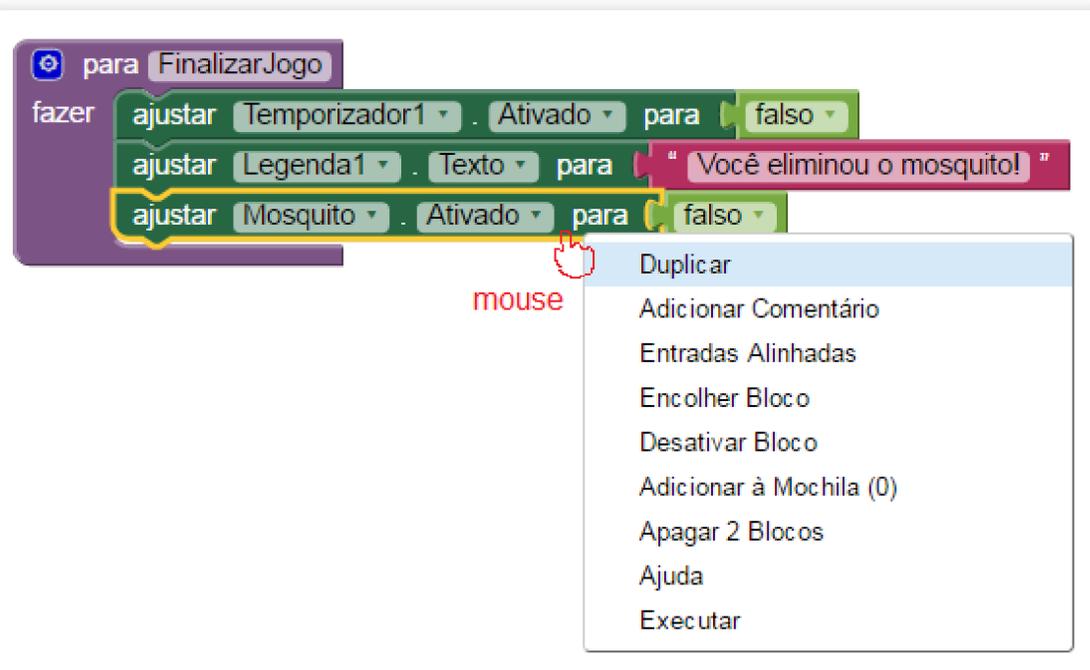
REINICIANDO O JOGO

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).

The image shows the Android Studio interface in Designer mode. On the left, the 'Paleta' (Palette) is visible, categorized into 'Interface de Usuário', 'Organização', 'Mídia', 'Desenho e Animação', and 'Sensores'. The 'Legenda' component is selected in the 'Interface de Usuário' category. On the right, the 'Visualizador' (Preview) shows a mobile app screen. The screen has a status bar at the top with the time 9:48. Below the status bar, there is a text label 'Texto para Legenda2 Vidas: 3' and a mosquito icon. At the bottom of the screen, there is a button labeled 'Iniciar Jogo'. Below the preview, the 'Componentes invisíveis' (Invisible components) section shows 'Temporizador1' and 'Som1'.

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 seleccionar o componente “Legenda2” e no campo Texto digitar “Tempo: 10”

The image shows a software interface with two main panels: 'Componentes' and 'Propriedades'. The 'Componentes' panel displays a tree view of the project's components. Under 'Screen1', the following components are listed: 'Legenda2' (selected), 'Legenda1', 'Pintura1', 'Mosquito', 'BotãoIniciar', 'Temporizador1', and 'Som1'. The 'Propriedades' panel shows the settings for the selected 'Legenda2' component. The 'Texto' property is set to 'Tempo: 10'. Other visible properties include 'CorDeFundo' (Nenhum), 'FonteNegrito' (unchecked), 'Fonteltálico' (unchecked), 'TamanhoDaFonte' (14.0), 'FamíliaDaFonte' (padrão), 'HTMLFormat' (unchecked), 'TemMargens' (checked), 'Altura' (Automático...), and 'Largura' (Automático...). At the bottom of the 'Componentes' panel, there are buttons for 'Renomear' and 'Apagar'.

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' category is expanded, showing various sensor components. A black arrow points from the 'Temporizador' component in this list to the 'Temporizador2' component on the stage.
- Visualizador:** Shows a game stage with a mosquito icon, a timer labeled 'Temporizador2', and a score of 'Vidas: 3'. A 'Tempo: 10' label is also visible. A 'Botão Iniciar' button is at the bottom.
- Componentes:** Lists the components on the stage: Screen1, Legenda2, Legenda1, Pintura1, Mosquito, Botão Iniciar, Temporizador1, Som1, and Temporizador2. A red arrow points from the 'Temporizador2' component in this list to its properties panel.
- Propriedades:** Shows the configuration for 'Temporizador2'. The 'Disparos Contínuos' checkbox is checked. The 'Ativado' checkbox is unchecked. The 'Intervalo' is set to 1000 milliseconds.

At the bottom of the interface, there is a 'Componentes invisíveis' section showing 'Temporizador1', 'Som1', and 'Temporizador2' with a green box around the latter.

ADICIONANDO TEMPO

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

```
quando BotãoIniciar .Clique
fazer
  ajustar Temporizador1 . Ativado para verdadeiro
  ajustar Temporizador2 . Ativado para verdadeiro
  ajustar global vidaRestante para 3
  chamar AtualizarVidaRestante
  ajustar Mosquito . Ativado para verdadeiro
```

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

The screenshot shows a Scratch-like block editor with two main panels: 'Blocos' (Blocks) on the left and 'Visualizador' (Viewer) on the right. The 'Blocos' panel shows a category tree with 'Variáveis' (Variables) selected. The 'Visualizador' panel shows a script area with the following code:

```
quando Temporizador1 .Disparo  
fazer chamar MoverMosquito  
  
inicializar global vidaRestante para 3  
  
inicializar global tempoRestante para 10  
  
quando BotãoIniciar .Clique  
fazer  
ajustar Temporizador1 .Ativado para verdadeiro  
ajustar Temporizador2 .Ativado para verdadeiro  
ajustar global vidaRestante para 3  
ajustar global tempoRestante para 10  
chamar AtualizarVidaRestante  
ajustar Mosquito .Ativado para verdadeiro
```

Red arrows point from the 'inicializar local' blocks in the 'Visualizador' panel to the 'inicializar global' blocks, indicating the transition from local to global variables.

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.

The image shows a Scratch-like block editor interface. On the left is the 'Blocos' (Blocks) palette with categories like 'Internos', 'Controle', 'Lógica', etc. The main workspace is the 'Visualizador' (Inspector) panel. A red arrow points to a yellow 'quando Temporizador2 .Disparo' block. Below it are four purple 'chamar Temporizador2 .Add' blocks for 'Days', 'Duration', 'Hours', and 'Minutes'. To the right, there are two green 'para' loops. The first is 'AtualizarVidaRestante' with 'ajustar' blocks for 'Legenda1 . Texto' and 'Temporizador1 . Ativado'. The second is 'FinalizarJogo' with 'ajustar' blocks for 'Temporizador1 . Ativado', 'Temporizador2 . Ativado', 'Legenda1 . Texto', and 'Mosquito . Ativado'. At the bottom right, there is another yellow 'quando Temporizador2 .Disparo' block with a 'fazer' block below it.

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

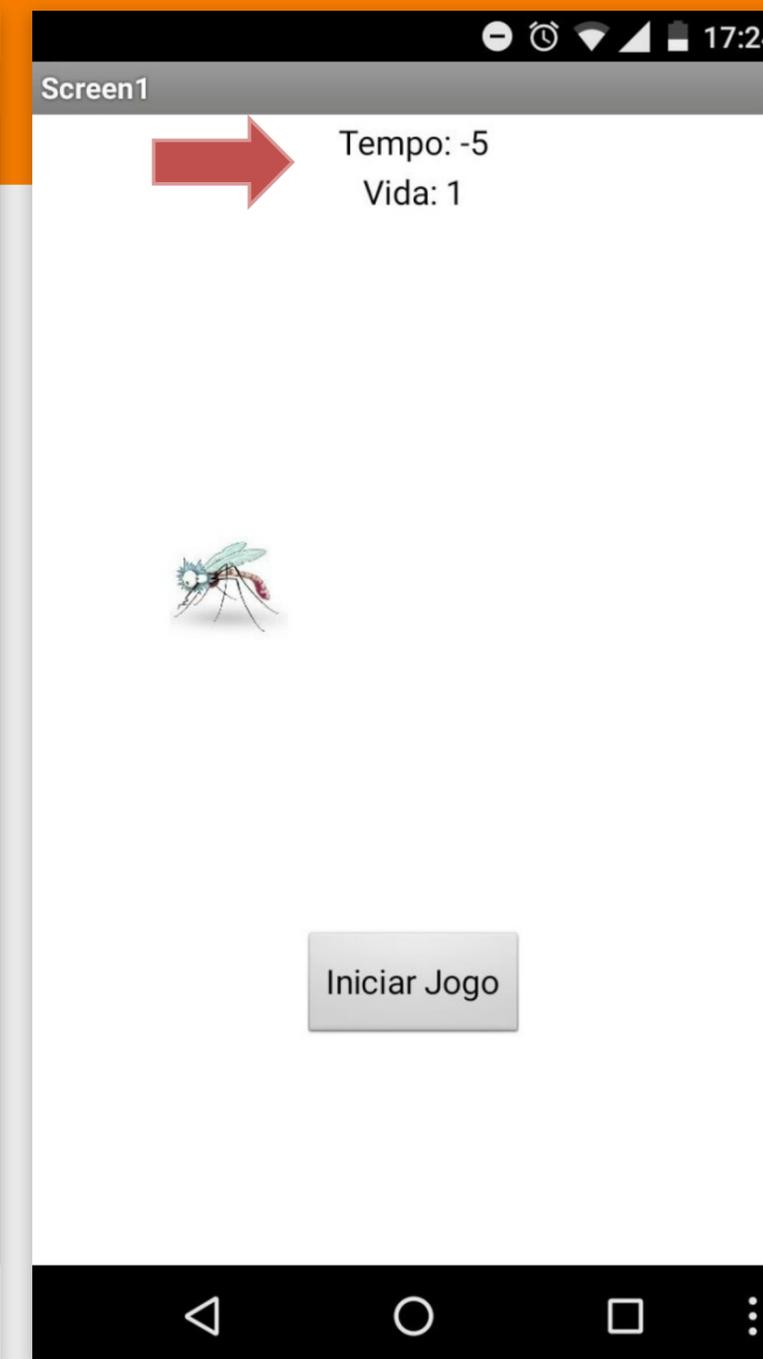
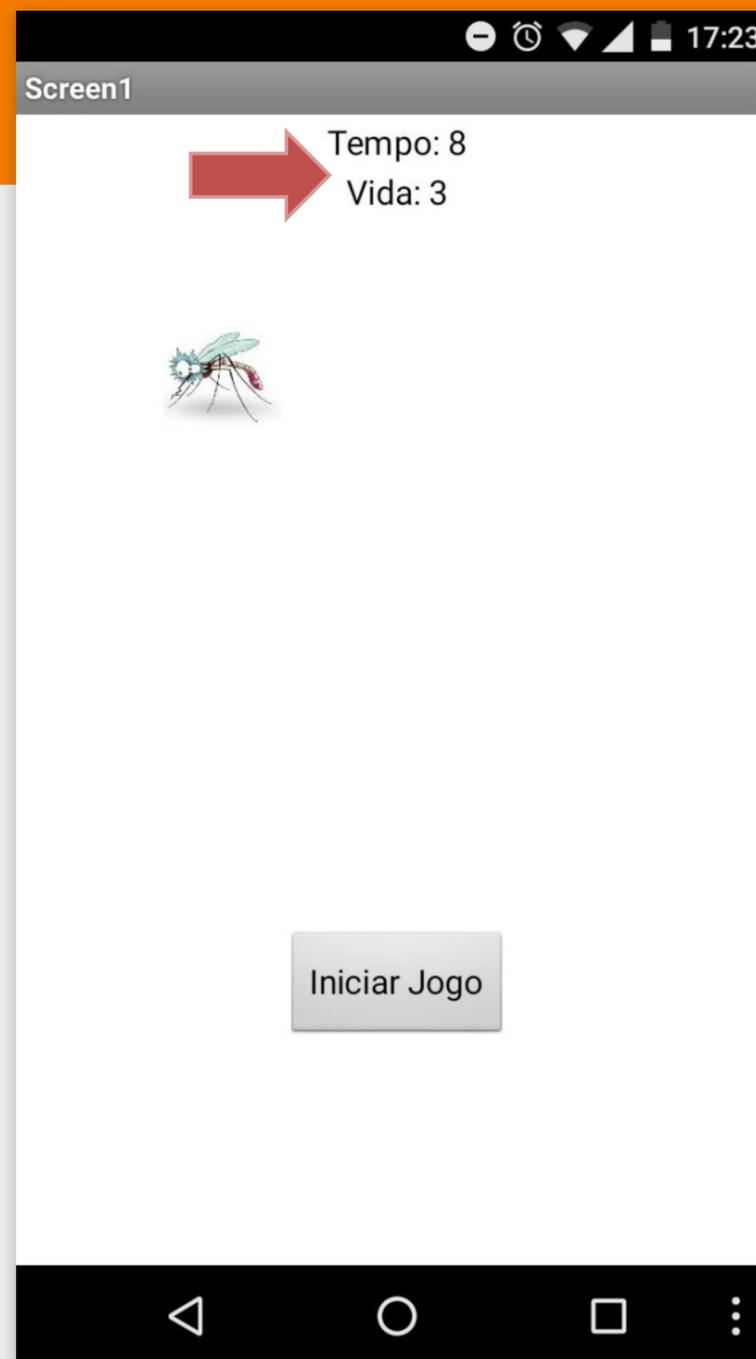


ADICIONANDO TEMPO

VAMOS TESTAR!

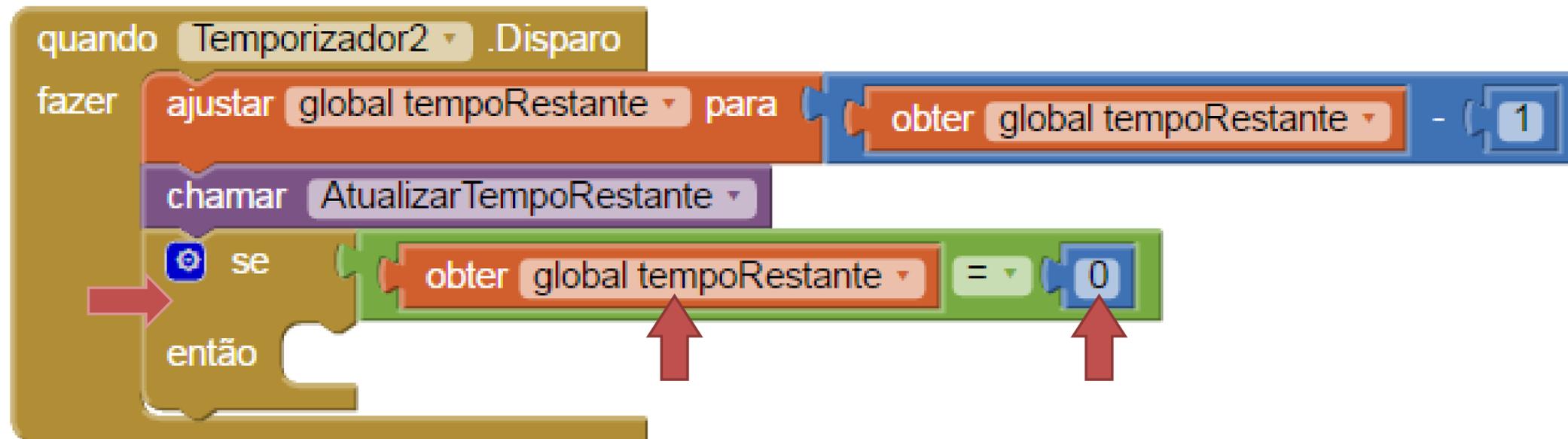
Verifique se o tempo está diminuindo na tela e se está se comportando corretamente.

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.

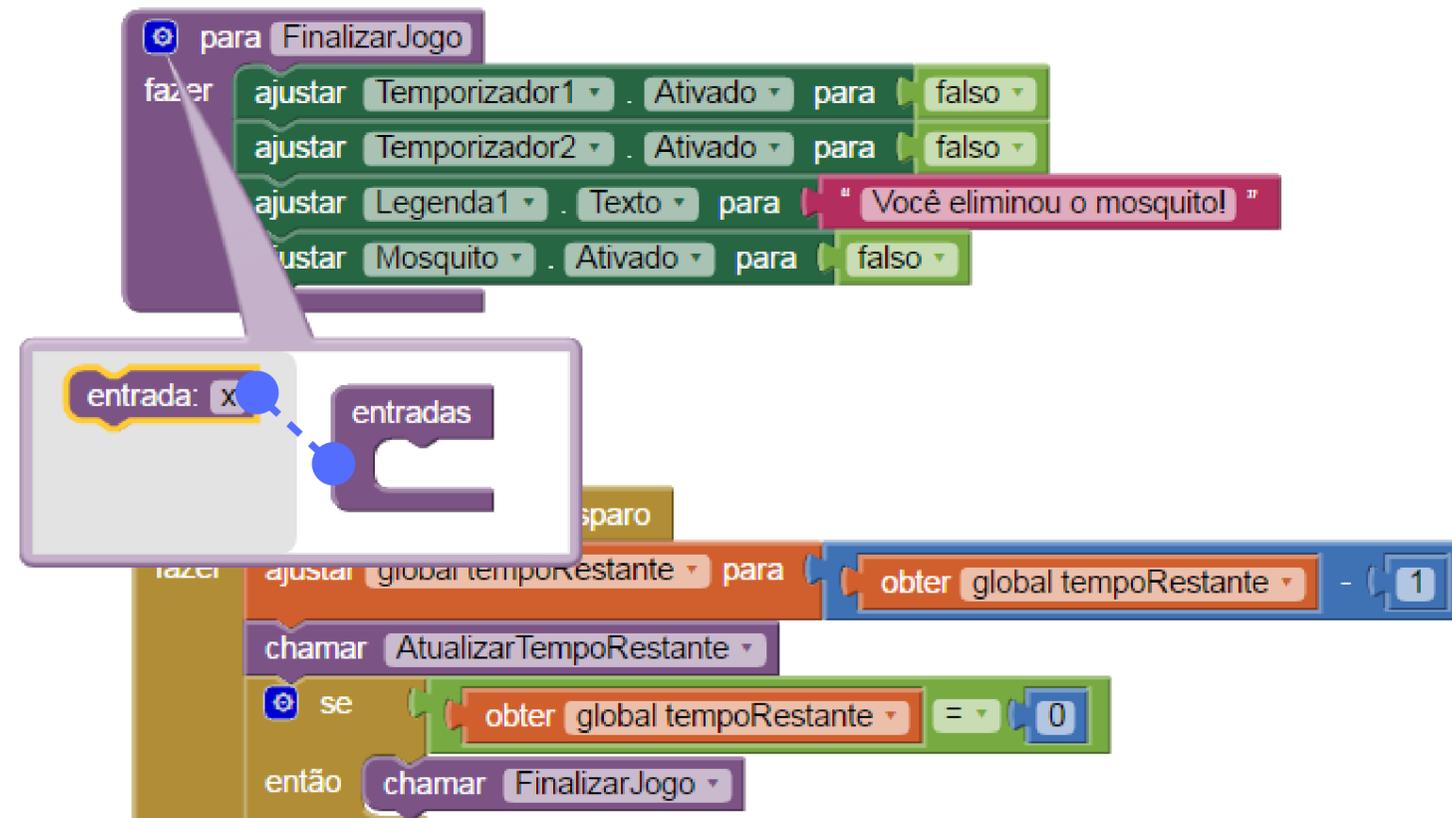
Você verificou que apesar do tempo ter se esgotado ainda apareceu que o jogador conseguiu eliminar o mosquito?



ADICIONANDO TEMPO

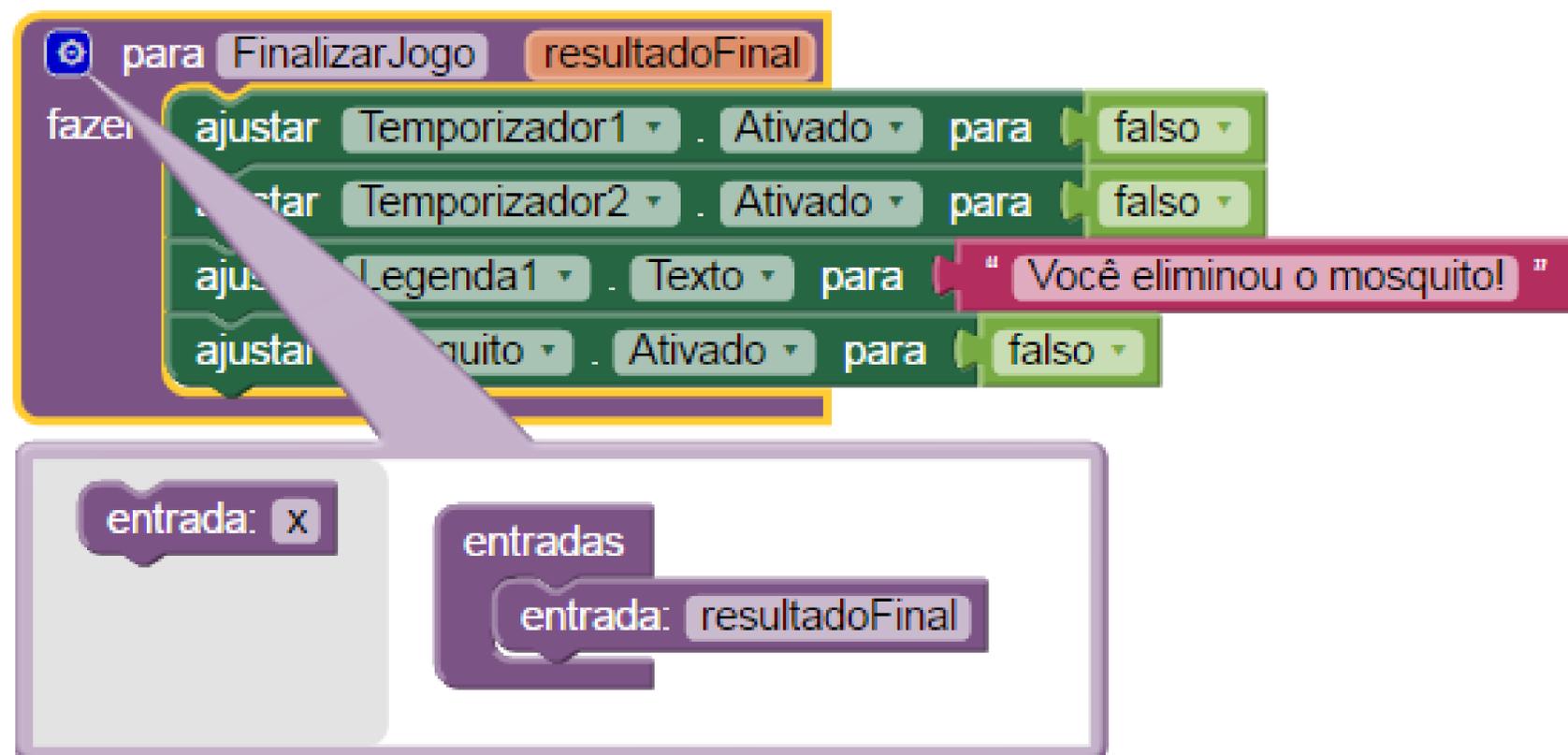
Temos que modificar o procedimento “FinalizarJogo” para receber a informação se o jogador ganhou ou perdeu, e mudar o texto da **Legenda1** corretamente.

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 (=).

The image shows a Scratch 'Visualizador' (Viewer) window. On the left, there is a palette of blocks including 'verdadeiro', 'falso', 'não', and logical comparison blocks like '=', 'e', and 'ou'. The main workspace contains two code blocks:

- A purple 'para' block labeled 'AtualizarVidaRestante' with a 'fazer' loop containing:
 - A green 'ajustar' block for 'Legenda1' . 'Texto' para a 'juntar' block.
 - The 'juntar' block contains the text 'Vida: ' and an 'obter' block for 'global vidaRestante'.
- A purple 'para' block labeled 'FinalizarJogo' with a 'resultadoFinal' variable and a 'fazer' loop containing:
 - A 'se então' block with a red arrow pointing to an '=' comparison block.
 - An 'então' block with a red arrow pointing to an 'ajustar' block for 'Temporizador1' . 'Ativado' para 'falso'.
 - A partially visible 'ajustar' block for 'Temporizador2' . 'Ativado' para 'falso'.

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**.

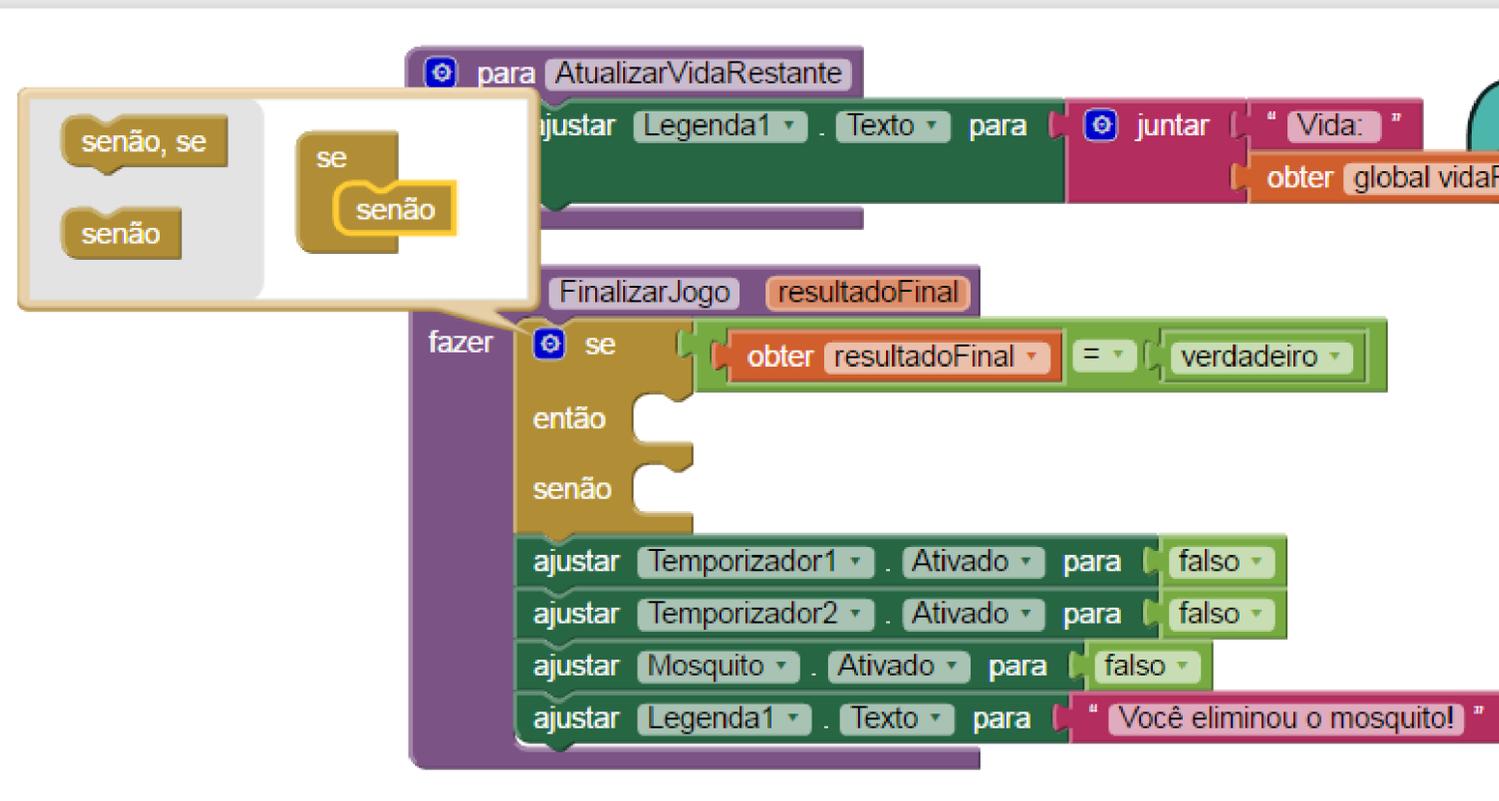


```
para FinalizarJogo resultadoFinal
fazer
  se obter resultadoFinal = verdadeiro
  então
    ajustar Temporizador1 . Ativado para falso
    ajustar Temporizador2 . Ativado para falso
    ajustar Mosquito . Ativado para falso
    ajustar Legenda1 . Texto para "Você eliminou o mosquito!"
```

The image shows a Scratch code block for a procedure named 'FinalizarJogo' with a local variable 'resultadoFinal'. The code is written in Portuguese and includes a 'fazer' loop containing an 'se' (if) condition. The condition is 'obter resultadoFinal = verdadeiro'. A red arrow points to the 'obter resultadoFinal' block. Below the 'se' block, there are four 'ajustar' (set) blocks: 'ajustar Temporizador1 . Ativado para falso', 'ajustar Temporizador2 . Ativado para falso', 'ajustar Mosquito . Ativado para falso', and 'ajustar Legenda1 . Texto para "Você eliminou o mosquito!"

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.

Visualizador

```
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 chamar FinalizarJogo resultadoFinal verdadeiro
```

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.

The screenshot shows the 'Visualizador' (Inspector) window in Scratch. The code block for 'quando Temporizador2.Disparo' (when timer2 ticks) is expanded. It contains the following blocks: 'ajustar Mosquito . Ativado para falso' (set Mosquito's 'Ativado' property to false), 'fazer' (do) loop containing: 'ajustar global tempoRestante para obter global tempoRestante - 1' (set global 'tempoRestante' to global 'tempoRestante' minus 1), 'chamar AtualizarTempoRestante' (call 'AtualizarTempoRestante'), 'se' (if) block with 'obter global tempoRestante = 0' (get global 'tempoRestante' equals 0) and 'então' (then) block with 'chamar FinalizarJogo resultadoFinal falso' (call 'FinalizarJogo' with parameter 'resultadoFinal' set to false). A red arrow points to the 'falso' parameter in the 'FinalizarJogo' block.



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**.

The image shows a screenshot of the Scratch IDE interface. On the left is the 'Paleta' (Palette) with the following categories: Interface de Usuário, Organização, Mídia, Desenho e Animação, Sensores, Social, Armazenamento, Conectividade, LEGO® MINDSTORMS®, Experimental, and Extension. The 'Organização' category is expanded, showing 'OrganizaçãoHorizontal', 'OrganizaçãoEmTabela', and 'OrganizaçãoVertical'. On the right is the 'Visualizador' (Viewer) window. It has two checkboxes: 'Mostrar os componentes ocultos no Visualizador' (unchecked) and 'Marque para Previsualização no tamanho de um Tablet.' (checked). The viewer shows a mobile device screen with a status bar at the top displaying '9:48'. The screen content includes 'Tempo: 10' and 'Vidas: 3'. Below this is a large white rectangle containing a fly icon and a small square icon. At the bottom of the screen is a grey button. Below the viewer, under the heading 'Componentes invisíveis', are three icons: 'Temporizador1', 'Som1', and 'Temporizador2'.

ADICIONAR DIFICULDADE

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

The screenshot displays a mobile application development environment with three main panels: Visualizador, Componentes, and Propriedades.

- Visualizador:** Shows a game screen with a mosquito icon, a score of 0, and a timer of 10. A button labeled "Iniciar Jogo" is positioned at the bottom of the screen, highlighted with a red arrow.
- Componentes:** Lists the components used in the application, including Screen1, Legenda2, Legenda1, Pintura1, Mosquito, OrganizaçãoHorizontal1, BotãoIniciar, Temporizador1, Som1, and Temporizador2. A red arrow points to the "OrganizaçãoHorizontal1" component.
- Propriedades:** Shows the properties for the selected "OrganizaçãoHorizontal1" component. The "Altura" (Height) property is set to "20", and the "Preencher principal" (Fill) property is set to "pontos". A red arrow points to the "20" value in the height property.

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**.

The image shows a software development interface with two main panels: 'Paleta' (Palette) and 'Visualizador' (Visualizer).

Paleta (Palette):

- Interface de Usuário (User Interface):**
 - Botão (Button)
 - CaixaDeSeleção (Selection Box)
 - EscolheData (Choose Date)
 - Imagem (Image)
 - Legenda (Legend)
 - EscolheLista (Choose List)
 - VisualizadorDeListas (List Viewer)
 - Notificador (Notifier)
 - CaixaDeSenha (Password Box)
 - Deslizador (Slider)
 - ListaSuspensa (Dropdown List)
 - CaixaDeTexto (Text Box)
 - EscolheHora (Choose Time)
 - NavegadorWeb (Web Navigator)
- Organização (Organization)**
- Mídia (Media)**
- Desenho e Animação (Drawing and Animation)**
- Sensores (Sensors)**

Visualizador (Visualizer):

- Mostrar os componentes ocultos no Visualizador (Show hidden components in the Visualizer)
- Marque para Previsualização no tamanho de um Tablet (Mark for preview in tablet size)
- Screen1
 - Tempo: 10 (Time: 10)
 - Vidas: 3 (Lives: 3)
 - Visual area containing a fly icon and a small colorful icon.
 - Botão Iniciar (Start button) with a dropdown menu 'adicionar ite' (add item).
- Componentes invisíveis (Hidden components): Temporizador1, Som1, Temporizador2

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.

The screenshot displays the Android Studio interface with three main panels: Visualizador, Componentes, and Propriedades.

- Visualizador:** Shows a mobile app preview. At the top, it displays "Tempo: 10" and "Vidas: 3". Below this is a large white area containing a mosquito icon and a small square icon. At the bottom, there is a button labeled "Iniciar Jogo" and a dropdown menu labeled "adicionar ite". A red arrow points to the "Iniciar Jogo" button.
- Componentes:** Lists the components of the app. "OrganizaçãoHorizontal1" is selected, and a red arrow points to it.
- Propriedades:** Shows the properties for "OrganizaçãoHorizontal1". The "Altura" (Height) property is set to "Automático" (Automatic), which is highlighted by a red arrow. Other properties include "AlinhamentoHorizontal" (Left), "AlinhamentoVertical" (Center), "CorDeFundo" (Default), and "Visível" (checked).

ADICIONAR DIFICULDADE

Agora devemos configurar o componente **ListaSuspensa1** para apresentar os valores “Facil, Medio e Dificil”. 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”.

The screenshot displays a software development interface with two main panels: 'Componentes' (Components) and 'Propriedades' (Properties). In the 'Componentes' panel, a tree view shows a hierarchy starting with 'Screen1', which contains several sub-components. 'ListaSuspensa1' is highlighted in green. In the 'Propriedades' panel, the selected component 'ListaSuspensa1' is shown with its configuration options. The 'CadeiaDeElementos' (Chain of Elements) property is set to 'Facil, Medio, Dificil'. The 'Seleção' (Selection) property is set to 'Facil'. Two red arrows point from the 'ListaSuspensa1' component in the tree to the 'CadeiaDeElementos' and 'Seleção' property fields in the properties panel.

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 **nivelSelecioneado** com o valor inicial em “1”.

Visualizador

```
inicializar global nome para
obter
ajustar para
inicializar local nome para dentro de
quando Temporizador1 .Disparo fazer chamar MoverMosquito
inicializar global vidaRestante para 3
inicializar global tempoRestante para 10
inicializar global nivelSelecioneado para 1
```

Iremos definir que o nivelSelecioneado 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 nivelSelecionado. Este bloco possui uma variável local chamada de seleção que contém o valor escolhido (Fácil, Médio ou Difícil).

The image shows a Scratch script editor with two script areas. The left area contains a sequence of blocks for handling a dropdown menu:

- quando ListaSuspensa1 . DepoisDeSelecionar seleção
- fazer
- chamar ListaSuspensa1 . MostrarListaSuspensa
- ListaSuspensa1 . Elementos
- ajustar ListaSuspensa1 . Elementos para
- ajustar ListaSuspensa1 . CadeiaDeElementos para
- ListaSuspensa1 . Altura
- ajustar ListaSuspensa1 . Altura para
- ajustar ListaSuspensa1 . PercentualDeAltura para

The right area contains a timer and a difficulty selection block:

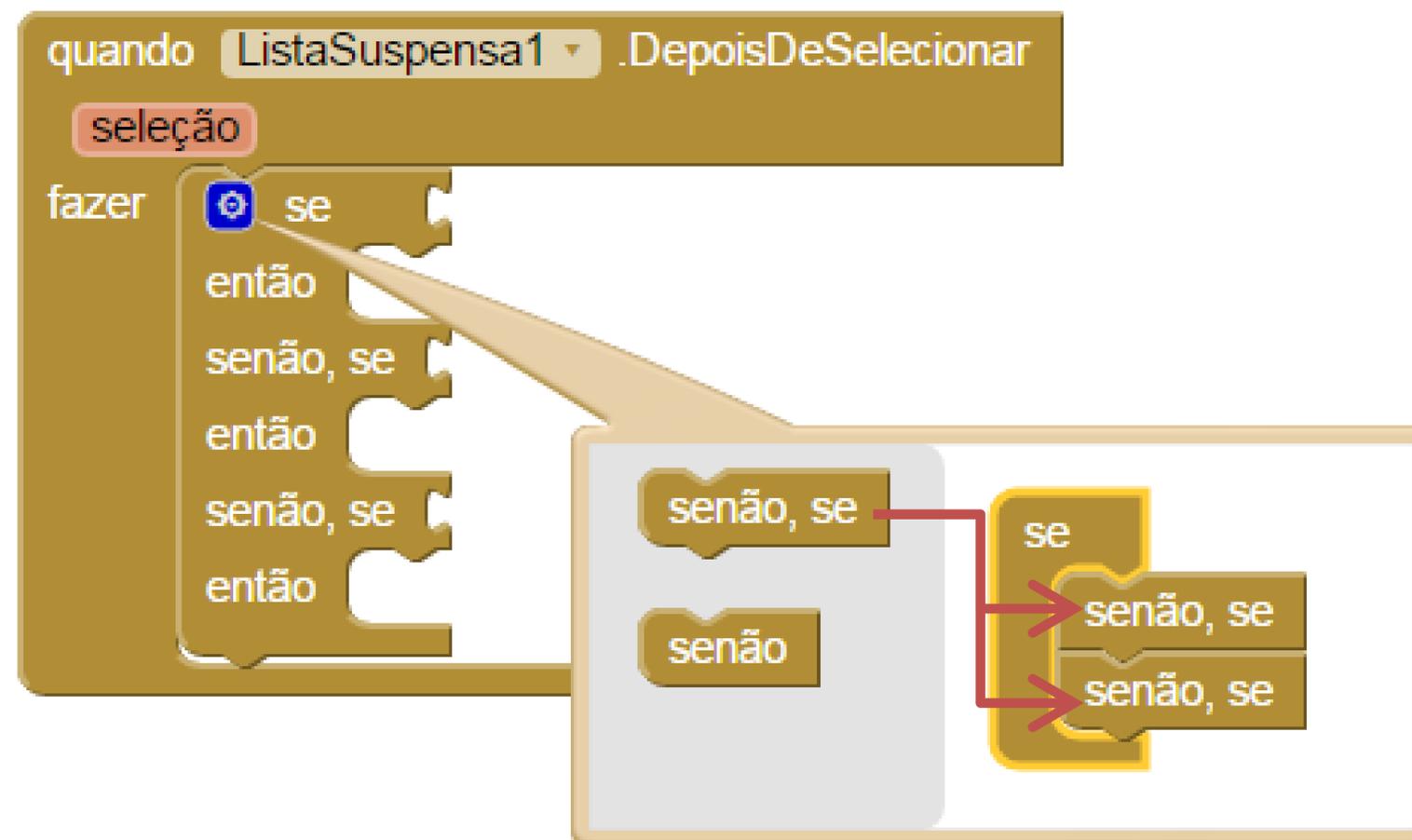
- quando Temporizador2 . Disparo
- fazer
- ajustar global tempoRestante para
- chamar AtualizarTempoRestante
- se obter global tempoRestante
- então chamar FinalizarJogo resultadoFinal falso
- para AtualizarTempoRestante
- fazer
- ajustar Legenda2 . Texto para
- quando ListaSuspensa1 . DepoisDeSelecionar seleção
- fazer

A red arrow points from the 'se' block in the right script to the 'seleção' block in the left script, indicating a data flow or dependency.

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 nivelSelecionado = 1

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

Senão Se (seleção = "Difícil")
Então nivelSelecionado = 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".

The screenshot shows the Scratch programming environment. On the left, the 'Blocos' (Blocks) palette is visible, with the 'Texto' (Text) category selected. The 'Visualizador' (Viewer) pane on the right shows a script for a difficulty level check. The script starts with a 'para AtualizarTempoRestante' block, followed by a 'fazer' block containing a 'juntar' block and an 'obter global tempoRestante' block. Below this is a 'quando ListaSuspensa1 .DepoisDeSelecionar' block. Inside the 'quando' block, there is a 'se' block. The 'se' block has a 'comparar textos' block as its condition, which is set to compare 'obter seleção' with the text 'Facil'. A red arrow points to the '=' operator in the 'comparar textos' block. A black arrow points from the 'comparar textos' block in the 'Blocos' palette to the 'comparar textos' block in the script. Another black arrow points from the 'comparar textos' block in the script to the 'obter seleção' block. A third black arrow points from the 'comparar textos' block in the script to the 'juntar' block in the 'para' block above.

ADICIONAR DIFICULDADE

O procedimento inteiro para seleção do nível de dificuldade na [ListaSuspensa1](#) deve ficar assim:

```
quando ListaSuspensa1 .DepoisDeSelecionar
  seleção
  fazer
    se
      comparar textos obter seleção = "Facil"
      então ajustar global nivelSelecioneado para 1
      senão, se
        comparar textos obter seleção = "Medio"
        então ajustar global nivelSelecioneado para 2
        senão, se
          comparar textos obter seleção = "Dificil"
          então ajustar global nivelSelecioneado para 3
```

ADICIONAR DIFICULDADE

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

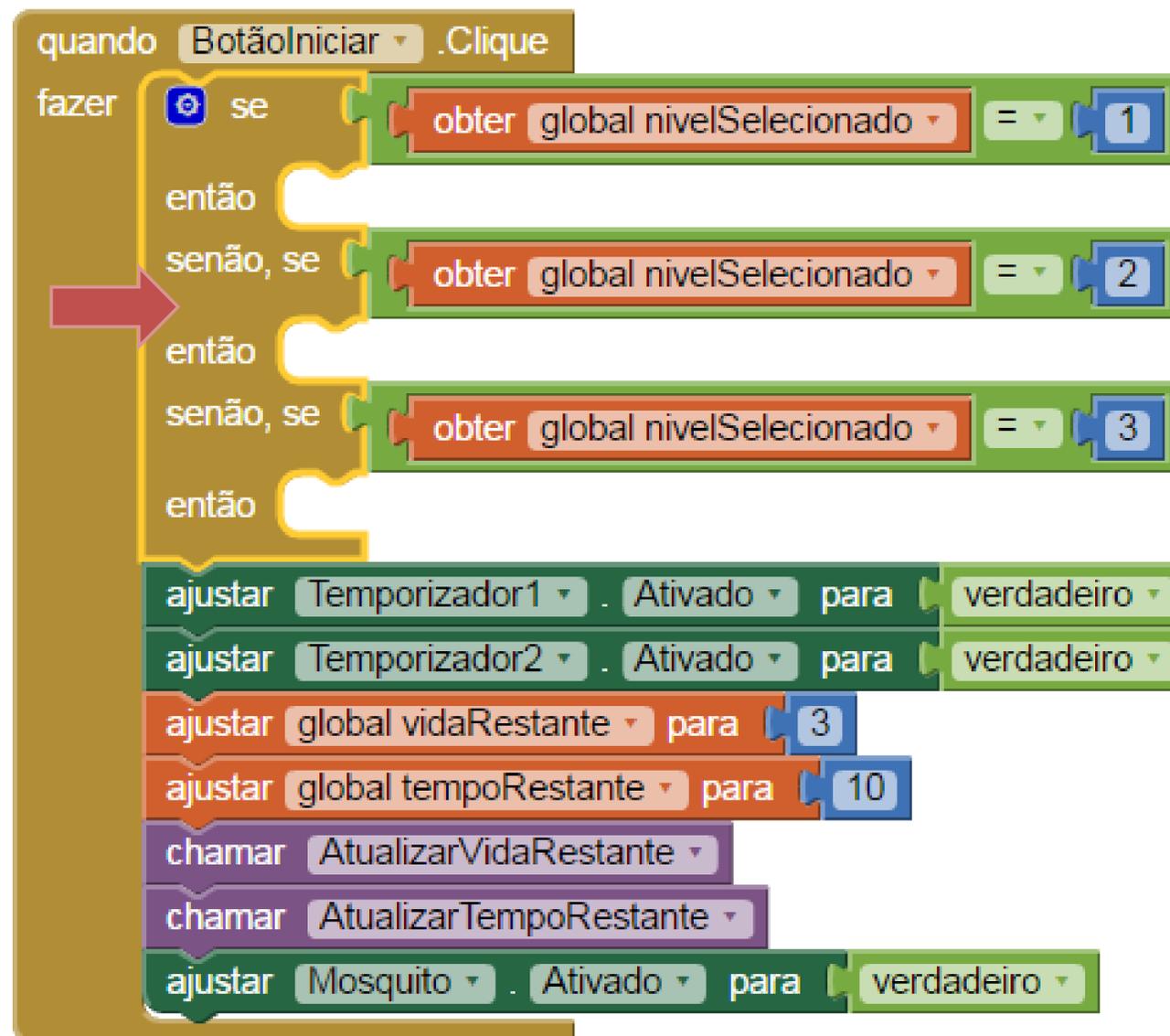
DICA

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 nivelSelecioneado.



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.

```
quando BotãoIniciar .Clique
fazer
  se
    obter global nivelSelecionado = 1
  então
    ajustar global tempoRestante para 20
    ajustar Temporizador1 . Intervalo para 500
  senão, se
    obter global nivelSelecionado = 2
  então
    ajustar global tempoRestante para 10
    ajustar Temporizador1 . Intervalo para 400
  senão, se
    obter global nivelSelecionado = 3
  então
    ajustar global tempoRestante para 8
    ajustar Temporizador1 . Intervalo para 350
  ajustar Temporizador1 . Ativado para verdadeiro
  ajustar Temporizador2 . Ativado para verdadeiro
  ajustar global vidaRestante para 3
  ajustar global tempoRestante para 10
  chamar AtualizarVidaRestante
  chamar AtualizarTempoRestante
  ajustar Mosquito . Ativado para verdadeiro
```

ADICIONAR DIFICULDADE

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

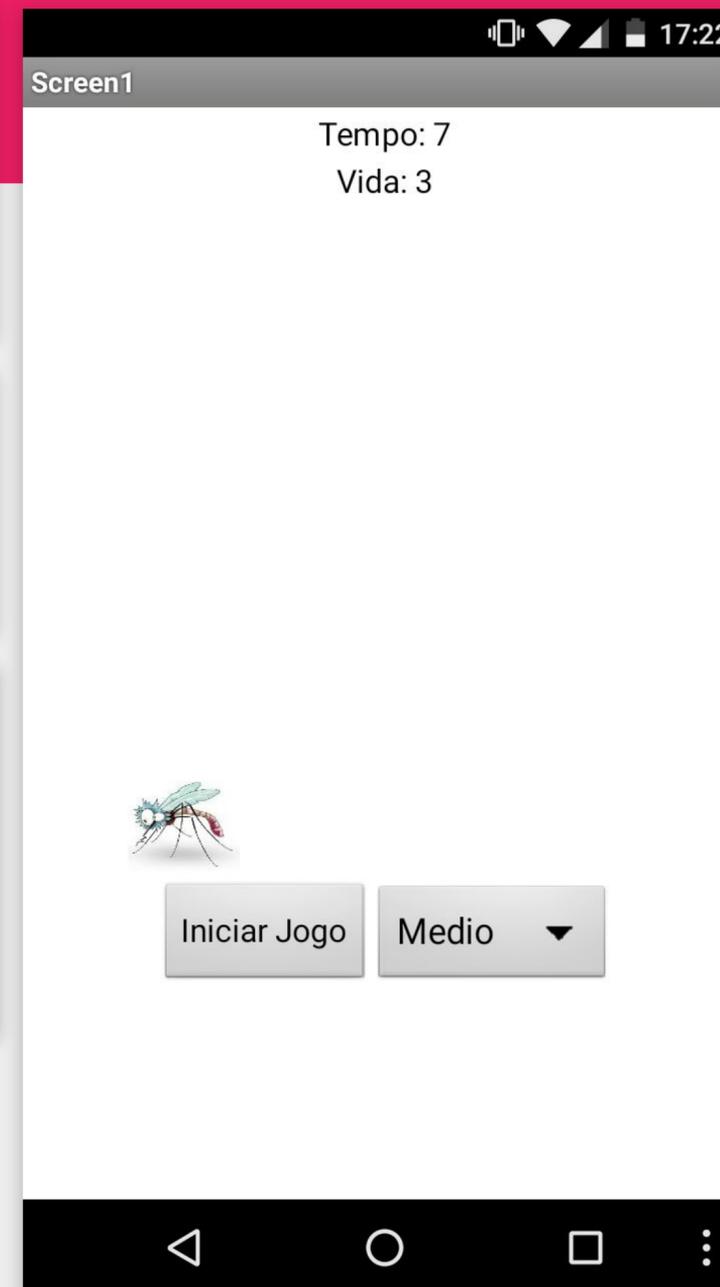
```
quando BotãoIniciar .Clique
fazer
  se
    obter global nivelSelecionado = 1
  então
    ajustar global tempoRestante para 20
    ajustar Temporizador1 . Intervalo para 500
  senão, se
    obter global nivelSelecionado = 2
  então
    ajustar global tempoRestante para 10
    ajustar Temporizador1 . Intervalo para 400
  senão, se
    obter global nivelSelecionado = 3
  então
    ajustar global tempoRestante para 8
    ajustar Temporizador1 . Intervalo para 350
  ajustar Temporizador1 . Ativado para verdadeiro
  ajustar Temporizador2 . Ativado para verdadeiro
  ajustar global vidaRestante para 3
  chamar AtualizarVidaRestante
  chamar AtualizarTempoRestante
  ajustar Mosquito . Ativado para verdadeiro
```

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
VOCÊ!

SEJA CRIATIVO



COMPUTAÇÃO NA ESCOLA

