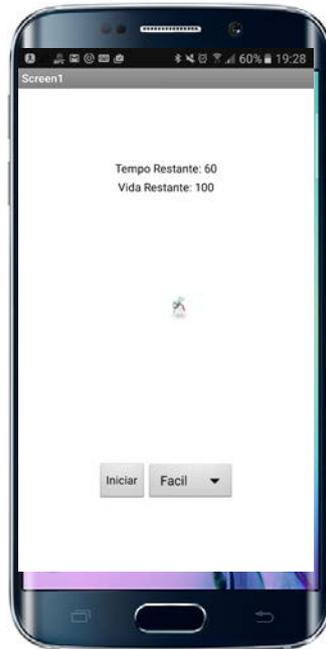


## Roteiro Oficina – Jogo Caça Mosquito



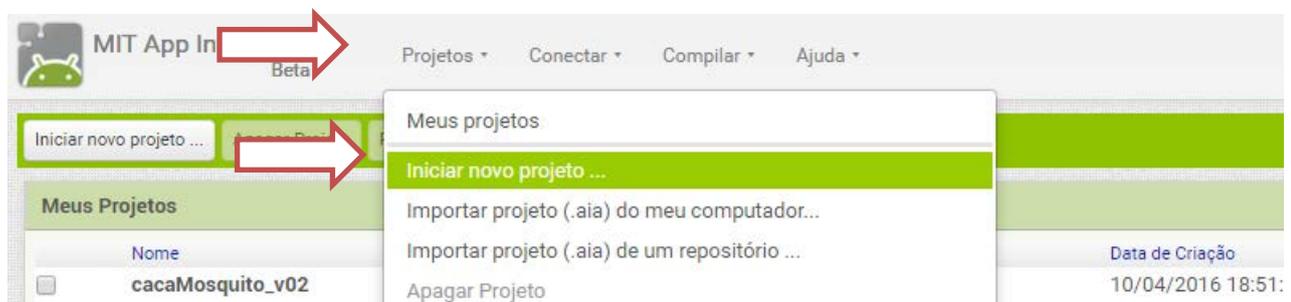
O jogo Caça Mosquito foi desenvolvido com a ideia de colocar a imagem de um mosquito voando pela tela para que o jogador tente mata-lo.

Após clicar no botão iniciar, será ativado um cronômetro dando um determinado tempo para o jogador apertar o mosquito 10 vezes. Caso o jogador consiga, ele se sai vitorioso. Caso o tempo se esgote, ele é derrotado.

Existe a possibilidade de o jogador escolher o nível de dificuldade em que ele pretende jogar. Cada nível de dificuldade maior aumenta a velocidade em que o mosquito se movimenta e diminui o tempo disponível.

### 1. Criando o Projeto

- 1) Vamos iniciar criando um novo projeto no AppInventor. Para isto, vamos clicar em Projetos e depois em Iniciar novo projeto...



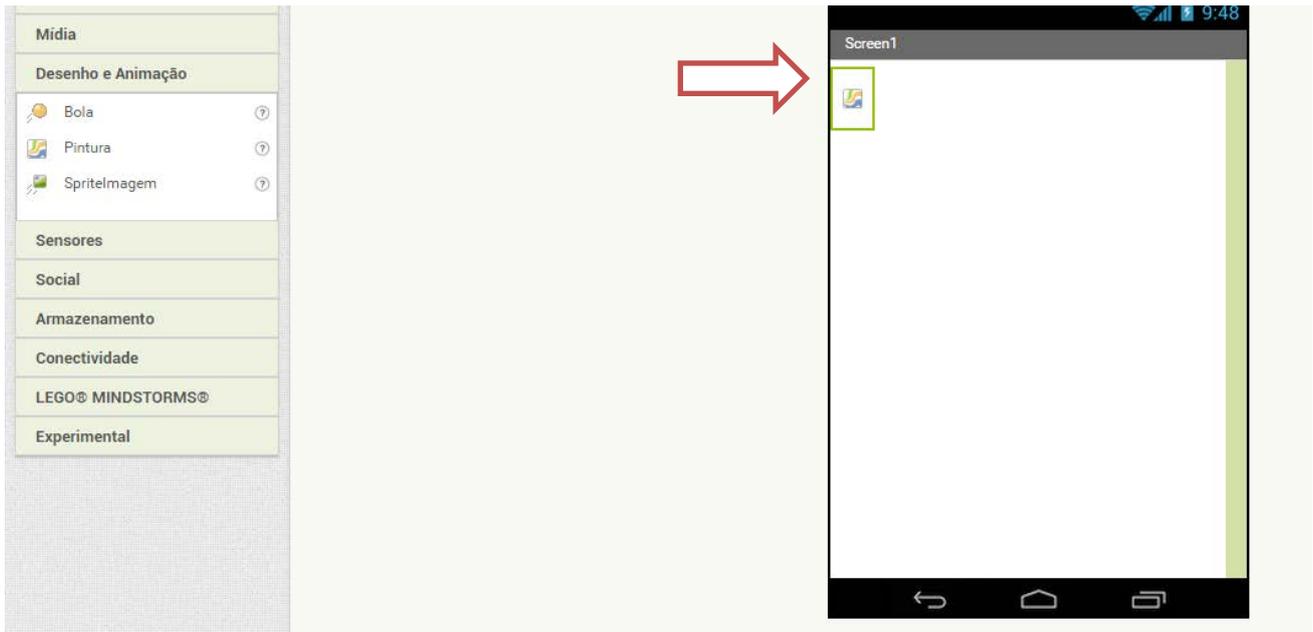
Vamos definir o nome do novo projeto como “CacaMosquito” e clicar em OK.



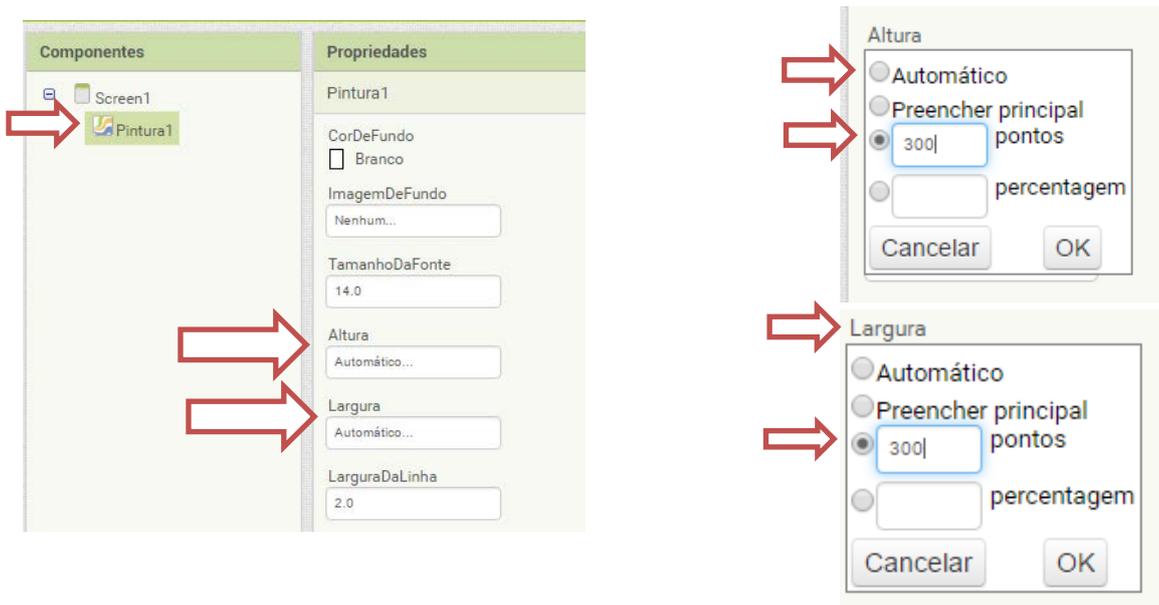
- 2) Vamos começar definindo a nossa interface. Primeiro vamos clicar na paleta “Desenho e Animação” e arrastar o componente “Pintura” para dentro da nossa tela.



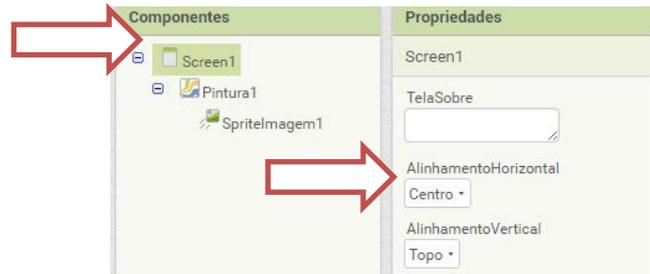
Você deve ver uma pintura na tela como na imagem a seguir:



Vamos alterar as propriedades da pintura para que ela possua altura de 300 pontos e largura de 300 pontos. Para isto, clique e selecione o componente “Pintura1” e altere o valor da sua altura e largura.

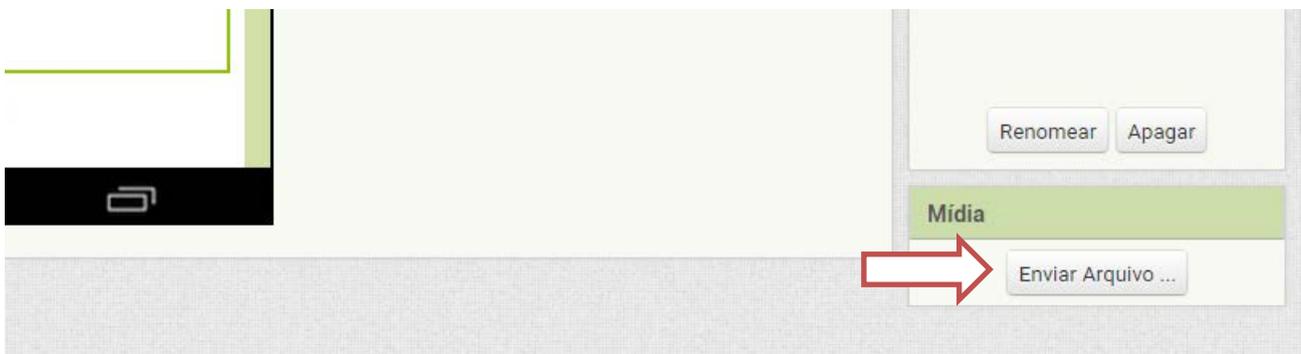


Para alinhar a pintura no centro da tela, vamos selecionar o componente “Screen1” e na propriedade “AlinhamentoHorizontal” vamos selecionar “Centro”.



3) Inserindo a figura do mosquito e movimentando pela “Pintura”.

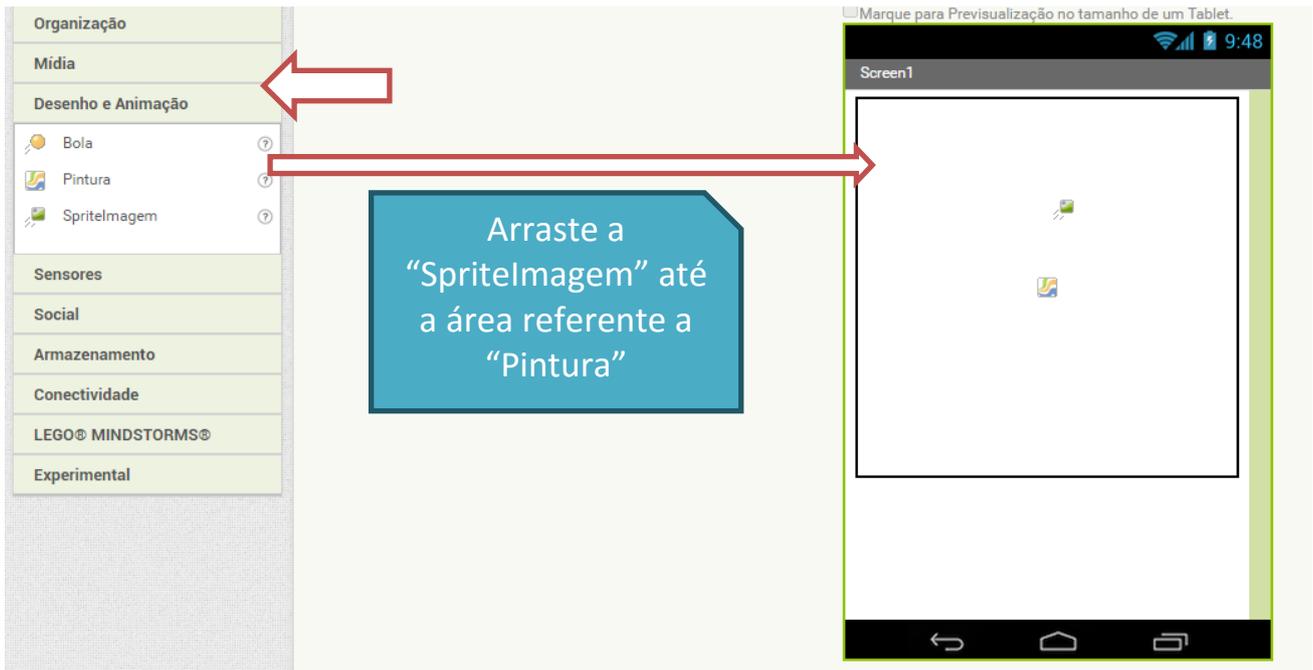
Para inserir a figura do mosquito no projeto devemos clicar no botão “Enviar arquivo...” em Mídia.



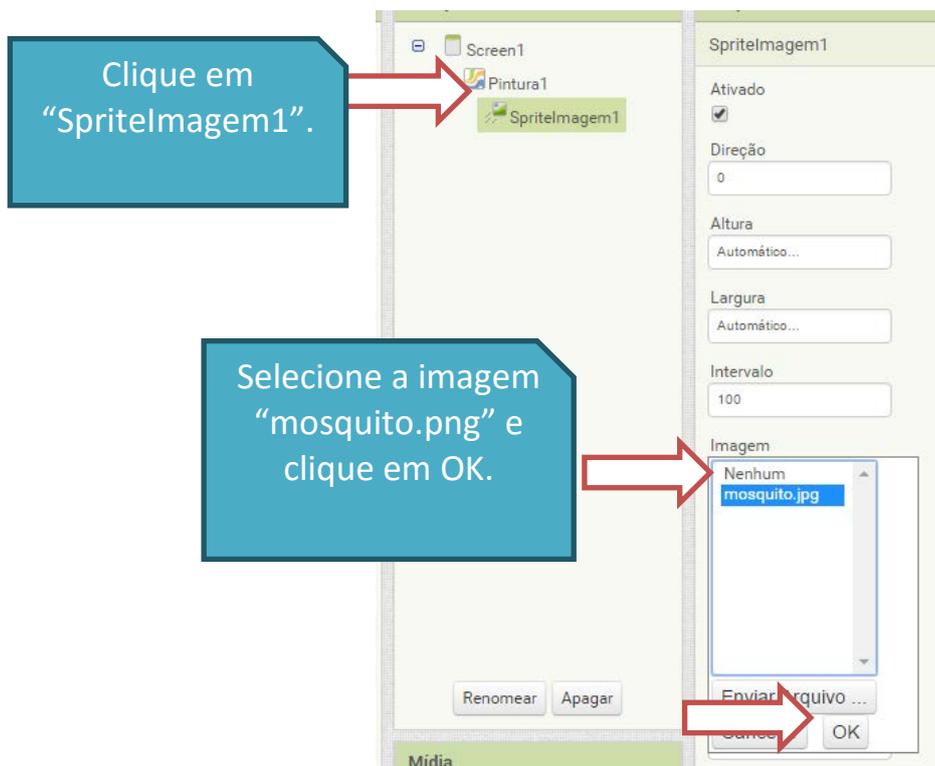
Clicar em escolher arquivo e seleccionar o arquivo com nome “mosquito.png” o qual encontra-se na pasta que foi descompactada no passo 2 em “Imagens”. Clicar em OK.

Agora iremos inserir o componente “SpriteImagem” na área referente ao componente “Pintura”. Clicando em “Desenho e Animação” e arrastando o “SpriteImagem” até a “Pintura”.

Clique em  
“Desenho e  
Animação”.

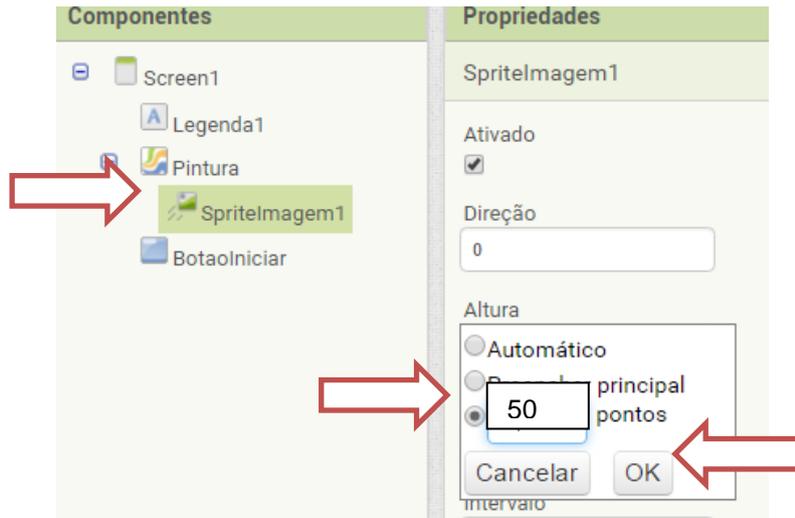


Agora devemos definir a imagem “mosquito.png” para o componente “SpriteImagem1”. Para fazer isso é necessário selecionar o componente “SpriteImagem1” e ao clicar no campo “Imagem” selecionar o arquivo “mosquito.png”.



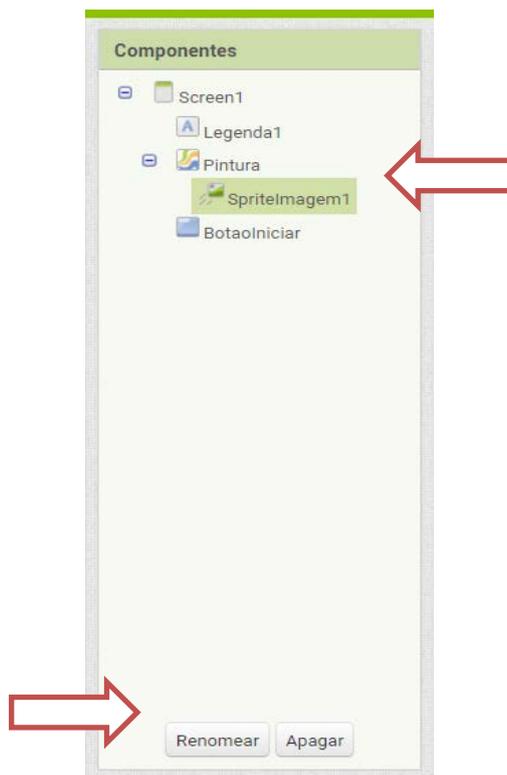
Ao colocar a imagem do mosquito no componente “SpriteImagem1” devemos redimensionar o componente para que ele fique do tamanho desejado.

Sendo assim, selecione o componente “SpriteImagem1” como feito anteriormente e clique no campo “Altura” definindo o valor “Pontos” para “50” e clique em “OK”.



Realize o mesmo procedimento para o campo “Largura” definindo o mesmo valor (50).

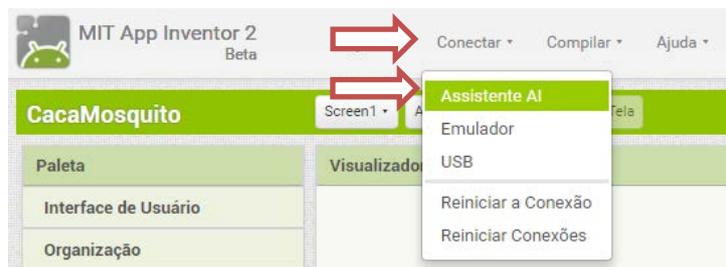
Devemos renomear o componente “SpriteImagem1” para “mosquito”. Selecionando o componente “SpriteImagem1” e clicando no botão Renomear e definindo o valor **Novo Nome** para “mosquito”.





Agora vamos verificar o que acontece! Pegue o seu celular e abra o aplicativo “MIT AI2 Companion”.

No App Inventor clique em “Conectar” -> “Assistente AI”.

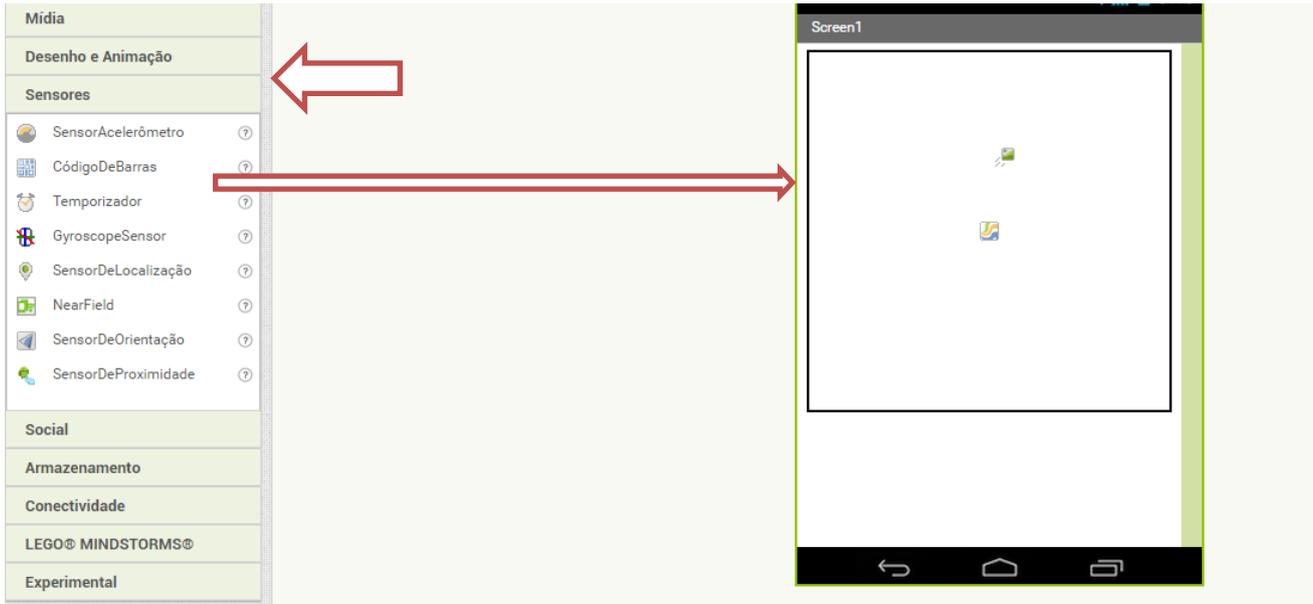


No aplicativo do celular clique em “scan QR code” e aponte a câmera para o símbolo da tela do computador. Pronto, o seu celular está conectado com o código que temos no computador.

Por enquanto só temos um mosquito parado na tela.

### 3.1) Mover o Mosquito pela tela.

Para que o mosquito se movimente pela tela iremos utilizar o componente **Temporizador**, este será responsável para mandar o mosquito se movimentar a cada 0,5s. Devemos clicar em **Sensores** e arrastar o componente **Temporizador** até a tela.



O componente **Temporizador** é um componente invisível na tela, ele fica apresentado na parte inferior do bloco “Designer” junto com outros componente que também não são visíveis na tela.



Como o componente “Temporizador1” deve mandar o mosquito se mover a cada 0,5s. O campo **intervalo** é definido em milissegundos, deste modo devemos definir o seu valor para 500.

Selecione o “Temporizador1” e defina o valor **intervalo** para “500”.

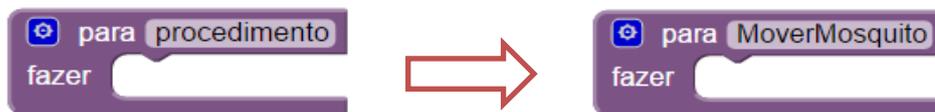
Selecione o campo **Disparos Contínuos**. Este campos serve para que o “Temporizador1” realize disparos a cada **intervalo** de tempo continuamente quando ele for ativado. Mantenha o campo **Ativo** ativado.



Devemos criar um **Procedimento** responsável por realizar o movimento do componente “mosquito” pela tela. Então vamos selecionar **Procedimentos** e adicionar o bloco de instruções **Para Procedimento Fazer** arrastando para a área do “Visualizador”.



Vamos renomear este procedimento para “MoverMosquito”. Clique sobre o campo “procedimento” e digite “MoverMosquito” e tecle enter.



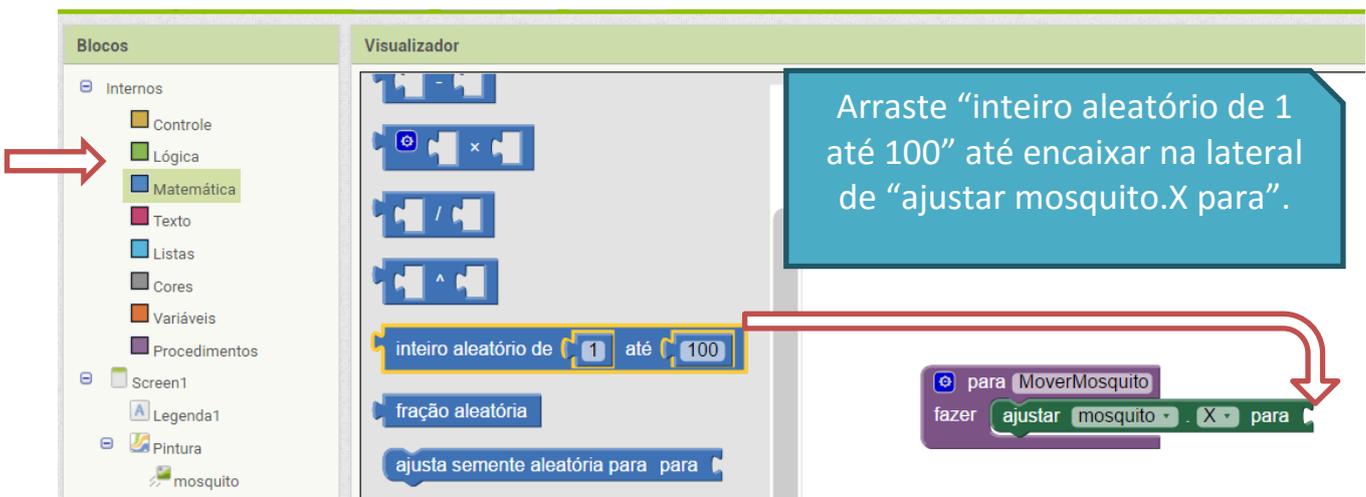
É necessário que o procedimento “MoverMosquito” toda vez que for chamado, deve trocar a posição em que o componente “mosquito” está sendo apresentado na tela.

O procedimento “MoverMosquito” deve ser responsável por trocar a posição do “mosquito” toda vez que for chamado. Para fazer isto, vamos utilizar a função de trocar a posição de **X** e **Y** do componente **SpriteImagem**, ou seja, o “mosquito”.

No procedimento “MoverMosquito” vamos definir **X** como sendo um valor aleatório entre 0 e o ponto máximo que o “mosquito” pode ir na “Pintura”. Para fazer isto devemos clicar em “mosquito” e arrastar o bloco “ajustar mosquito.X para”, que fica no final da lista de comandos, para dentro do procedimento “MoverMosquito”.



Agora devemos incluir o valor de **X** em que o mosquito deve ser posicionado. Vamos utilizar a operação “inteiro aleatório de 1 até 100” do bloco de matemática. Para isto, deve-se clicar em matemática e arrastar o bloco referente a operação para encaixar na operação de “ajudar mosquito.X para”.



Desse modo, a posição do mosquito irá variar em X de 1 até 100. Entretanto para que ele possa voar por toda a extensão da “Pintura”, ele deve ir de 0 até o ponto máximo da largura e em que o mosquito seja totalmente visível na pintura, ou seja, 270 (Largura da “Pintura” – Largura do “Mosquito”).

Para isto, devemos trocar o primeiro valor de “1” para “0” e o valor “100” para “270”.



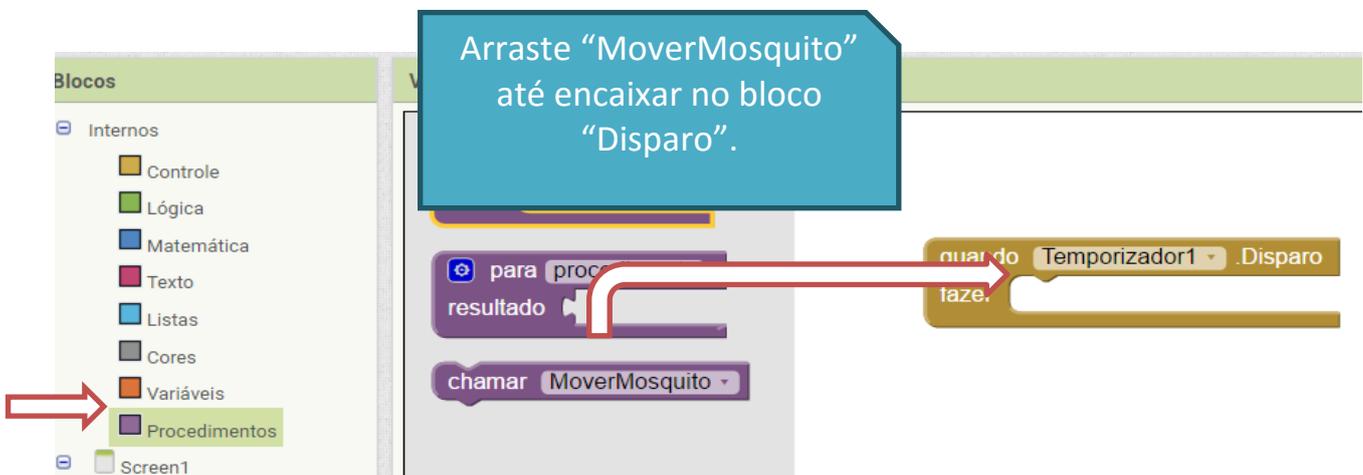
Após definir como o “mosquito” deve se mover, ele deve saber quando se mover, e isto é, quando o “Temporizador1” disparar.

Devemos selecionar o componente “Temporizador1” e arrastar o bloco de instruções relacionado ao evento de Disparo para a área do “Visualizador”.



Agora, a cada 0,5s será executado as instruções que estão no bloco referente ao disparo do “Temporizador1”. Entretanto, ainda não possui nenhuma instrução para ser realizada.

Devemos chamar o procedimento “MoverMosquito” toda vez que o nosso “Temporizador1” executar o disparo. Então, deve-se clicar em **procedimentos** e arrastar o bloco **Chamar (MoverMosquito)** para dentro do bloco Temporizador1.Disparo.



Teste o seu aplicativo!

```
quando Temporizador1 .Disparo
fazer chamar MoverMosquito
```

Você verificou que o seu “mosquito” : necessário também alterar o seu valor na coordenada **Y**! Faça o mesmo procedimento anterior para alterar o valor mosquito.Y.

O resultado do bloco “MoverMosquito” é:

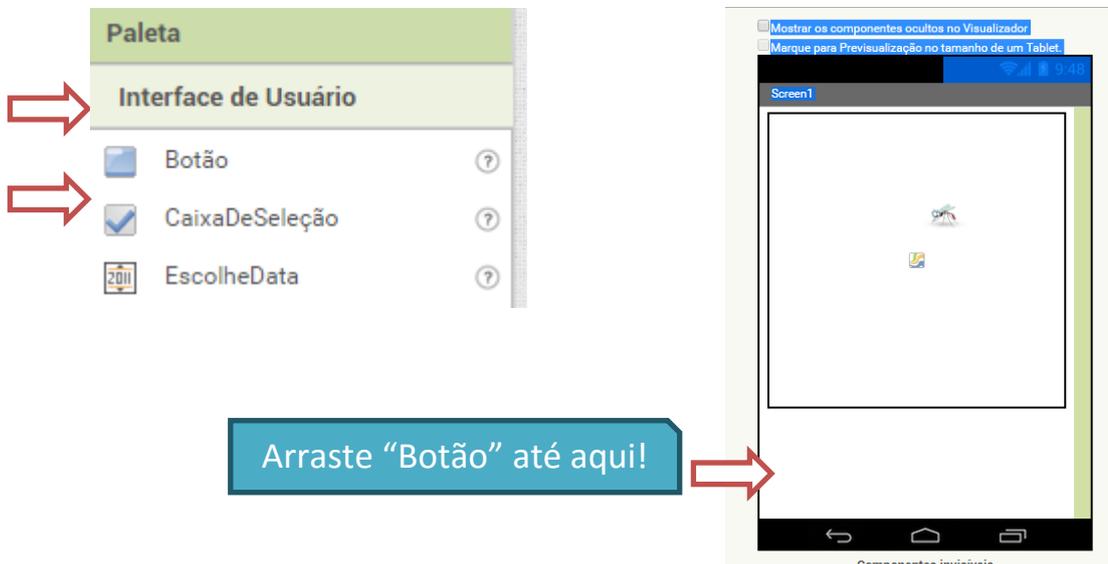
```
para MoverMosquito
ajustar mosquito . X para inteiro aleatório de 0 até 270
ajustar mosquito . Y para inteiro aleatório de 0 até 270
```

Teste o jogo!

4) Iniciar o Jogo

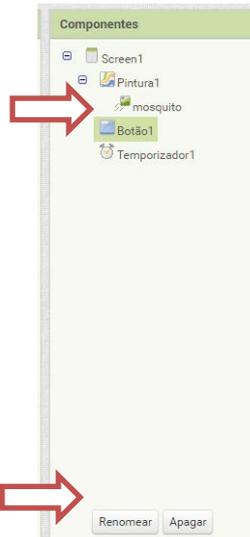
O mosquito está se movimentando a cada disparo do “Temporizador1”. Contudo precisamos que isso seja realizado somente quando formos jogar. Para isto, vamos inserir um botão “BotaIniciar”.

Vá em “Designer”, na paleta clique em “Interface de Usuário” e arraste o componente “Botão” até a tela, preferencialmente embaixo da área da “Pintura1”.



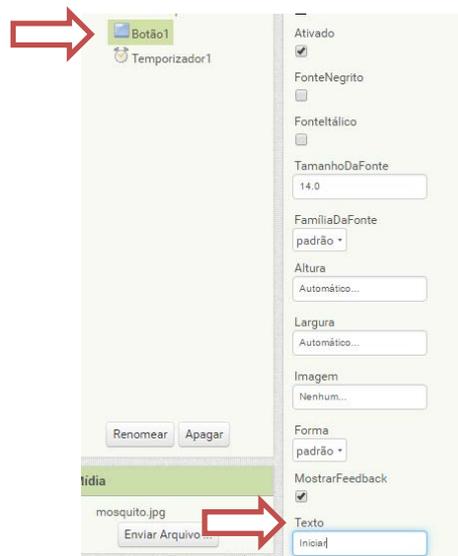
Agora, da mesma maneira que renomeamos o “mosquito”, vamos renomear o componente “Botao1” para “BotaIniciar”.

Clique em “Botao1” e em  
“Renomear”.  
Altere para “Botaoiniciar”.



Devemos também trocar a configuração do “Botaoiniciar” para ele apresentar o texto “Iniciar”.

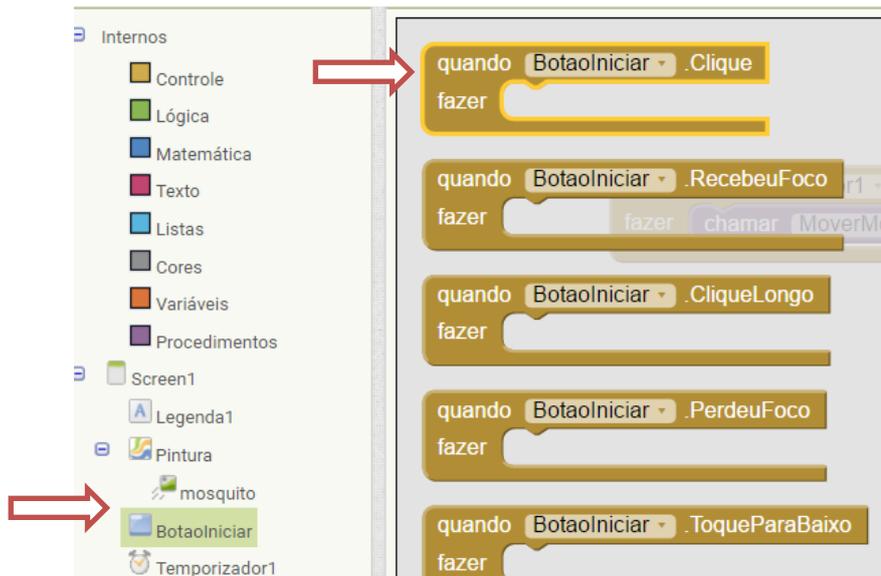
Para isto, selecione o componente “Botaoiniciar” e no campo “Texto” digite “Iniciar”.



Veja no seu aplicativo se o botão está aparecendo corretamente!

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

Para isto, vamos adicionar o bloco “quando Botaoiniciar.clique”. Este bloco se encontra nos blocos referentes ao “BotãoIniciar”.

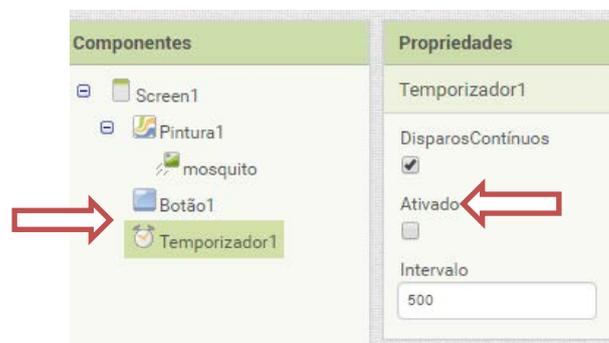


Agora sempre que for realizado um clique no botão “BotãoIniciar” ele vai executar as funções que estiverem dentro do bloco “quando BotãoIniciar.Clique”.

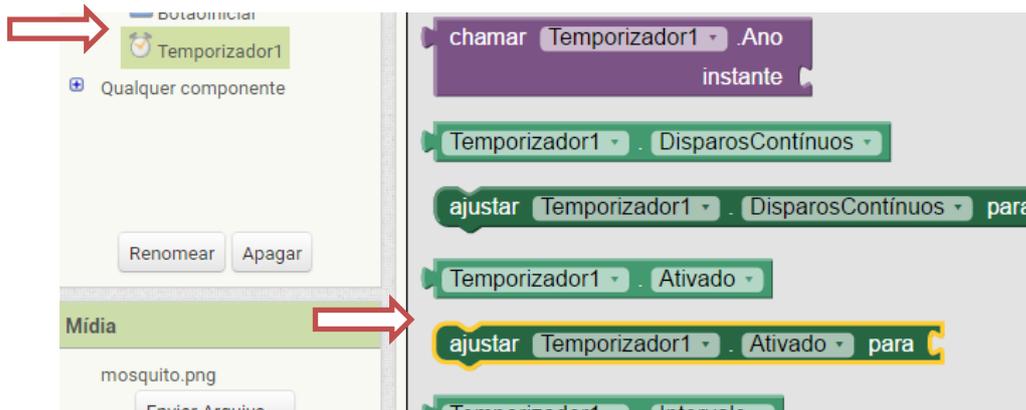
Veja se está tudo correto com o seu aplicativo. Ele está fazendo o mosquito se movimentar somente ao apertar o botão “BotãoIniciar”?

Pois é, não está correto. Precisamos alterar a configuração do “Temporizador1”, o qual controla o movimento do mosquito para ele ser ativado somente ao clicarmos no botão iniciar!

Para isto, vamos ao “Designer”, selecione o “Temporizador1” e desmarque a opção “Ativado”



Agora precisamos ativar o “Temporizador1” quando o “BotãoIniciar” for clicado. Ou seja, vamos adicionar o bloco “ajustar Temporizador1.Ativado para” e colocar o valor lógico “verdadeiro”.



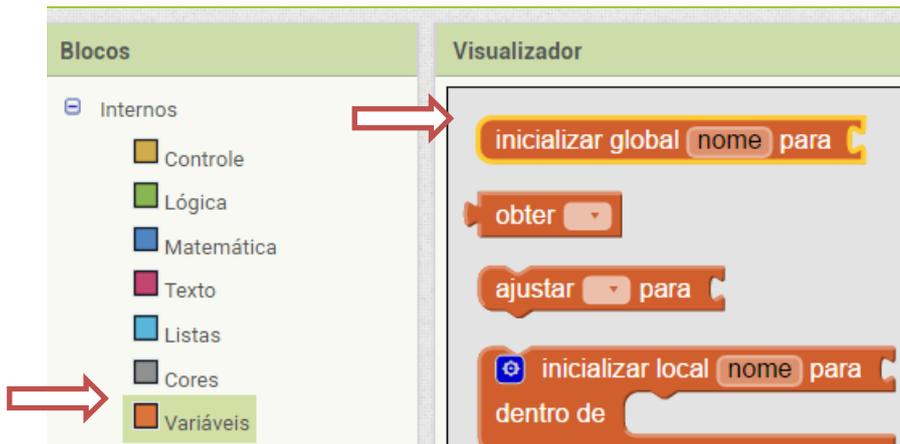
Agora teste o seu aplicativo e verifique que o mosquito se mova após apertar no botão iniciar!

A partir de agora seria desenvolvido as funcionalidades mais específicas que aparecem no screenshot na capa do roteiro. Adicionar “Vida” ao mosquito e a cada clique nele diminuir a sua vida, um “Tempo” para o mosquito ser derrotado e também a seleção de diferentes níveis de dificuldade.

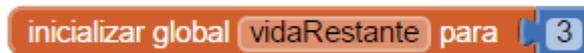
### 1. Adicionando “Vida” ao mosquito.

Nessa funcionalidade vamos iniciar o jogo com o mosquito possuindo o valor “3” de vida, e cada vez que ele for clicado na tela devemos diminuir a sua vida em “1”. Ou seja, quando conseguirmos clicar 3 vezes sobre o mosquito ganhamos o jogo.

Para isto devemos criar um valor que armazene o quanto de vida o mosquito possui. Vamos utilizar uma variável global que será chamada de “vidaRestante”. Usando o bloco “inicializar global nome para” e atribuir o valor “3”.



O bloco de iniciar variável global não deve ser inserido dentro de outro bloco de função. Ele é executado automaticamente toda a vez que o aplicativo for aberto. Sendo assim, basta arrastarmos ele para a área do Visualizador e colocar o valor do bloco matemático “3” junto a ele.



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” e neste bloco colocamos a operação matemática de subtração com o valor de “vidaRestante – 1”.



No bloco mosquito.Tocou vamos alterar o valor da variável “vidaRestante” aplicando a operação “vidaRestante – 1”. Para isto devemos colocar o bloco “ajustar ... para” que fica nos blocos de variável dentro do procedimento “quando mosquito.Tocou”.



Obtendo como resultado o bloco desta maneira:



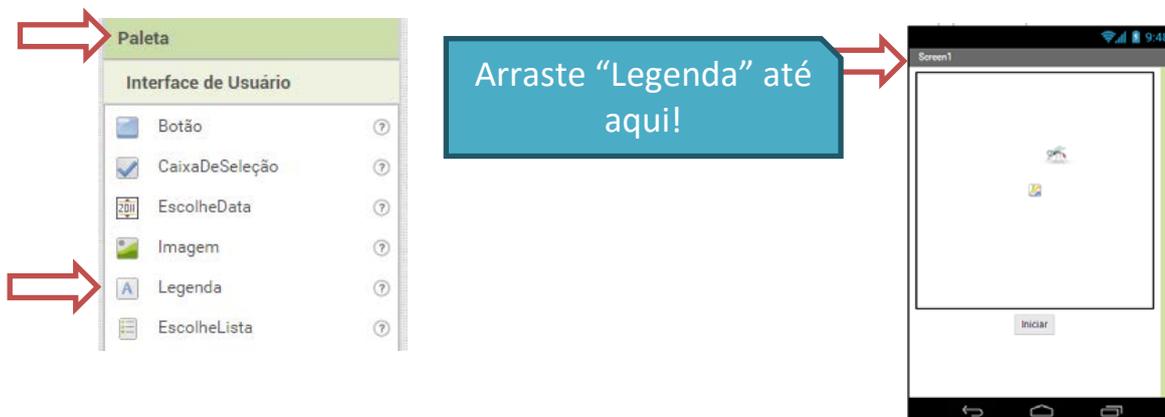
Precisamos clicar na setinha do bloco “ajustar ... para” e definir a variável vidaRestante, e também precisamos definir como o primeiro valor da operação o valor de “vidaRestante”. Para isto devemos incluir o bloco “obter ...” que está dentro dos blocos das variáveis como o primeiro elemento da subtração e selecionar o valor “vidaRestante” clicando na setinha.



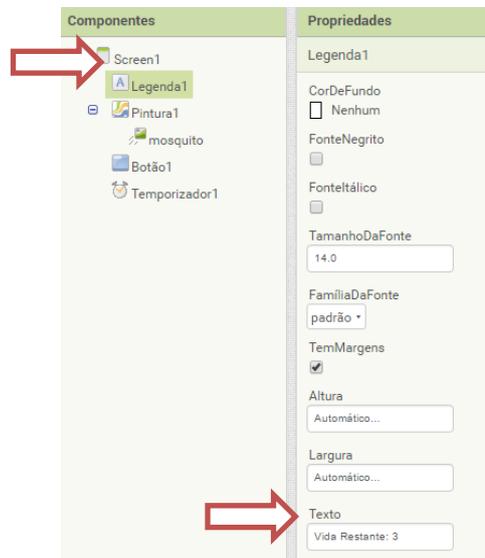
Obtendo como resultado o bloco:



Para o usuário saber se o mosquito foi tocado ou não vamos informá-lo de duas maneiras: Alterando o valor da vidaRestante apresentada na Tela e Vibrando o telefone quando o “mosquito” for clicado. Vamos ao “Designer” inserir um componente “Legenda” a tela. Clique em “Interface de Usuário” e arraste o componente “Legenda” até a tela, preferencialmente acima da “Pintura1”.



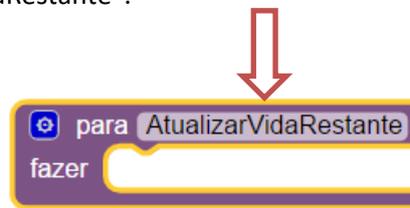
Vamos trocar o texto da “Legenda1” para “Vida Restante: 3”.



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! Quando o mosquito for tocado e após o valor da variável “vidaRestante” for diminuído devemos chamar um procedimento para atualizar o texto contido na “Legenda1”.

Vamos criar o procedimento chamado de “AtualizarVidaRestante” e dentro deste bloco alterar o valor do seu texto baseado no valor que temos na variável “vidaRestante”.



Para alterar o valor do texto escrito no componente “Legenda1” utilizamos o bloco “ajustar Legenda1.Texto para”, utilizar o bloco de manipulação de texto “Juntar” pertencente aos blocos de texto e o bloco de texto “ ” atribuindo os valores “Vida Restante: ” e o valor da variável “vidaRestante”.



Obtendo o bloco:



Não devemos esquecer de quando o mosquito for tocado devemos chamar o procedimento "AtualizarVidaRestante" para atualizar o texto na tela.

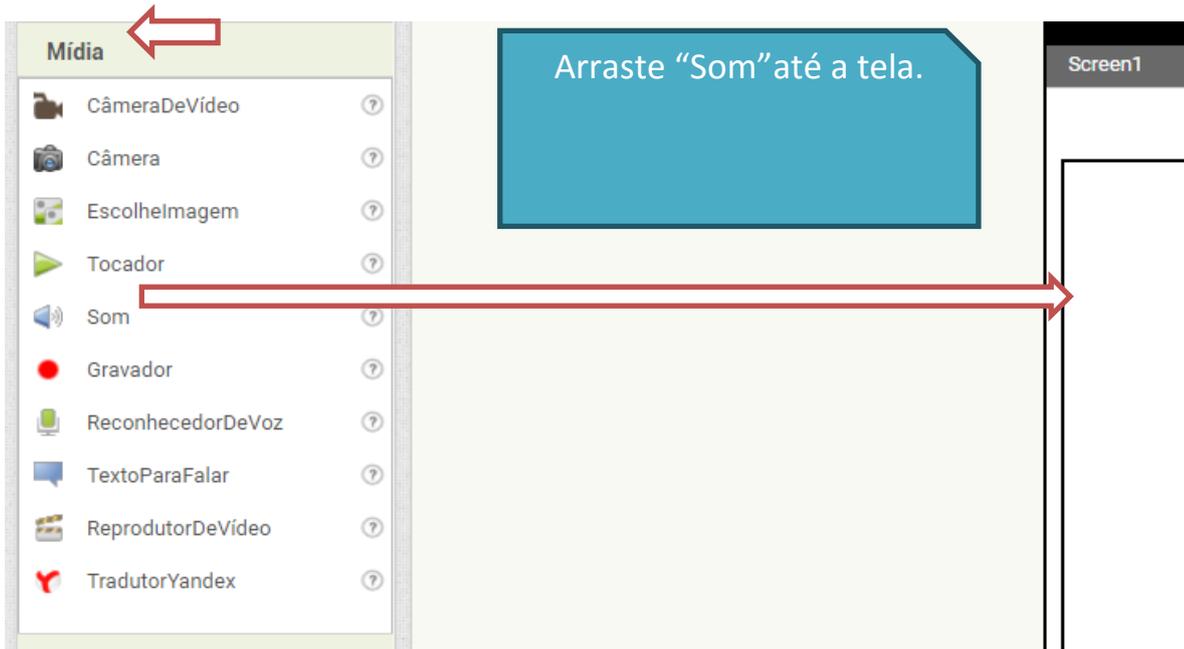


Agora além de mostrar a quantidade de vidas na tela, vamos vibrar o telefone quando o mosquito for tocado, precisamos inserir o componente "Som" no "Designer" para poder usar a sua funcionalidade de vibrar o telefone.

Para isto, vá até o "Designer" selecione a paleta "Mídia" e arraste o componente "Som" até a tela. O componente "Som" é invisível, deste modo, deve estar listado entre os componentes invisíveis.



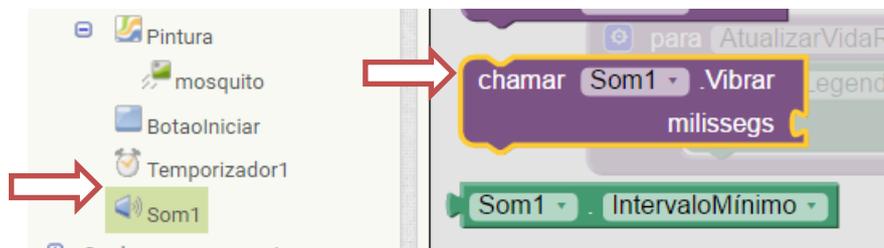
Selecione a paleta "Mídia"



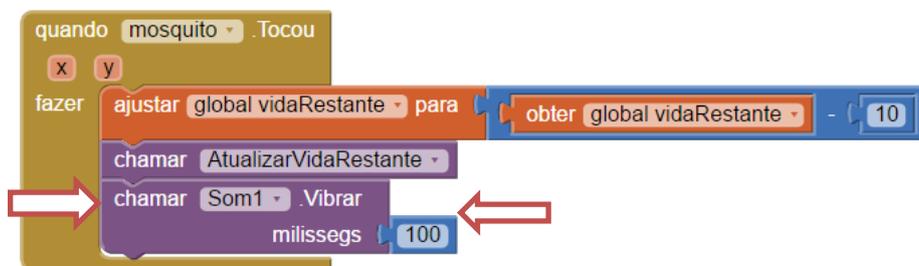
Obtendo “Som1” como um componente invisível.



Agora devemos voltar ao “Blocos” para no procedimento “mosquito.Tocou” chamar o componente “Som1” para vibrar o telefone. O bloco que devemos utilizar é “chamar Som1.Vibrar” e informar o tempo que desejamos que ele vibre, vamos definir este tempo como “100” milissegundos.



Tendo o bloco “mosquito.Tocou” deste modo:



Teste o seu jogo, clique sobre o mosquito e veja se o seu celular está tremendo!

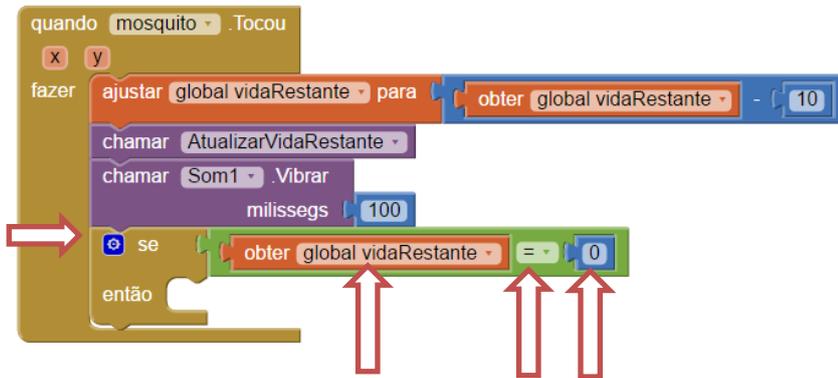
2. Terminar o jogo quando o mosquito não tiver mais “Vida”.

Como devem 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. Para fazer esta verificação vamos usar o bloco de controle “Se, então” e utilizar a operação logica “vidaRestante = 0” para terminar o jogo.

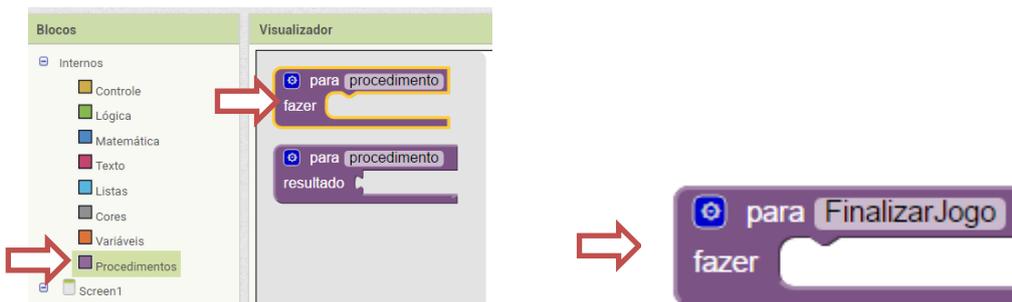


Obtendo o bloco:



Se o valor de “vidaRestante” for maior do que zero o jogo deve continuar sem nenhuma alteração, agora se o valor de “vidaRestante” for igual a zero devemos chamar um procedimento responsável por terminar o jogo.

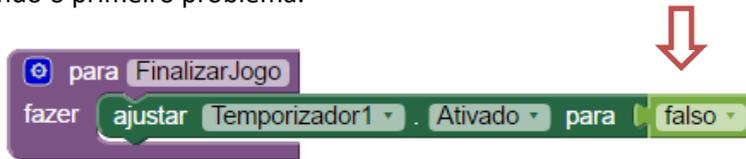
Vamos criar um novo procedimento “FinalizarJogo” através dos blocos procedimento .



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.

Para isto vamos adicionar o bloco “ajustar Temporizador1.Ativado para” informando o valor falso, pois ele deve parar e solucionando o primeiro problema.



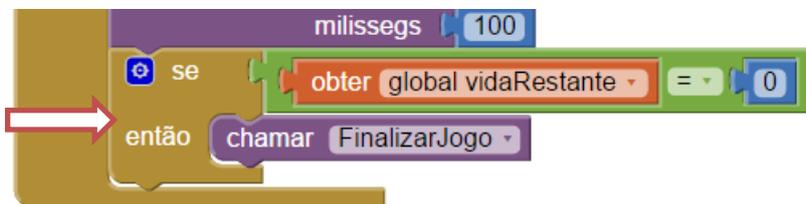
Para solucionar o segundo problema vamos adicionar o bloco “ajustar Legenda1.Texto para” informando o texto “Você eliminou o mosquito!”.



Já para não atualizar o número de vidas do mosquito, temos que adicionar o bloco “ajustar mosquito.Ativado para” informando o valor Falso.



Agora, dentro do bloco de controle “Se, então” devemos chamar o procedimento “FinalizarJogo”.



Vamos testar, veja se o jogo está finalizando!

### 2.1 Inicializar o texto da “Legenda1”

Você verificou que ao finalizar o jogo e recomençar 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!

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 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.

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



Para trocar o texto de acordo com a “vidaRestante” vamos chamar o procedimento “AtualizarVidaRestante” que já realiza a função e para ativar o “mosquito” basta utilizarmos o mesmo bloco que estava em “Botaoiniciar.Clique” e alterá-lo para o mosquito.

Obtendo o bloco:

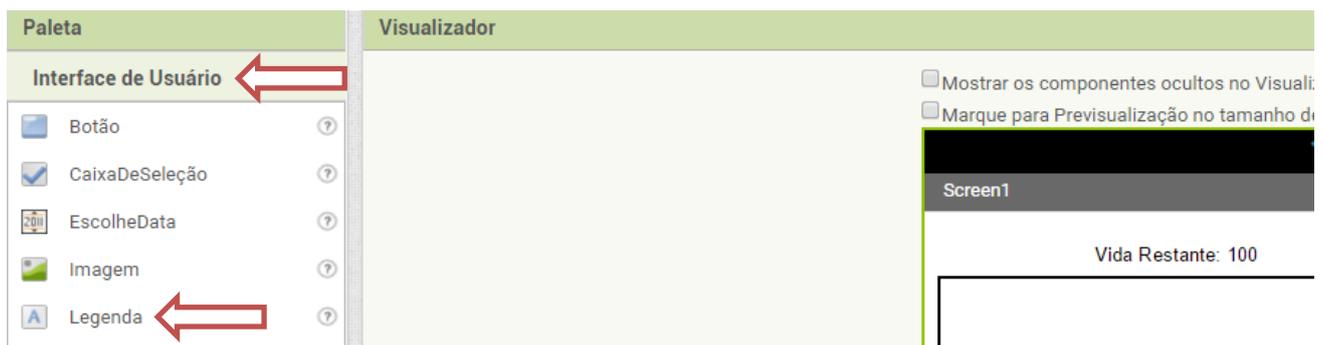


Vamos testar, o jogo deve estar funcionando perfeitamente!

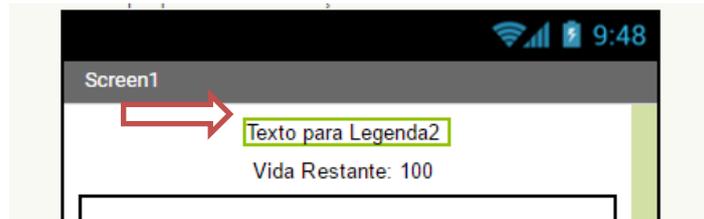
## 2.2 Adicionar “Tempo” para matar o mosquito.

Como foi visto, o jogador ainda não possui nenhuma maneira de sair derrotado. Para isto ocorrer vamos fornecer um tempo para o mosquito ser derrotado, caso ele não seja eliminado no tempo fornecido o jogador será derrotado.

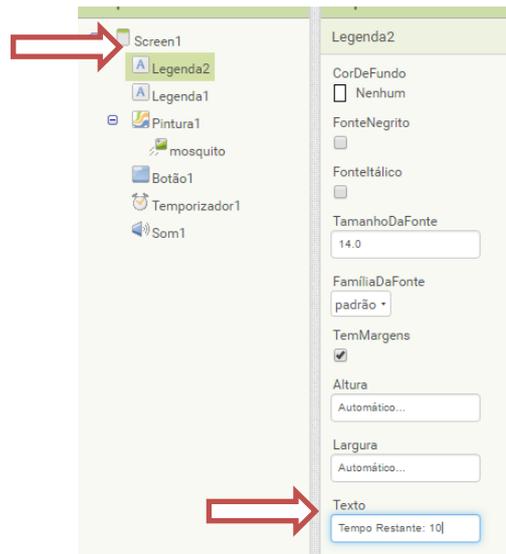
Para fazer isto, vamos ao “Designer” e adicionaremos um novo componente **Legenda** chamada de “Legenda2”, preferencialmente acima da “Legenda1”.



Obtendo o seguinte resultado:

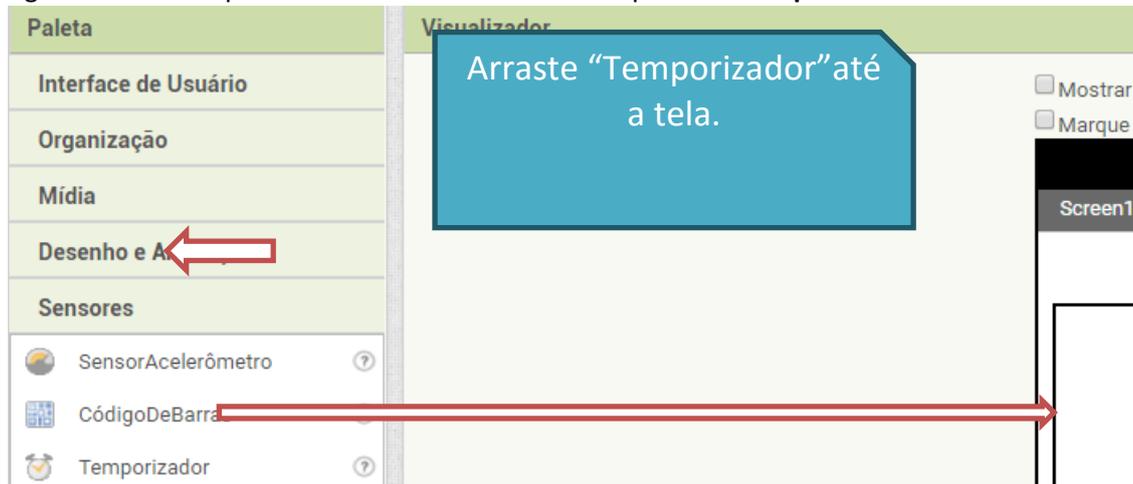


Agora precisamos alterar o texto da “Legenda2”. Para fazer isto é necessário selecionar o componente “Legenda2” e no campo Texto digitar “Tempo Restante: 10”.



Verifique se o texto foi inserido na tela do seu celular!

Agora selecione a paleta “Sensores” e arraste o componente **Temporizador** até a tela.



Verifique os componentes invisíveis e veja se foi adicionado o “Temporizador2”.



Devemos também configurar o “Temporizador2” mantendo ele responsável por realizar disparos contínuos a cada 1 segundo (1000 milissegundos) e por enquanto deixá-lo desativado.

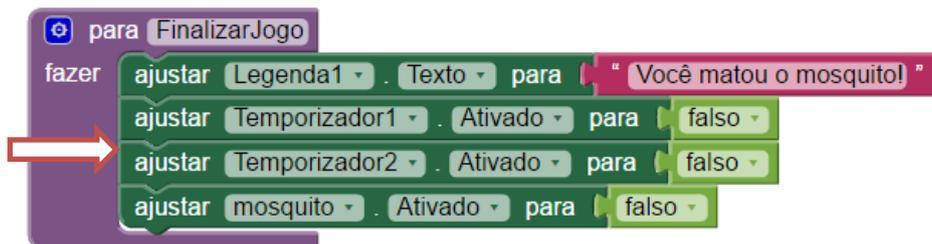


Voltamos ao “Blocos” para adaptar o nosso jogo ao tempo existente para o jogo terminar.

Primeiramente vamos ativar o nosso “Temporizador2” quando o jogo for iniciado. Incluído o bloco “ajustar Temporizador2.Ativado” com valor verdadeiro no procedimento “quando Botao1.Clique”.



E também devemos desativá-lo quando o jogo for finalizado. Incluído o mesmo bloco com valor falso no procedimento “FinalizarJogo”.



Agora precisamos criar uma variável global responsável por armazenar o tempo restante do jogo e iniciando com valor “10”. Também devemos ajustar o seu valor para “10” no procedimento de iniciar o jogo, para evitar o problema ao iniciar um novo jogo.

Precisamos atualizar o tempo restante quando o nosso “Temporizador2” disparar avisando que se passou 1 segundo. Para isto, vamos adicionar no nosso visualizador o bloco “quando Temporizador2.Disparo” e sempre que ele disparar vamos diminuir o valor da variável “tempoRestante” em “1”.

O valor do tempoRestante está sendo diminuído, entretanto o jogador ainda não está percebendo isto. Precisamos atualizar o texto da “Legenda2” para o valor do tempoRestante. Para isto vamos criar um procedimento “AtualizarTempoRestante”, similar ao procedimento “AtualizarVidaRestante”, responsável por trocar o texto da “Legenda2”.

Após diminuir o valor do “tempoRestante” no bloco “Temporizador2.Disparo” devemos chamar o procedimento “AtualizarTempoRestante” para atualizar o valor mostrado na tela.

No procedimento “quando Botaoiniciar.Clique” também devemos chamar o procedimento “AtualizarTempoRestante”.

```

quando Botão1 .Clique
fazer
  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
  
```

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”.

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

```

quando Temporizador2 .Disparo
fazer
  ajustar global tempoRestante para obter global tempoRestante - 1
  chamar AtualizarTempoRestante
  se obter global tempoRestante = 0
  então
    Aqui iremos finalizar o jogo!
  
```

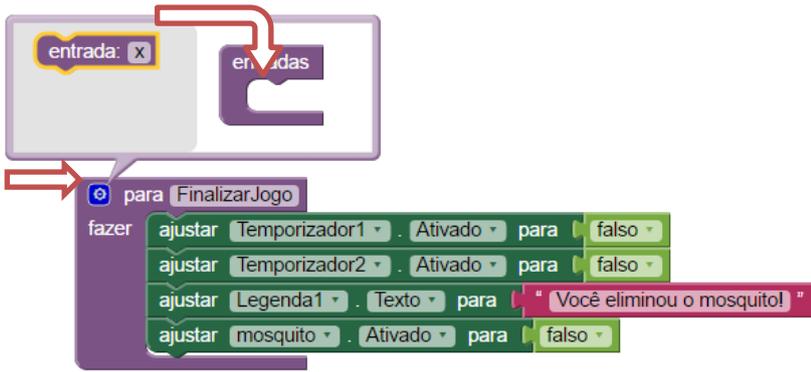
Devemos finalizar o jogo caso o tempo tenha esgotado (dentro do bloco “Se, então” recém criado). Agora precisamos chamar o procedimento finalizar o jogo.

```

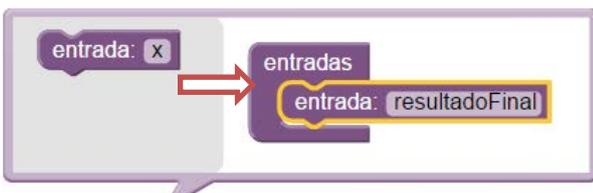
quando Temporizador2 .Disparo
fazer
  ajustar global tempoRestante para obter global tempoRestante - 1
  chamar AtualizarTempoRestante
  se obter global tempoRestante = 0
  então
    chamar FinalizarJogo
  
```

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? Por isto temos que modificar o procedimento “FinalizarJogo” para receber a informação se o jogador ganhou ou perdeu. Para isto, clique no icone 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”.



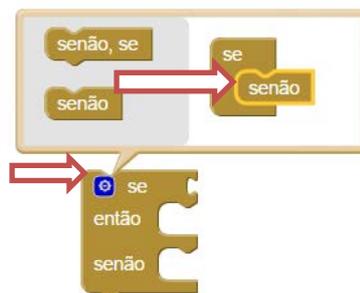
Deve ficar assim:



Agora devemos adicionar um bloco de controle “Se então, Senão” no procedimento “FinalizarJogo” de acordo com o valor da variável local “resultadoFinal”. Caso esse resultado for verdadeiro o jogador foi o vencedor, caso contrário ele perdeu. Isto será informado ao jogador através da “Legenda1”.

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

Para definir o texto de vitória ou derrota precisamos utilizar o bloco de controle “Se, então” e através do icone azul realizar um procedimento similar ao anterior e incluído o bloco “senão” dentro do bloco.



O bloco deve ficar assim:

```

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

Agora devemos atualizar a chamada do procedimento “FinalizarJogo” no bloco “Mosquito.Tocou” passando o valor “Verdadeiro” como parâmetro. Pois neste procedimento o jogador foi vencedor.

```

quando mosquito . Tocou
fazer
  ajustar global vidaRestante para obter global vidaRestante - 10
  chamar AtualizarVidaRestante
  chamar Som1 . Vibrar milisegs 100
  se obter global vidaRestante = 0
  então chamar FinalizarJogo resultadoFinal verdadeiro
  
```

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.

```

quando Temporizador2 . Disparo
fazer
  ajustar global tempoRestante para obter global tempoRestante - 1
  chamar AtualizarTempoRestante
  se obter global tempoRestante = 0
  então chamar FinalizarJogo resultadoFinal falso
  
```

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**.

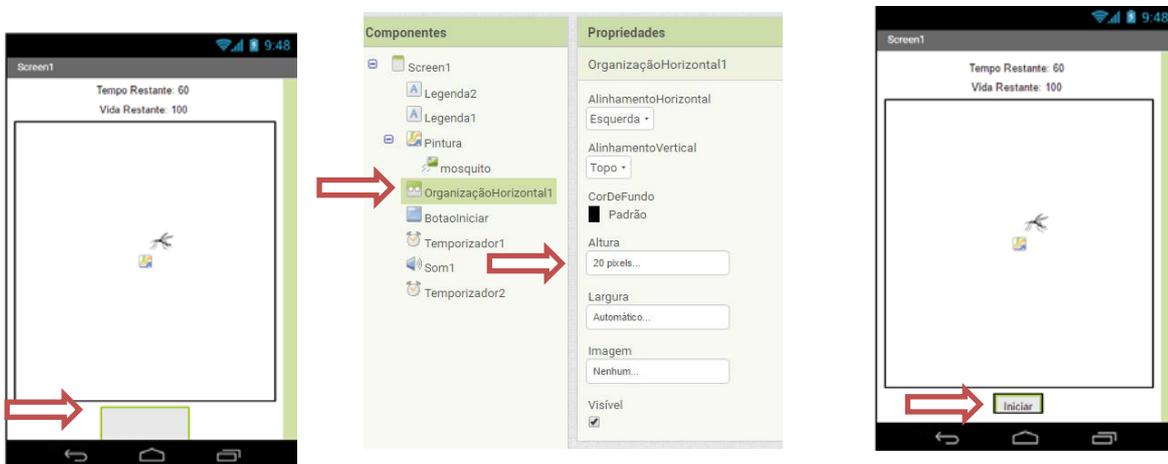
### 3. Adicionar Seleção de Dificuldade.

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

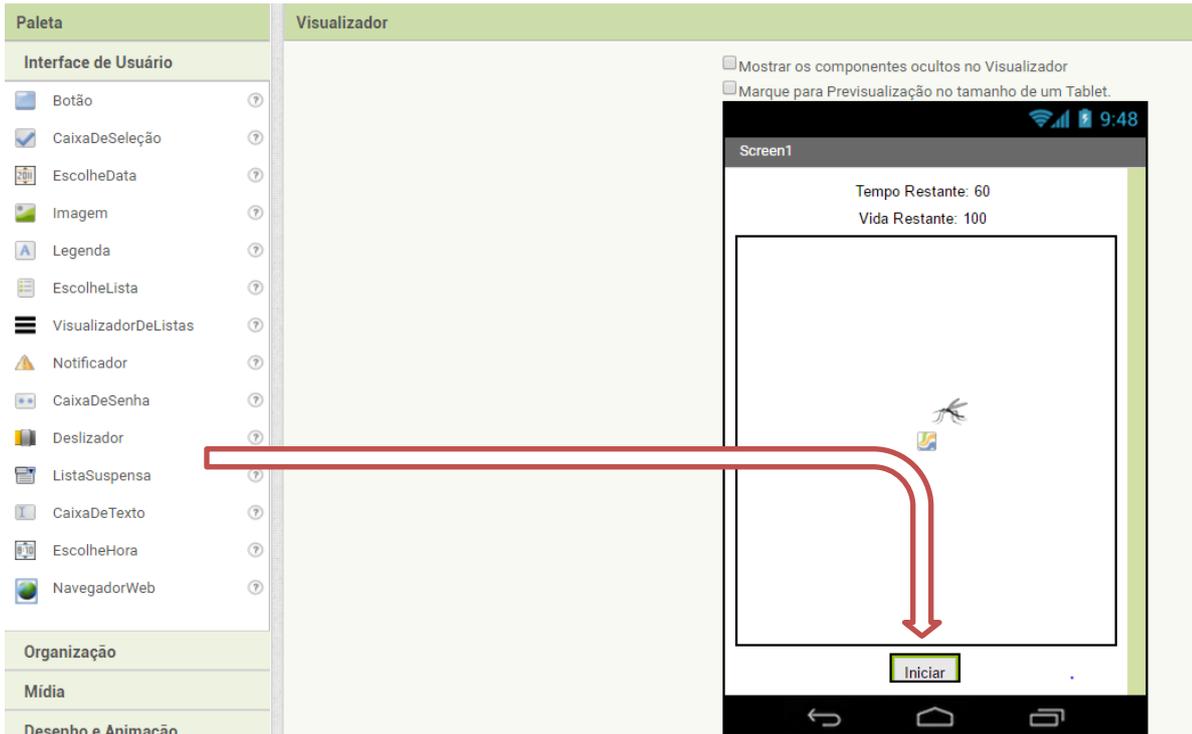
Para exibir na tela os níveis de dificuldade vamos adicionar o componente **ListaSuspensa** ao lado do botão “BotãoIniciar”. Para isto devemos ir ao “Designer” e selecionar a paleta Organização e arrastar o componente **OrganizaçãoHorizontal** para a posição acima do “BotãoIniciar”.



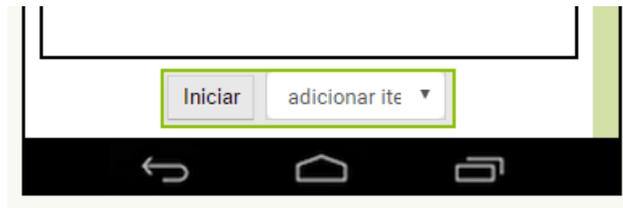
O componente **OrganizaçãoHorizontal** acabou por colocar o “BotãoIniciar” para fora da tela, por isso temos que colocar o seu tamanho como “20” para altura. E arrastar o botão “BotãoIniciar” para dentro da “OrganizacaoHorizontal”.



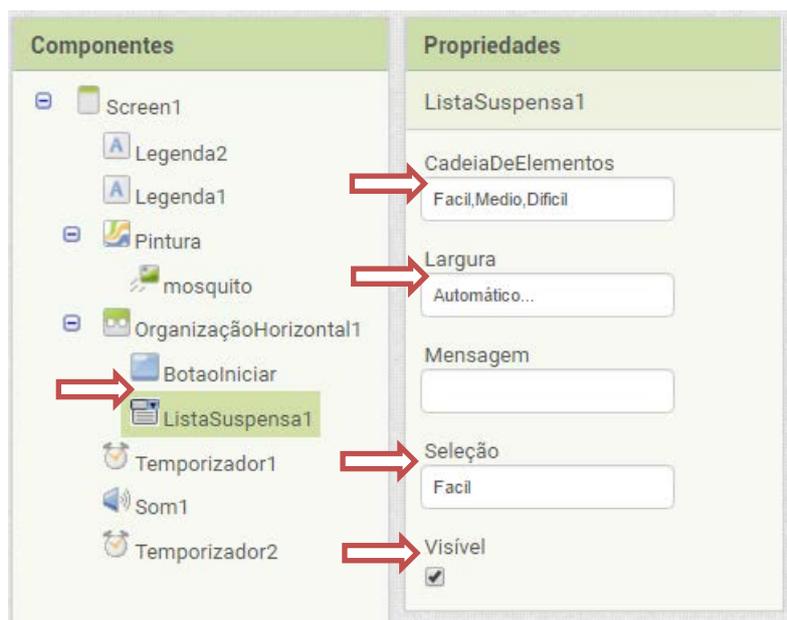
Agora devemos clicar na paleta Interface de Usuário e arrastar o componente **ListaSuspensa** para dentro da “OrganizaçãoHorizontal” recém adicionada.



Após arrastar, devemos trocar a altura da “Organização Horizontal” para automática.



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



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

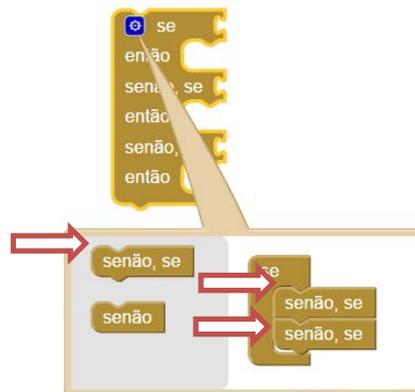
Após adicionar a “ListaSuspensa1” no bloco “Designer” precisamos voltar ao “Blocos” para configurar a sua funcionalidade.

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



Iremos definir que o nivelSelecionado receberá os seguintes valores: Facil = 1, Medio = 2, Dificil = 3.

Para isto, utilizaremos o bloco “quando ListaSuspensa1.DepoisDeSelecionar” para definir qual o valor da variável “nivelSelecionado”. Este bloco possui uma váriavel local chamada de “seleção” a qual contém o valor “Facil”, “Medio” ou “Dificil” de acordo com a seleção realizada. 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” como feito na imagem acima. Nossa lógica será:

Se (seleção = “Facil”)

Então nivelSelecionado = 1

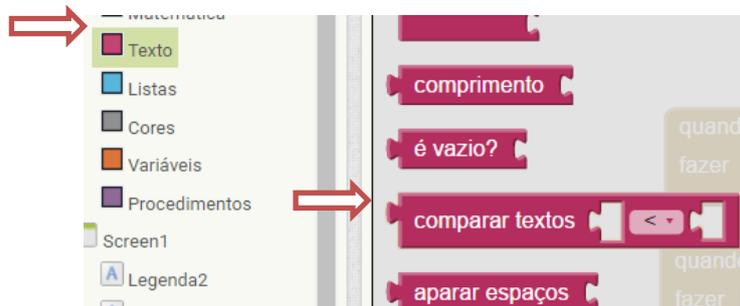
Senão Se (seleção = “Medio”)

Então nivelSelecionado = 2

Senão Se (seleção = “Dificil”)

Então nivelSelecionado = 3

Contudo como o valor “seleção” é um texto, **devemos utilizar o bloco de comparação entre textos** para saber se o valor de “seleção = “Facil”” ou algum dos outros valores definidos. Devemos substituir o símbolo “<” por “=”.



Nosso bloco de código deve ficar:



Agora que temos a variável “nivelSelecionado” contendo o valor de acordo com o que está selecionado na “ListaSuspensa1”, devemos utilizar esta informação no procedimento “quando BotaIniciar.Clique” para alterar os valores de “tempoRestante” e alterar o valor “Intervalo” do “Temporizador1”. Lembrar que o “Temporizador.1” é o responsável por mandar o mosquito se mover.

Primeiro devemos adicionar o mesmo bloco “Se, então” com dois blocos “Senão, Se” igual realizado no passo anterior, trocando os valores das operações lógicas “=” para “nivelSelecionado = 1”, “nivelSelecionado=2” e “nivelSelecionado=3”.

```

para IniciarJogo
fazer
  se
    obter global nivelSelecioneado = 1
  então
  senão
    obter global nivelSelecioneado = 2
  então
  senão
    obter global nivelSelecioneado = 3
  então

```

Dentro dos 3 “Então” do bloco, devemos atualizar o valor “tempoRestante” e ajustar o “intervalo” do “temporizador1”.

Vamos definir as dificuldades de acordo com a seguinte regra:

Fácil – variável “tempoRestante” = 20

Intervalo “Temporizador1” = 500

Médio – variável “tempoRestante” = 10

Intervalo “Temporizador1” = 400

Difícil – variável “tempoRestante” = 8

Intervalo “Temporizador1” = 350

Devemos passar essas regras para dentro do procedimento “IniciarJogo” ajustando o valor da variável “tempoRestante” e ajustando o valor “intervalo” do “Temporizador1. Também devemos remover o ajuste da variável “tempoRestante” a qual havia sido adicionada anteriormente no bloco “IniciarJogo”, deixando somente os ajustes adicionados neste momento.

Fácil =

```

quando Botão1 .Clique
fazer
  se
    obter global nivelSelecioneado = 1
  então
    ajustar global tempoRestante para 20
  então
    ajustar Temporizador1 Intervalo para 500

```

Bloco completo:

```

quando Botão1 Clique
fazer
  se
    obter global nivelSelecionado = 1
    ajustar global tempoRestante para 20
    ajustar Temporizador1 Intervalo para 500
  senão, se
    obter global nivelSelecionado = 2
    ajustar global tempoRestante para 10
    ajustar Temporizador1 Intervalo para 400
  senão, se
    obter global nivelSelecionado = 3
    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
  
```

Removido o bloco “ajustar tempoRestante”

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

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

Para fazer a instalação da versão final no seu smartphone basta clicar em “Compilar-> App ( Fornecer o QR Code para .apk)” e usar o Aplicativo MIT AI2 Companion já instalado no seu smartphone clicar em “Scan QR code” e prosseguir com a instalação.

