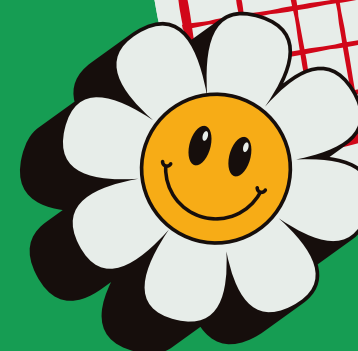
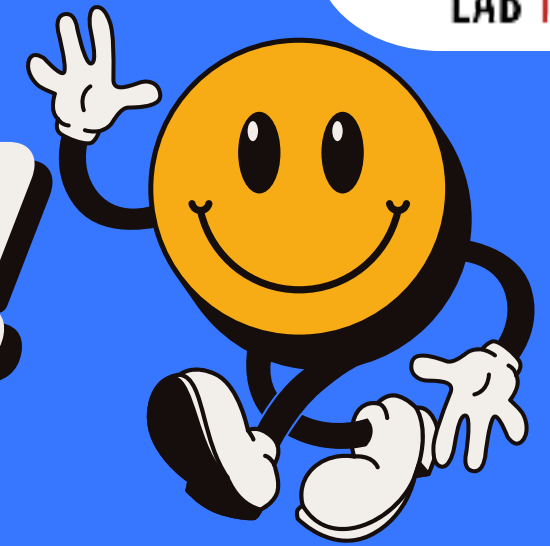


MIT APP INVENTOR





BEM - VINDOS!



AGENDA

Hoje, iremos relembrar alguns conceitos de aulas passadas:

- **Básico de Programação**
- **Programação em Blocos**

E depois iremos apresentar uma nova ferramenta onde pode-se criar aplicativos para celular e tablets

REVISÃO DE PROGRAMAÇÃO

Variáveis

Váriáveis são **armazenamentos** de informações. Elas **guardam** algo para ser usado em algum código que vem em sequência

```
VARIABEL1 = 14  
VARIABEL2 = "OLÁ"  
VARIABEL3 = TRUE
```

- 1 – Não podem iniciar com números
- 2 – Não podem ter caracteres especiais (ex: ` , @, #...)
- 3 – Não podem ter espaços
- 4 – Possuem tipos

REVISÃO DE PROGRAMAÇÃO

Entrada e Saída

Também chamado de **input** e **output**, é a forma que dados são **inseridos** em um programa e como suas **saídas** são **visualizadas**

ENTRADA (INPUT)

```
VARIAVEL = int(input())  
  
print("O usuário  
inseriu", VARIAVEL)
```

SAÍDA (OUTPUT)

O usuário inseriu 2

- 1 – int(input()) espera que o usuário insira obrigatoriamente um inteiro em VARIAVEL
- 2 – O programa fica bloqueado enquanto a entrada não for inserida
- 3 – print() é a função de impressão

REVISÃO DE PROGRAMAÇÃO

Laços

Garantem **repetição** de um código até uma **condição** ser satisfeita, um exemplo prático seria, “**enquanto** não acaba a aula, preste atenção”

FOR

```
for i in range(0,2):  
    print(i * 2)
```

WHILE

```
while(i != 1)  
    i = 0  
    print(i)  
    i += 1
```

O que será imprimido nos dois casos e quantas vezes o laço é repetido?



REVISÃO DE PROGRAMAÇÃO

Condicionais

Fazem com que o código seja executado mediante uma condição, um exemplo prático seria “**se** for chover hoje, então leve guarda-chuva, **se não**, deixe em casa”

SE

```
if (chover == True):  
    guarda_chuva = True
```

SE NÃO

```
else:  
    guarda_chuva = False
```

- 1 – Podem ser aninhados (repetidos) um dentro do outro para criar condições bem específicas
- 2 – Podem existir mais de uma condição dentro de um if
- 3 – Nem todo if precisa de um else

INTRODUÇÃO

O QUE É MIT APP INVENTOR?

App Inventor é um serviço gratuito baseado em nuvem que permite criar seus próprios aplicativos móveis usando uma linguagem de programação baseada em blocos

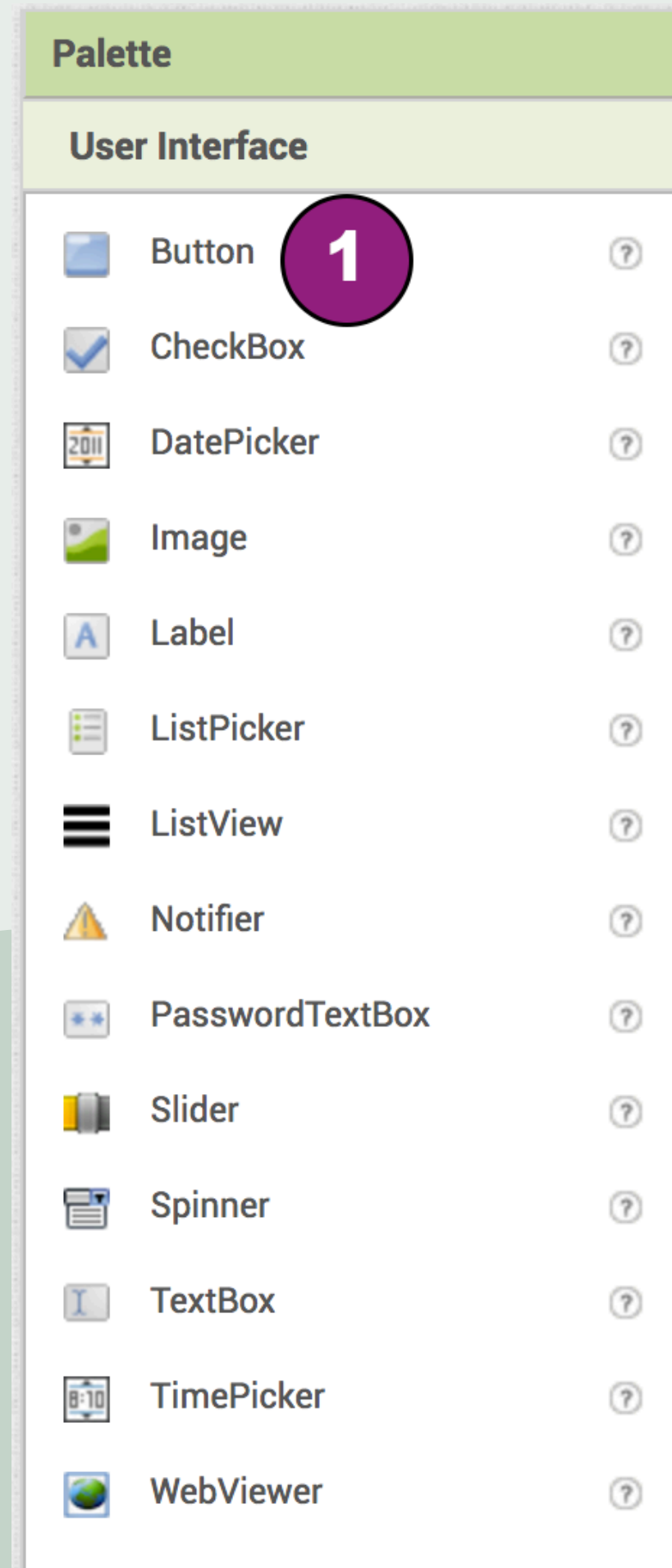


Olá, me chamo
Mita, irei te
acompanhar
nessa jornada



Para acessar o
MIT, scanei
QrCode ou
acesse o site





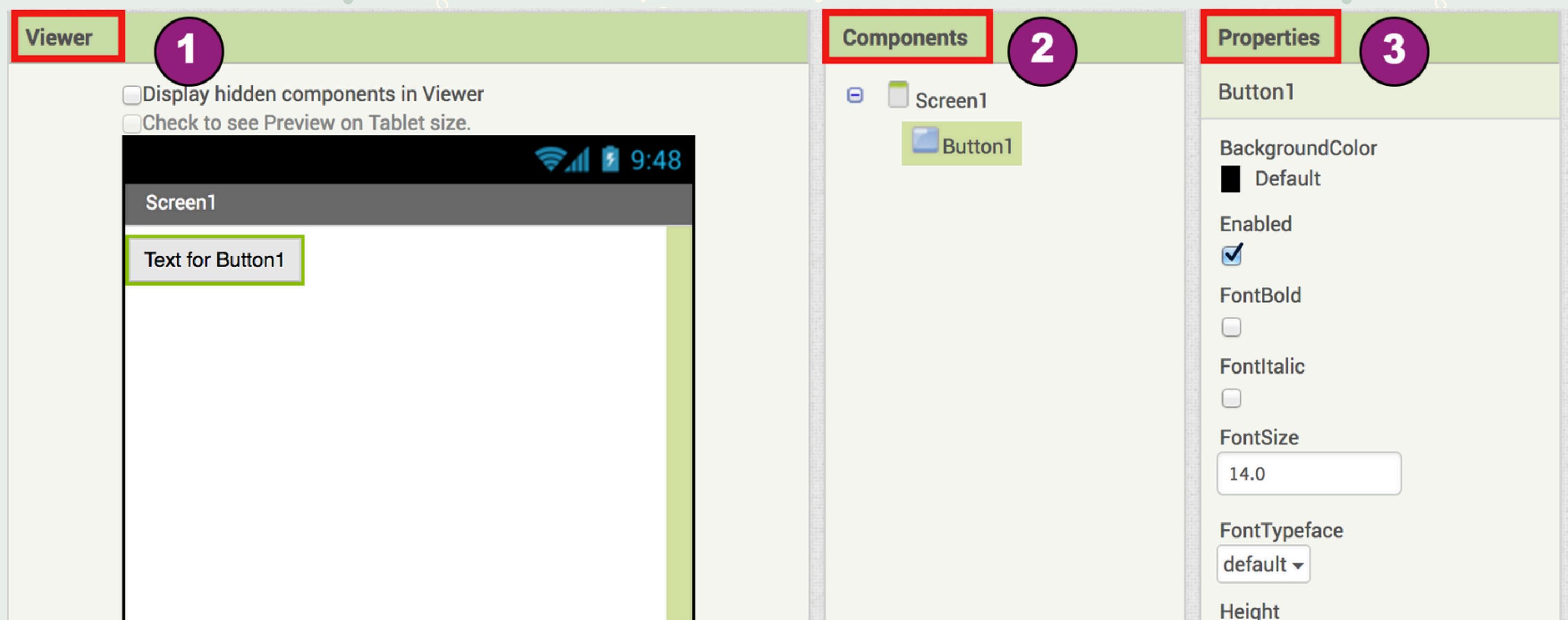
Os componentes do App Inventor estão localizados no lado esquerdo da janela Designer sob o título “Palette”.

**OS COMPONENTES
SÃO COMO OS
INGREDIENTES DE
UMA RECEITA!**

**COMPONENTES SÃO
OS ELEMENTOS
BÁSICOS QUE VOCÊ
USA PARA CRIAR
APLICATIVOS EM UM
TELEFONE ANDROID
OU IPHONE.**

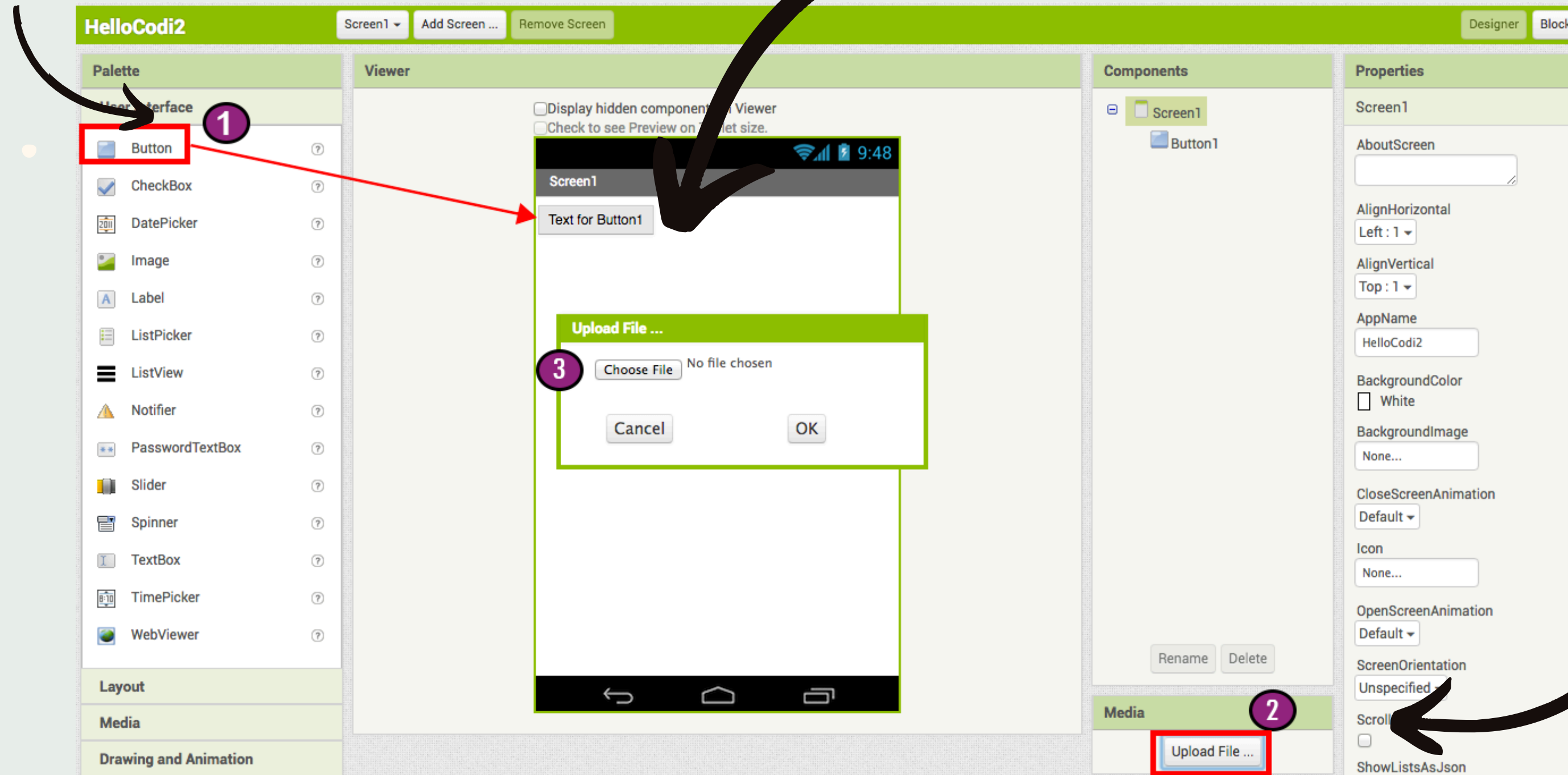


Para usar um Componente em seu aplicativo, você precisa clicar e arrastá-lo para o visualizador no meio do Designer. Quando você adiciona um componente ao Visualizador, ele também aparecerá na lista de componentes no lado direito do Visualizador.



Na paleta User Interface, arraste e solte o componente Button na Screen1

Uma janela aparecerá para permitir que você escolha o arquivo de imagem. Clique em "Procurar" e navegue até o local do arquivo codi.jpg que você baixou anteriormente



Para dar ao botão a imagem da abelha, no painel Propriedades, em Imagem, clique no texto "Nenhum..." e clique em "Carregar arquivo"



BackgroundColor do Label1 clicando na caixa. Você pode alterá-lo para qualquer cor que desejar



FontSize para 30

Propriedade de texto de Label1 para ler "Touch the Bee". Você verá a alteração do texto no Designer e no seu dispositivo

TextColor para qualquer cor que você desejar



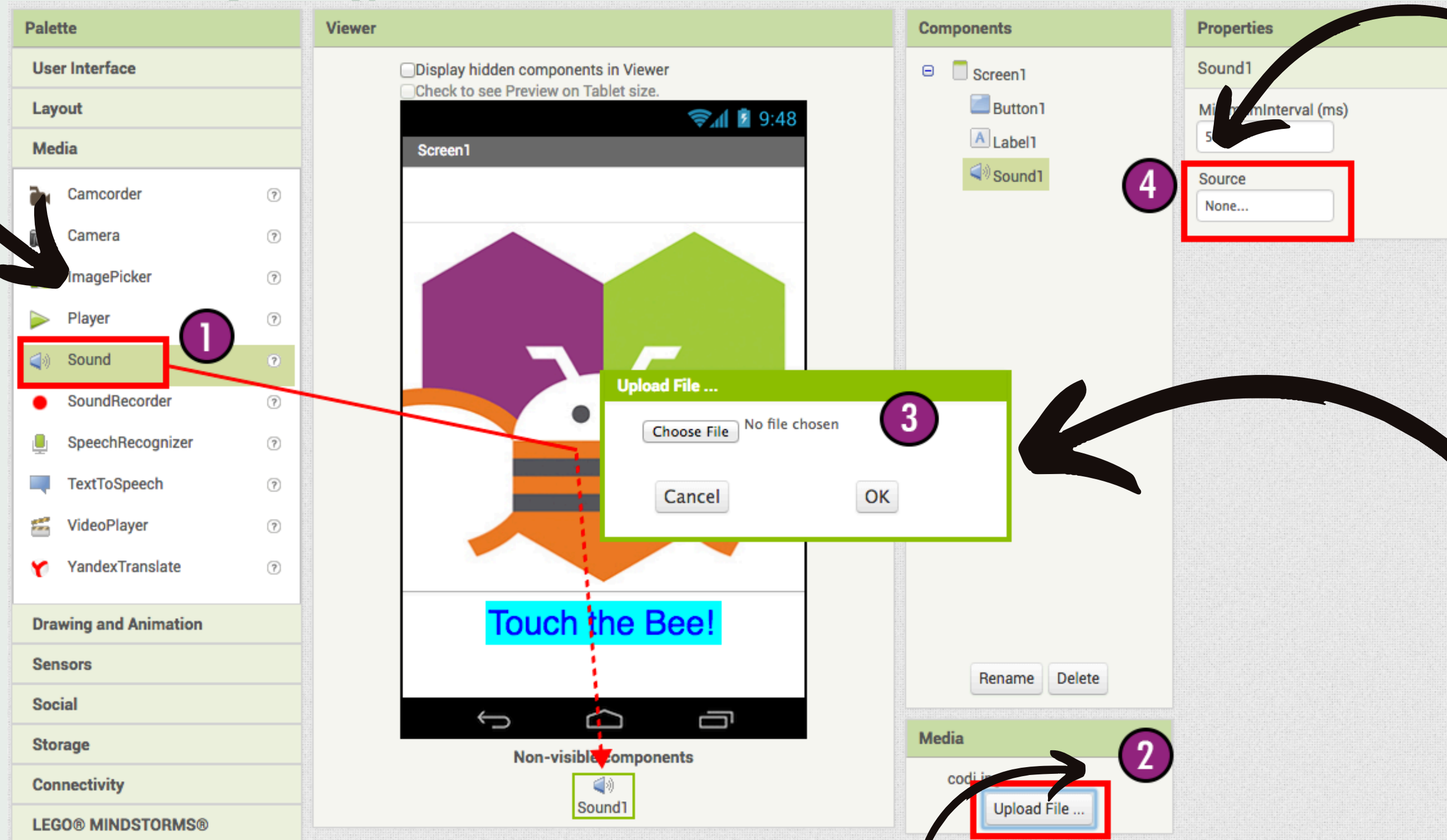
Em Palette, clique na gaveta Media e arraste um componente Sound e coloque-o no Viewer


No painel Propriedades, veja que a propriedade Source atualmente diz None....

Clique na palavra None... para alterar a fonte do componente Sound1 para Bee-Sound.mp3

Navegue até o local do arquivo Bee-Sound.mp3 que você baixou anteriormente e carregue-o neste projeto

No painel Mídia, clique em Carregar arquivo





Mas, e a
programação?
Não vamos ver?

**VAMOS
NESSA!**



Os blocos marrons são chamados de blocos manipuladores de eventos.

Os blocos do manipulador de eventos especificam como o dispositivo móvel deve responder a determinados eventos



```
when Button1 .GotFocus  
do
```

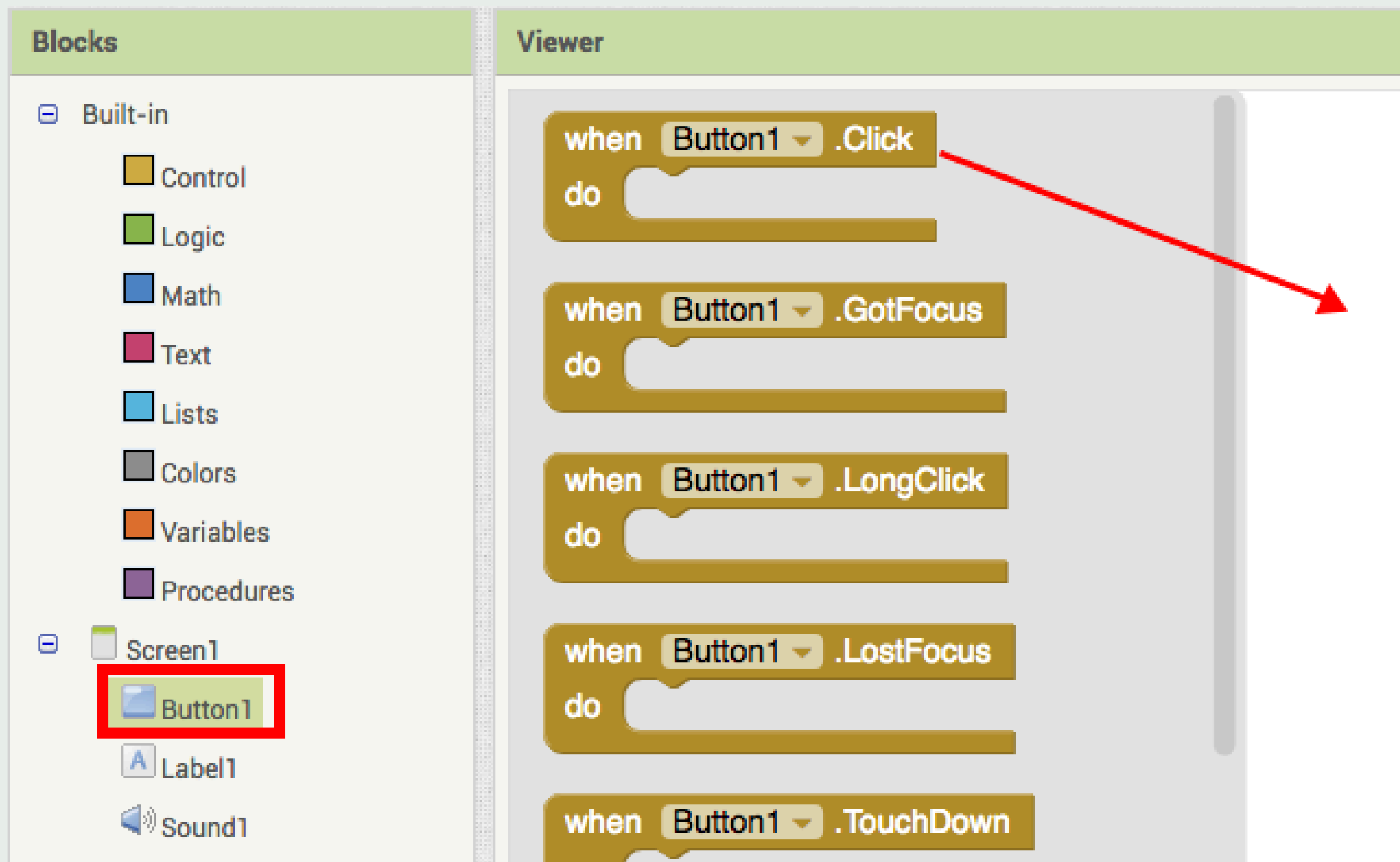
```
when Button2 .Click  
do  
  call FusiontablesControl1 .GetRows  
    tableId  
    columns
```

Os blocos roxos são chamados de blocos de comando, que são colocados no corpo dos manipuladores de eventos. Quando um manipulador de eventos é executado, ele executa a sequência de comandos em seu corpo.





No lado esquerdo do Blocks Editor,
clique na gaveta Button1 para abri-la.
Arraste e solte o bloco When
Button1.Click na área de trabalho



Clique no botão Blocos no
canto superior direito da
tela para acessar o
Editor de Blocos.



Blocos
manipuladores
de comando



Clique na gaveta Sound1 e arraste o bloco Sound1.Play e conecte-o à seção "fazer" do bloco quando Button1.Click.

Blocks

Built-in

Control

Logic

Math

Text

Lists

Colors

Variables

Procedures

Screen1

Button1

Label1

Sound1

Any component

Viewer

when Sound1 .SoundError

message

do

call Sound1 .Pause

call Sound1 .Play

call Sound1 .Resume

call Sound1 .Stop

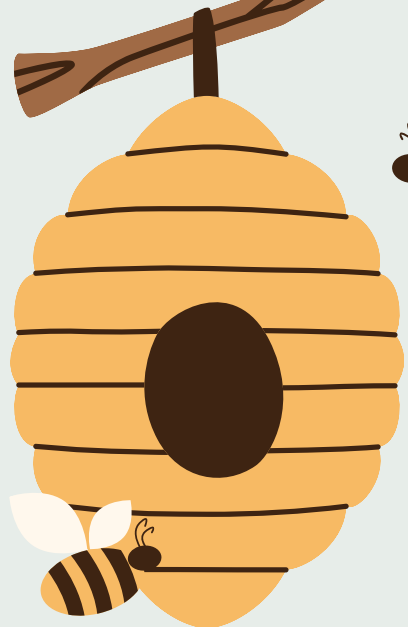
call Sound1 .Vibrate

milliseconds

Sound1 . MinimumInterval (ms)

when Button1 .Click
do call Sound1 .Play

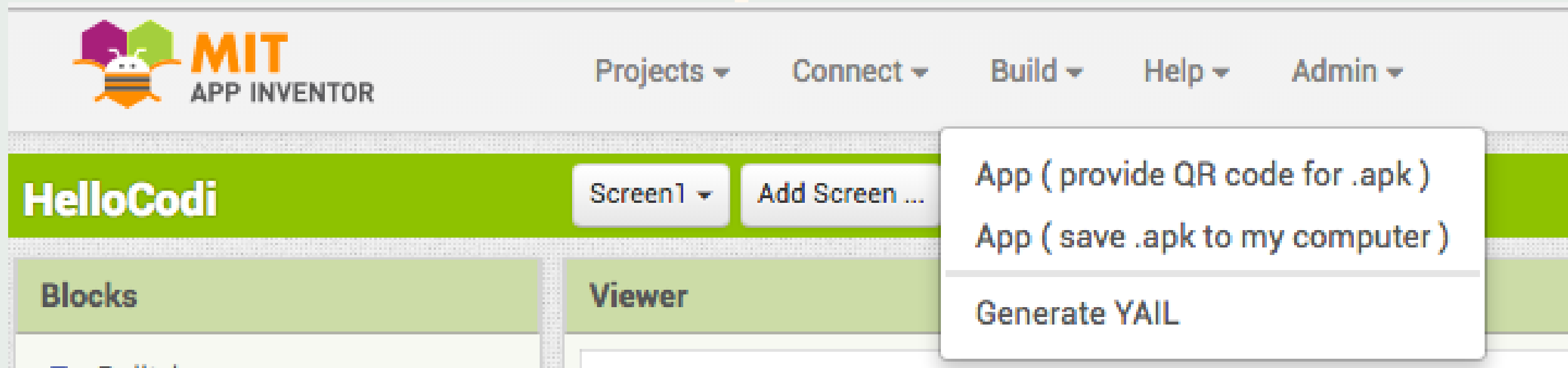
OS BLOCOS SE
CONECTAM COMO
PEÇAS DE UM
QUEBRA-CABEÇA



AGORA É SÓ
EMPACOTAR



Para "empacotar" o aplicativo para instalar no seu dispositivo ou enviar para outra pessoa, clique na guia Construir na parte superior da tela. Em Build, há duas opções disponíveis para você escolher:



Parabéns, seu primeiro aplicativo está
em execução!



Experimente no seu dispositivo
ou com o emulador! Ao
pressionar o botão, você deverá
ouvir o zumbido da abelha



***VOCÊ TEM
ALGUMA
DÚVIDA?***



PODE PERGUNTAR!

OBRIGADO!



CODELAB TEEN