

Programação para Dispositivos Móveis

Curso de Ciência da Computação

Universidade Paulista (UNIP)

UNIP

UNIVERSIDADE PAULISTA

CRIANDO A INTERFACE DO USUÁRIO

Prof. Ms. Clayton A. Valdo
clayton.valdo@docente.unip.br

Prof. Ms. Peter Jandl
peter.junior@docente.unip.br

Prof. Ms. Télvio Orrú
telvio.orrú@docente.unip.br

AULA 2



O que é Interface de Usuário?

A Interface de Usuário (IU) tem relação com a **apresentação do App** e consiste em **três fatores**:

- ❑ **Aparência do App:** design gráfico, tipografia (fontes) e cores;
- ❑ **Interação com o App:** como as pessoas dizem ao App o que fazer;
- ❑ **Informações:** como são estruturadas e mostradas aos usuários: *tela, autofalante, etc.*



Veja um Exemplo: Angry Birds

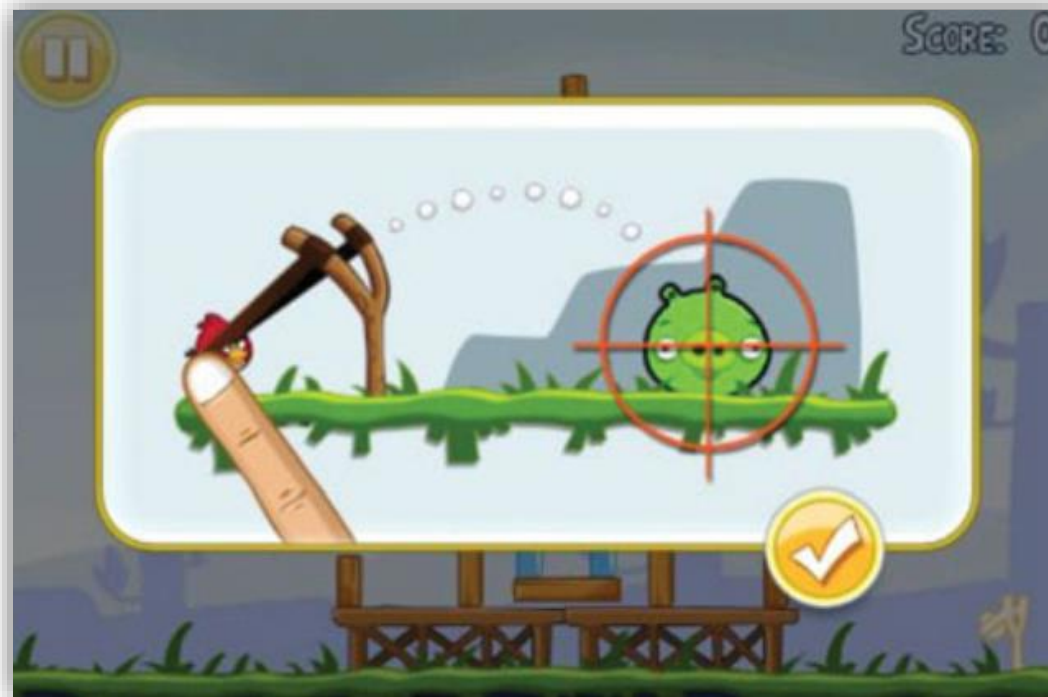
- ❑ **Angry Birds** é um dos jogos mais populares disponíveis para a plataforma android;
- ❑ **Possui uma ótima aparência:** gráficos, sons, imagens e textos trabalham juntos em sintonia;
- ❑ **É fácil de jogar:** o usuário apenas precisa informar a velocidade e o ângulo para que o pássaro seja disparado;
- ❑ Os **designers do game** ainda tiveram a **brilhante ideia** de permitir essa **interação** de forma **similar a uma catapulta**.



Entrada > Processamento > Saída

Entrada:

O usuário diz ao celular para **fazer alguma coisa**.
Ex: clicar e arrastar um objeto.



Saída:

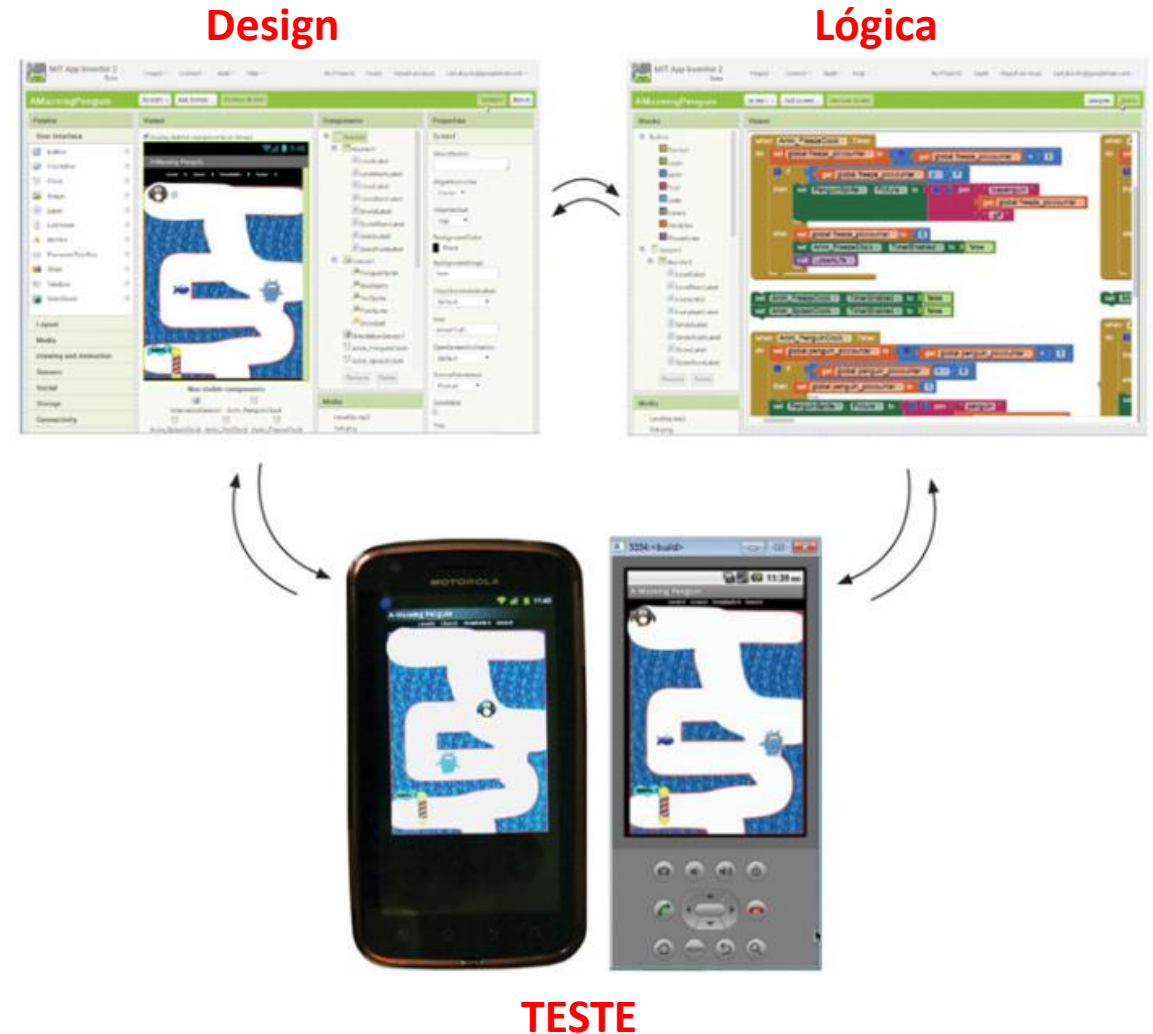
O celular mostra o **pássaro voando e colidindo**, mudar o **placar** e **tocar diferentes sons**.

Processamento:

Ao soltar o pássaro, o processador **calcula a sua trajetória**, o seu **local de pouso** e como o **panorama será afetado**.

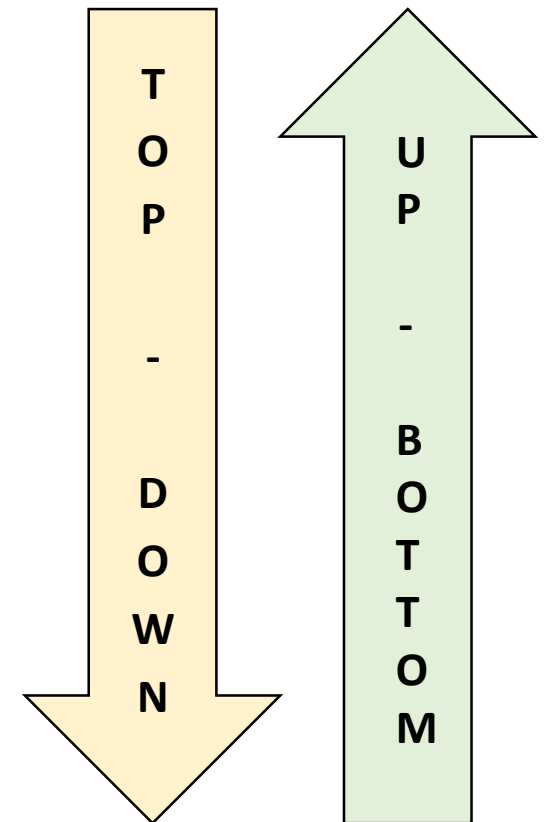
Tela de Designer e a IU

- ❑ A Interface do Usuário (IU) é criada na **Tela de Designer**;
- ❑ O **Editor de Blocos** permite dizer ao App o que ele deve fazer;
- ❑ O teste do app acontece no **Smartphone ou Emulador**;
- ❑ Você alternará **constantemente** entre essas tarefas.



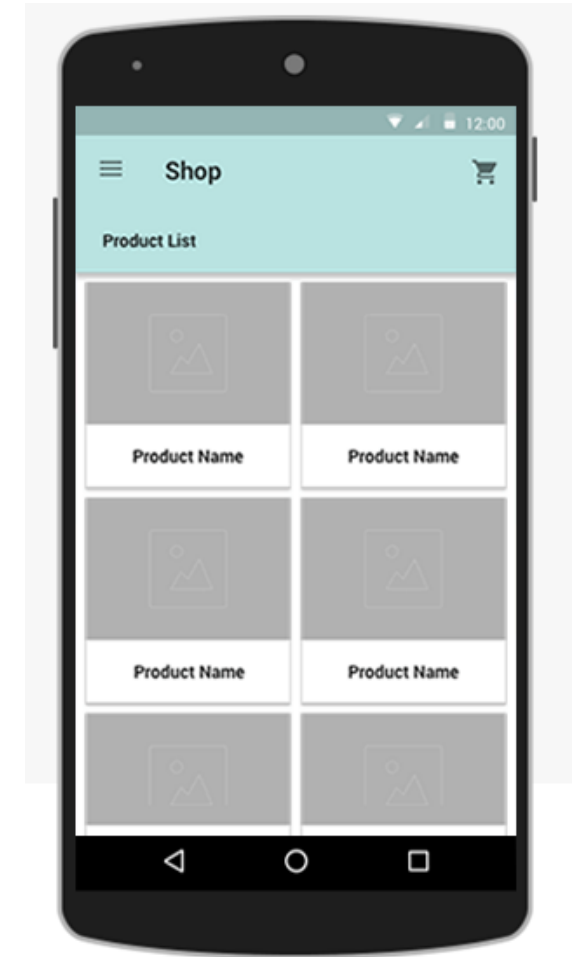
Abordagens de Desenvolvimento

- ☐ No App Inventor, **primeiro** você desenvolve a IU para depois codificar a lógica (*top-down*);
- ☐ Em outros programas é possível **primeiro** codificar a **lógica** e depois se preocupar com a IU (*bottom-up*);
- ☐ Ambas **possuem vantagens e desvantagens**. Na prática fazemos **um pouco de cada** uma delas.



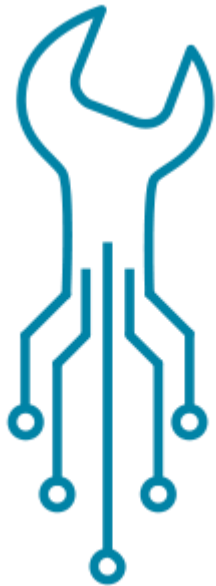
Importância dos Componentes para a IU

- ☐ Você já deve ter reconhecido muitos dos **componentes utilizados no App Inventor**;
- ☐ **Botões, etiquetas e caixas de texto** são muito comuns em **programas e páginas web**;
- ☐ Os **componentes disponíveis** facilitam muito o processo de **criação da IU**.



Componentes do App Inventor

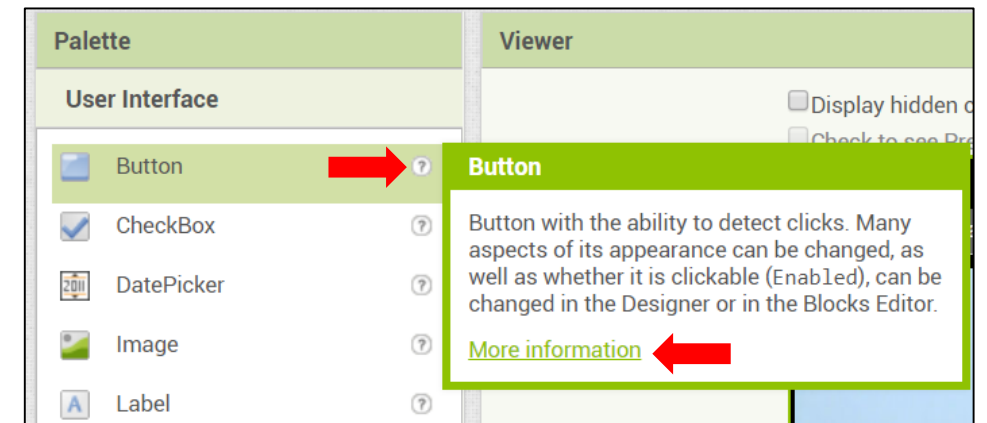
Esses são **alguns componentes** do App Inventor. Não fique preocupado com a quantidade. Ao **passar das aulas** vamos **conhecendo suas funcionalidades**.



User Interface	Layout	Media	Social
Button	HorizontalArrangement	Camcorder	ContactPicker
CheckBox	TableArrangement	Camera	EmailPicker
Clock	VerticalArrangement	ImagePicker	PhoneCall
Image		Player	PhoneNumberPicker
Label	Drawing and Animation	Sound	Texting
ListPicker	Ball	SoundRecorder	Twitter
Notifier	Canvas	SpeechRecognizer	
PasswordTextBox	ImageSprite	TextToSpeech	Sensors
Slider	Connectivity	VideoPlayer	AccelerometerSensor
TextBox	ActivityStarter	LEGO® MINDSTORMS®	BarcodeScanner
WebView	BluetoothClient	NxtColorSensor	LocationSensor
	BluetoothServer	NxtDirectCommands	NearField
	Web	NxtDrive	OrientationSensor
	Storage	NxtLightSensor	
	FusiontablesControl	NxtSoundSensor	
	TinyDB	NxtTouchSensor	
	TinyWebDB	NxtUltrasonicSensor	

Dica sobre Componentes

- ❑ Toda aula novos componentes serão apresentados (*procure estudar sobre cada um deles*);
- ❑ É possível obter informações sobre cada componente clicando no **ponto de interrogação**;
- ❑ Se necessitar de mais informações, clique no link: ***“More Information”***.



Criando uma IU na Prática

Propósito do App:

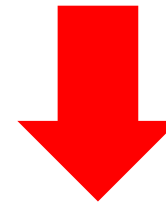
Apresentar a imagem de uma ovelha em um pasto. Quando o usuário clicar nela, ela deverá berrar.



Conteúdo Necessário:

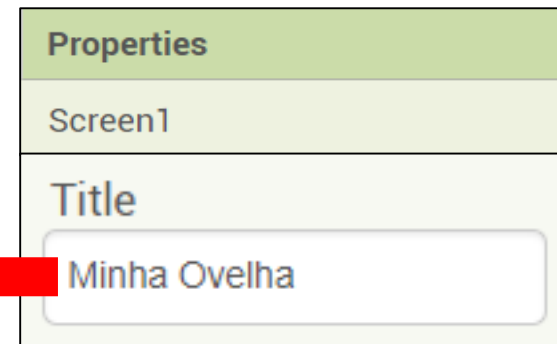
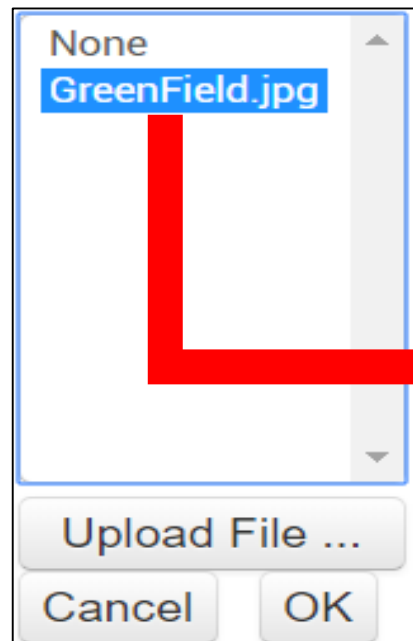
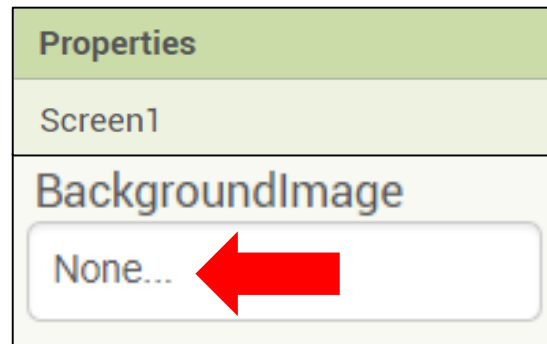
Imagens: ovelha e campo;

Sons: “berro da ovelha”.



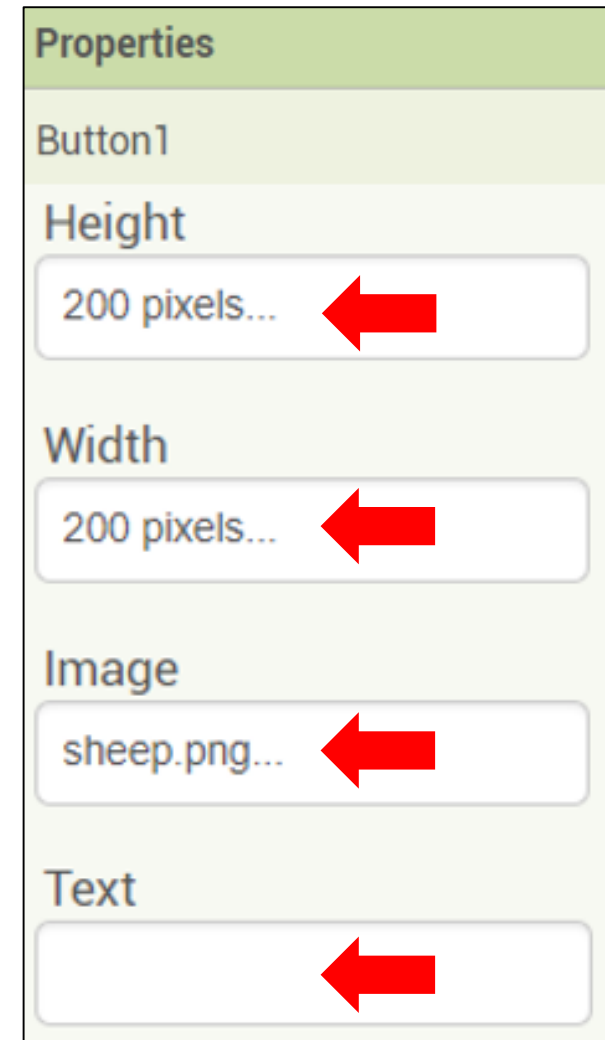
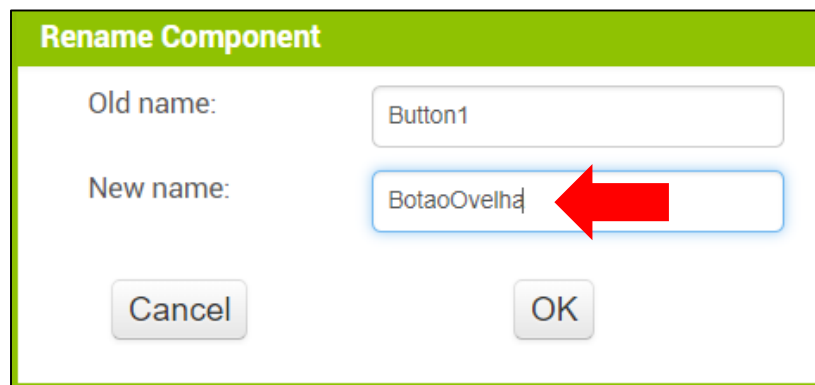
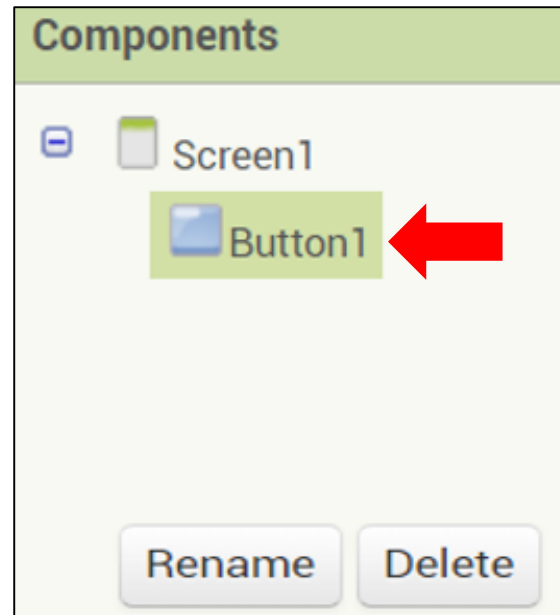
Disponível em nossa pasta compartilhada:
“Recursos Usados em Aula”

Passo 1: Defina o Plano de Fundo



Todos os componentes possuem propriedades. Alguns tem poucas, outros muitas. Uma propriedade são aspectos que podem ser mudados no componente.

Passo 2: Acrescente um Botão



Imagens Não Possuem Eventos!

- ❑ Há muitos componentes capazes de detectar entradas, como o toque;
- ❑ As imagens, por sua vez, não identificam eventos, pois *não possuem event handlers*;
- ❑ Dessa forma, a maneira mais simples de se resolver o problema é utilizando um botão;
- ❑ Portanto, a imagem servirá apenas para fins estéticos, ou seja, para embelezar o App.



Passo 3: Adicione um Componente de Som

The image illustrates the steps to add a sound component in an Android application:

- Palette:** The 'Media' section is expanded, and 'Sound' is selected.
- Components:** The 'Sound1' component is added to the hierarchy under 'Screen1'. A red arrow points to it.
- Properties:** The 'BerroSom' properties are shown. The 'Source' is set to 'baa.wav...'. A red arrow points to the 'Source' field.
- Rename Component:** A dialog box shows the component being renamed from 'Sound1' to 'BerroSom'.

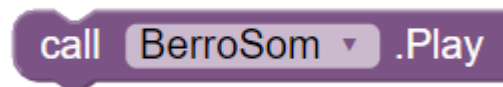
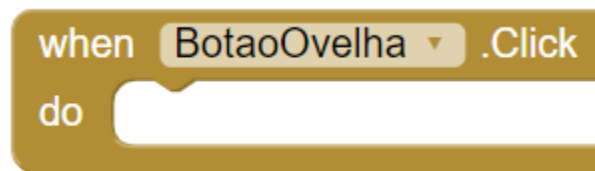
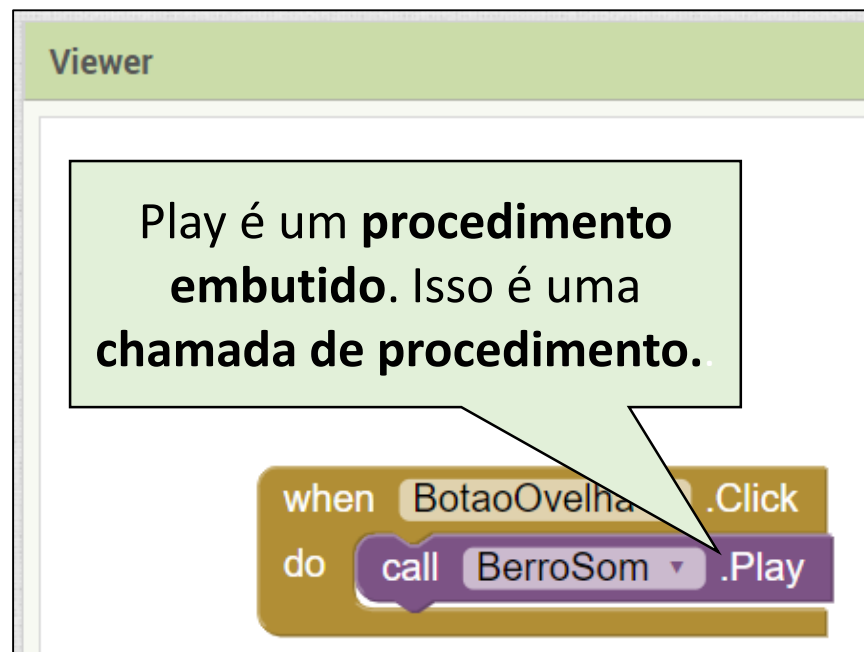
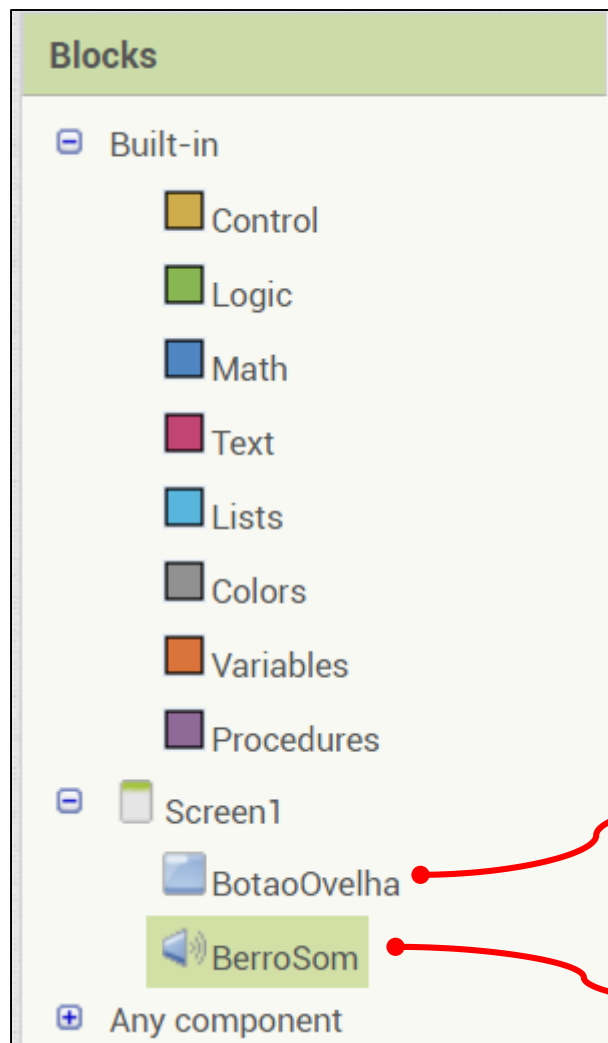
O interessante é usar o formato **WAV** ou **MP3**, pois **funcionam em todos os celulares!**

Componentes Não Visíveis

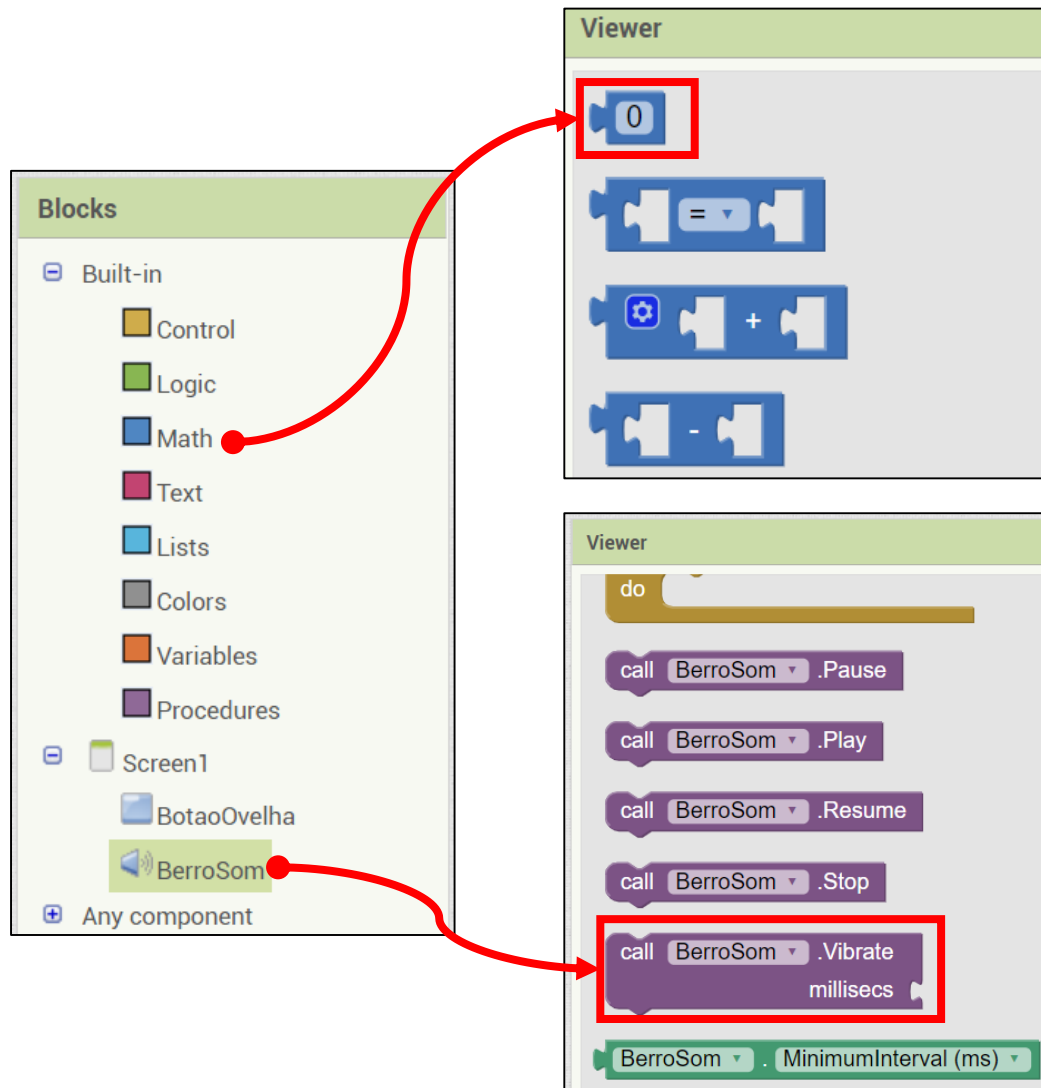
- ☐ Você deve ter notado que o **componente de som não está visível**, ou seja, não aparece na tela;
- ☐ Existem **vários componentes** que também **não são visíveis: câmera, acelerômetro, sensores, etc;**
- ☐ Quando um componente não é visível ele **aparece em uma área específica** (*abaixo do designer*).



Passo 4: Programando os Blocos



Como Tocar e Vibrar?



Antes (apenas som):

```
when BotaoOvelha .Click  
do call BerroSom .Play
```



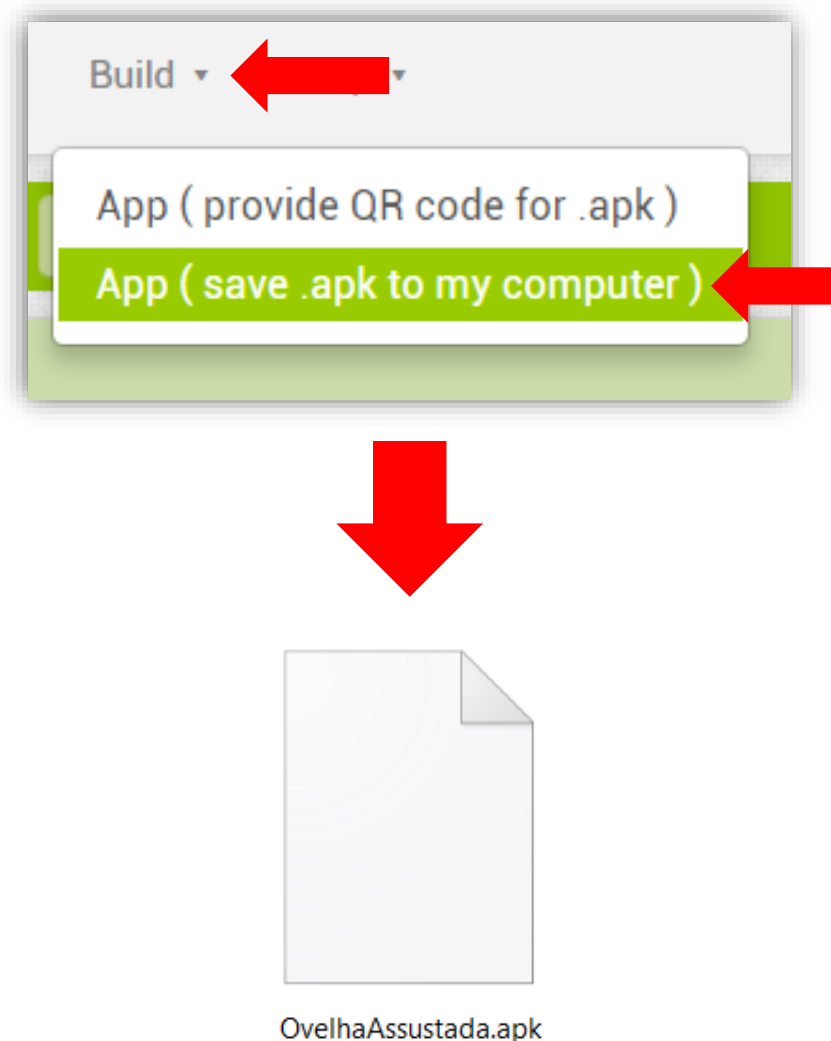
Depois (som e vibração):

```
when BotaoOvelha .Click  
do call BerroSom .Play  
  call BerroSom .Vibrate  
    millisecs 1000
```

Atalho: Clique na área de trabalho, digite 1000 e pressione enter.

Chegou a Hora de Testar!

- ☐ Gere o **APK** do App e transfira para o seu celular;
- ☐ Instale o APK clicando no arquivo transferido;
- ☐ É necessário **habilitar** a instalação de fonte de terceiros;
- ☐ Se tudo der certo, você terá o **App rodando** em seu aparelho.



Desafio: Agora é sua Vez!

Experimente utilizar o procedimento **BerroSom.Vibrate** com diferentes valores para o tempo de execução. Com base nos testes responda:

- a) Qual é o tempo mais curto que pode ser informado e ainda assim sentir o telefone vibrar?
- b) Quantos milissegundos devem ser informados para fazer com que a vibração dure exatamente o tempo do som?



Aumentando a Interação com o App

Propósito do App:

Se clicar na ovelha, ela irá berrar. Balance o telefone para que ela desapareça. Se o usuário clicar no botão “Assustou a ovelha...” ela irá aparecer novamente.



Conteúdo Necessário:

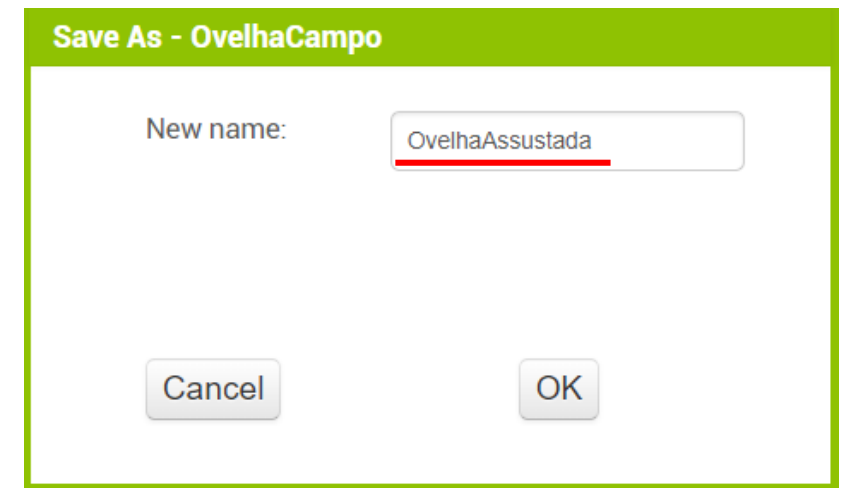
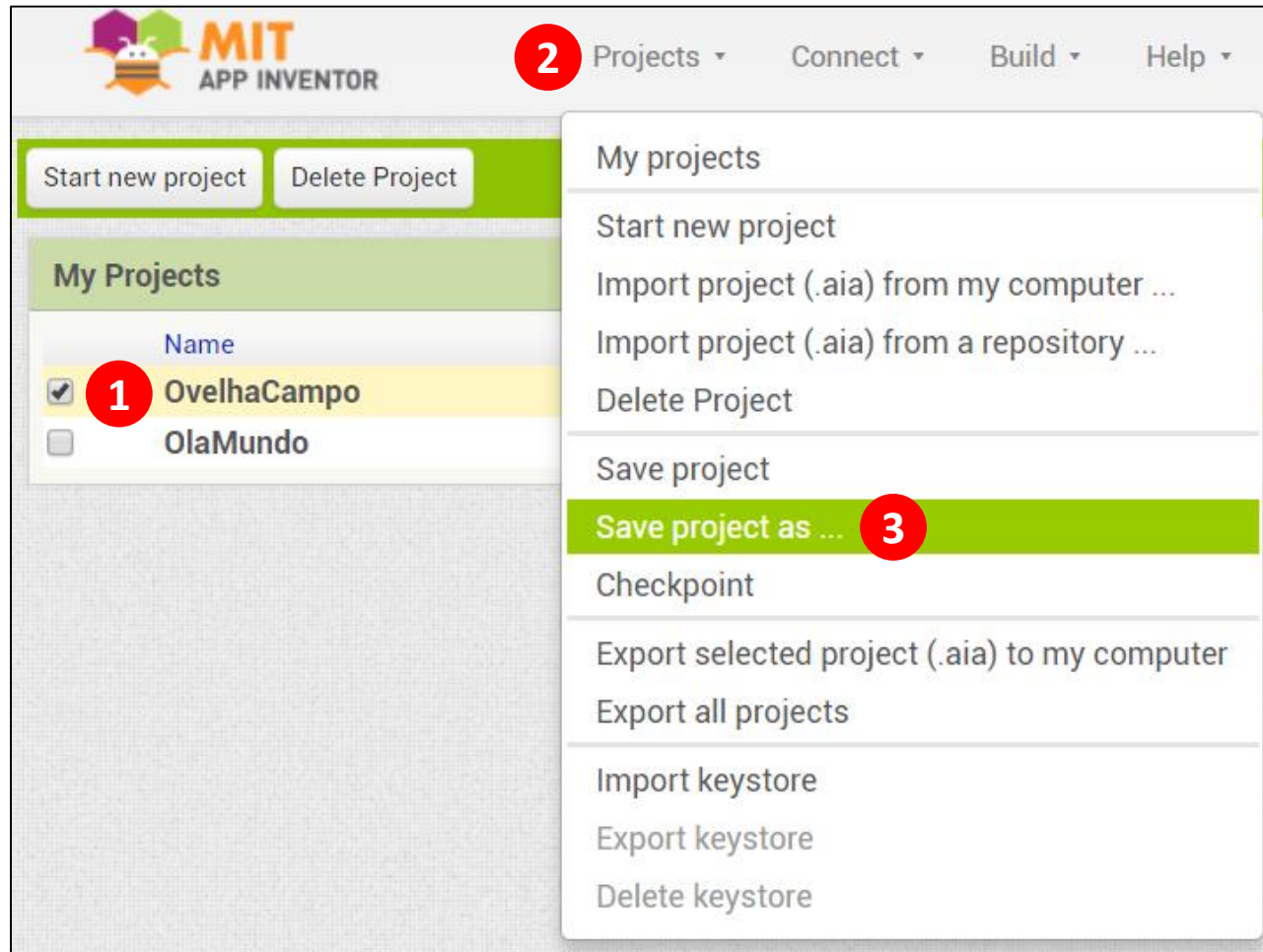
Crie uma copia da versão anterior para realizar as modificações.



Funcionamento Esperado do App

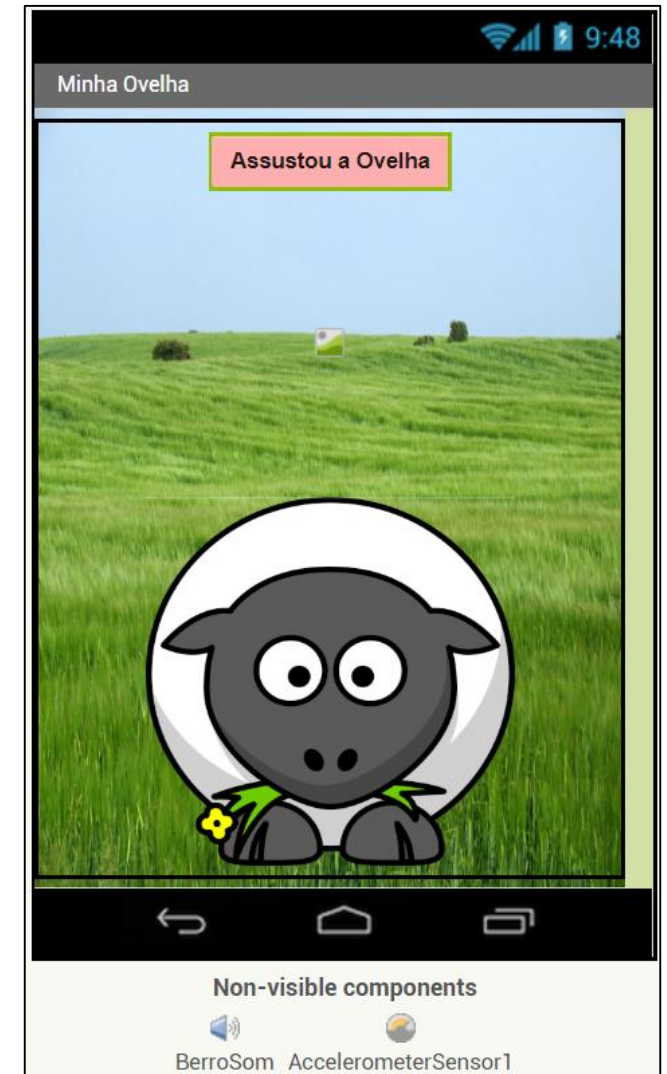


Passo 1: Salve um Novo Projeto





Passo 2: Adicione os Novos Componentes

- ❑ **VerticalArrangement:** permite organizar a aparência dos componentes um embaixo do outro;
- ❑ **Button:** servirá como um botão de reset para voltar a aparecer a ovelha quando o usuário clicar;
- ❑ **Image:** servirá apenas como um espaçador entre o botão de reset e o botão da ovelha;
- ❑ **AccelerometerSensor:** identifica movimentos no celular, como por exemplo: um balanço.



Passo 3: Ajustar as Propriedades

Layout

 VerticalArrangement 



AlignHorizontal
Center : 3 ▾


BackgroundColor
☐ None

Height
Fill parent...

Width
Fill parent...

User Interface

 Button 



BackgroundColor
 Pink

FontBold
☒

Text
Assustou a Ovelha

Visible
☐

User Interface

 Image 

Height
150 pixels...

Não precisa ser feita nenhuma configuração adicional no sensor do acelerômetro!

Passo 4: Programação dos Blocos

(Continua)

- ❑ Será necessário mais **dois blocos** para tratar dos eventos dos componentes do **Acelerômetro** e do **Botão de Reset**;
- ❑ Os *event handlers* desses componentes são respectivamente: **AccelerometerSensor.Shaking** e **Button.Click**;
- ❑ A propriedade de **visibilidade dos botões** irá **variar** dependendo do **evento acontecido**. Veja:

Evento	BotaoOvelha	BotaoReset
AccelerometerSensor1.Shaking	Escondido	Visível
BotaoReset.Click	Visível	Escondido

Passo 4: Programação dos Blocos

Usuário balançou o celular?

```
when AccelerometerSensor1 .Shaking
do
  set BotaoOvelha . Visible to false
  set BotaoReset . Visible to true
```

Os valores lógicos estão disponível em: **Built-In** → **Logic**.

Os blocos que **setam** propriedades são verdes!

Usuário clicou no botão de Reset?

```
when BotaoReset .Click
do
  set BotaoReset . Visible to false
  set BotaoOvelha . Visible to true
```

Também pode **digitar** o valor na área de trabalho e teclar enter!

Agora é a sua Vez!

Faça uma **adaptação do exercício anterior**.

Ao invés de esconder a ovelha, troque a imagem por uma **ovelha zangada**. Troque o texto do botão de reset para: “**Você deixou ela zangada!**”.



Dúvidas?





Programação para Dispositivos Móveis

Curso de Ciência da Computação

Universidade Paulista (UNIP)

CRIANDO A INTERFACE DO USUÁRIO

Todos os Créditos para Nathan Cirillo e Silva
nathan.silva@docente.unip.br

 **AULA 2**

