



# Unidade 5: Desenho de interface de usuário com o AppInventor

## Índice

1. Introdução.....	2
2. Modificação da Fonte de Letra.....	4
3. Uso de Imagens.. ..	7
4. Alterar Fundo da Tela.....	11
5. Botões Personalizados.....	13
6. Organização de Elementos em Tela.....	13
7. Alterar o Ícone do Aplicativo.....	17
8. Melhorias no Desenho do Aplicativo Turismo.....	19

## Unidade 3. Desenho de interface de usuário com o App Inventor

### 1.Introdução

Neste ponto, vamos tentar mostrar a importância das interfaces nos aplicativos para celulares. Não apenas para aplicativos desenvolvidos no AppInventor, mas também para qualquer plataforma atual ou futura.

Os aplicativos para celulares e os sistemas operacionais sofreram uma drástica evolução desde os primeiros Nokia até os atuais sistemas Android, iOS ou Windows Phone. Originalmente os sistemas operacionais móveis eram complexos e não estavam voltados para o público geral ou às pessoas alheias à tecnologia. Isto fez com que os aplicativos fossem também complexos, cheios de botões, nomes técnicos, menus, submenus e opções. Hoje, contudo, os sistemas operacionais para celulares são tão simples que qualquer pessoa pode usar. Os aplicativos também sofreram a mesma evolução.

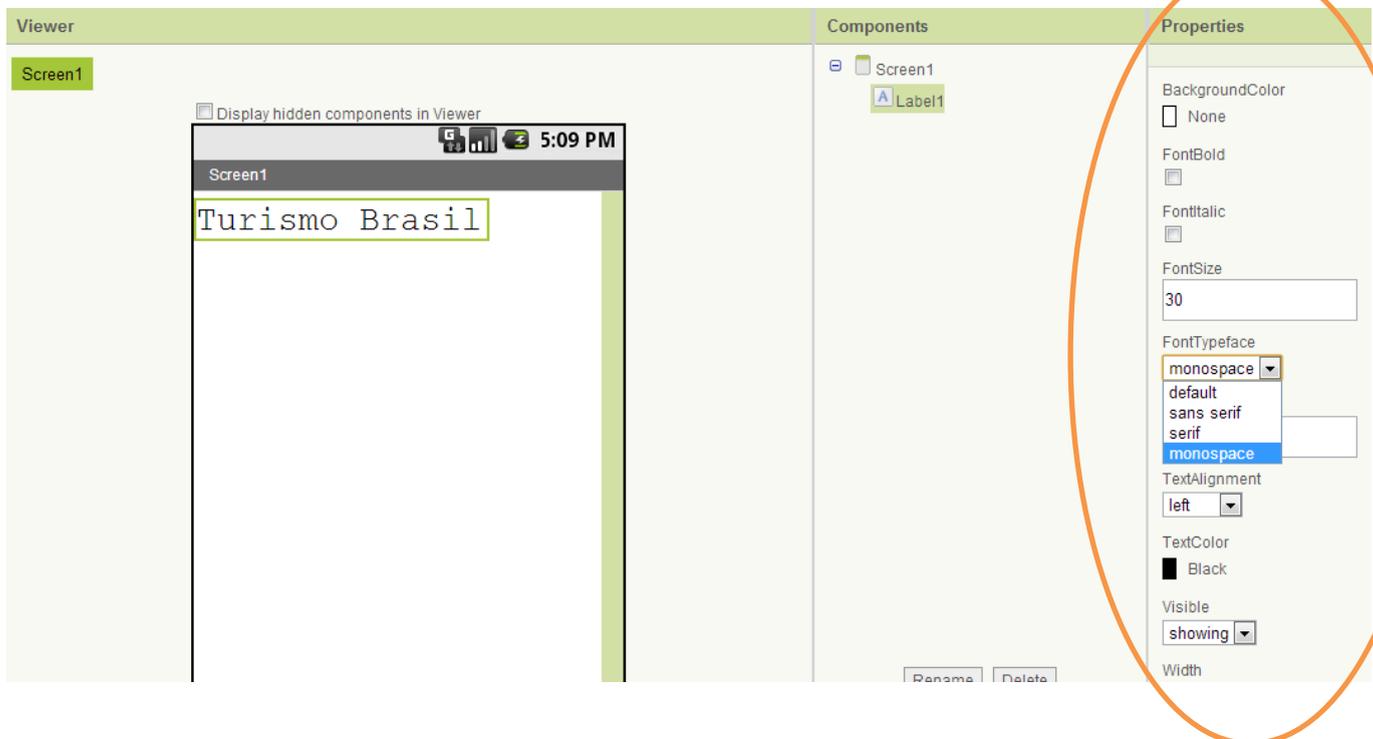
Atualmente podemos ver aplicativos muito complexos, mas com interfaces extremamente simplificadas. Agora o usuário não precisa saber o que o aplicativo faz para chegar ao resultado que quer. Um exemplo extremo é o famoso aplicativo “Facebook” com acesso pelo celular, ou ainda o Instagram.

Nesta unidade vamos conhecer alguns recursos para tornar nosso aplicativo mais bonito, mais agradável para o usuário. A tela que aparece para o usuário chamamos de interface de usuário é o que desenharemos neste estudo.

## 2. Modificação da fonte de letra

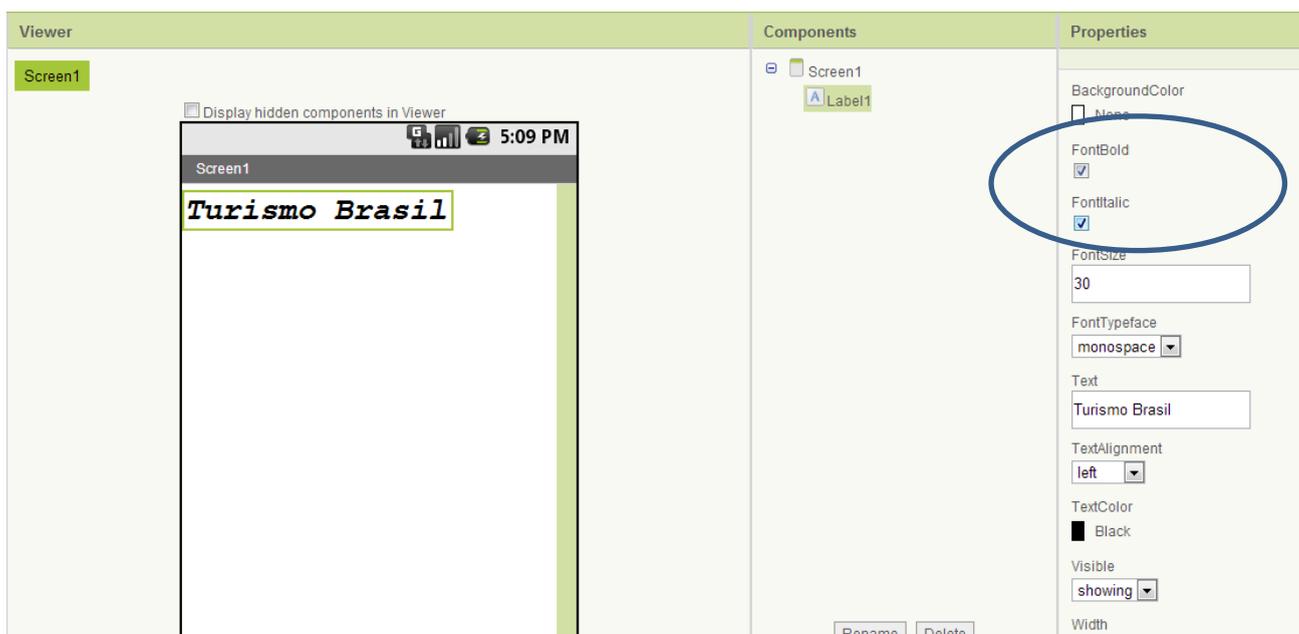
Uma forma simples de alterar o aspecto de nosso aplicativo é modificar o estilo da fonte. Ou seja, alterar tanto o tipo quanto o tamanho ou a cor da letra que utilizamos. Isto nos ajudará a personalizar nosso aplicativo de acordo com os nossos objetivos.

Para ver como alterar o tipo de letra, vamos criar um novo projeto, que chamaremos ProjetoBrasil. Incluímos um Label que chamaremos “Turismo Brasil”. Depois clicamos na Label e em Properties (itemFontTypeface), das outras fontes disponíveis. Por exemplo, selecionamos “Monospace”. Você também pode testar os outros tipos e ver o resultado. Outra opção disponível é alterar o tamanho da fonte. Por exemplo, selecionamos o tamanho “16”. Tente aumentar mais o tamanho de letra, por exemplo, a 30 para ver como fica. Parece que ficou um pouco melhor, não é?



Depois fazemos o contrário, diminuimos a, por exemplo, 14 para ver como fica. Você deve encontrar o tipo e tamanho de letra adequado em cada caso. Por exemplo, se estamos definindo o tipo de letra do título, o tamanho deverá ser grande e legível. Pelo contrário, se estamos definindo a fonte para texto dentro do aplicativo, o tamanho deverá ser mais reduzido.

Além de alterar o tipo e o tamanho, também podemos modificar o estilo. Por exemplo, se queremos fazer com que o texto apareça em cursivo ou itálica, uma vez selecionada o Label que queremos modificar, clicamos nos campos chamados “FontBold” (“Fonte Negrito”) ou “FontItalic” (“Fonte Itálica”). Ao clicar nas caixas correspondentes, poderemos ver como o texto muda de estilo.



Por último, também podemos alterar a cor da fonte. Para isso, clicamos na caixa “TextColor” e selecionamos a cor para o texto. Podemos tentar selecionar várias cores para ver seu desenho. Por exemplo, o azul e vermelho.

Se em vez de alterar a cor do texto queremos alterar o fundo do texto, basta alterar a propriedade “BackgroundColor”.

Atenção! Ao selecionar as cores do texto e do fundo, temos que levar em conta o contraste entre ambas as cores para que possa ser visualizado sem problemas. Por exemplo, durante o dia algumas telas de celular não dão para ver muito bem. Se nosso aplicativo não tiver um contraste de cores adequado, será pouco usável.



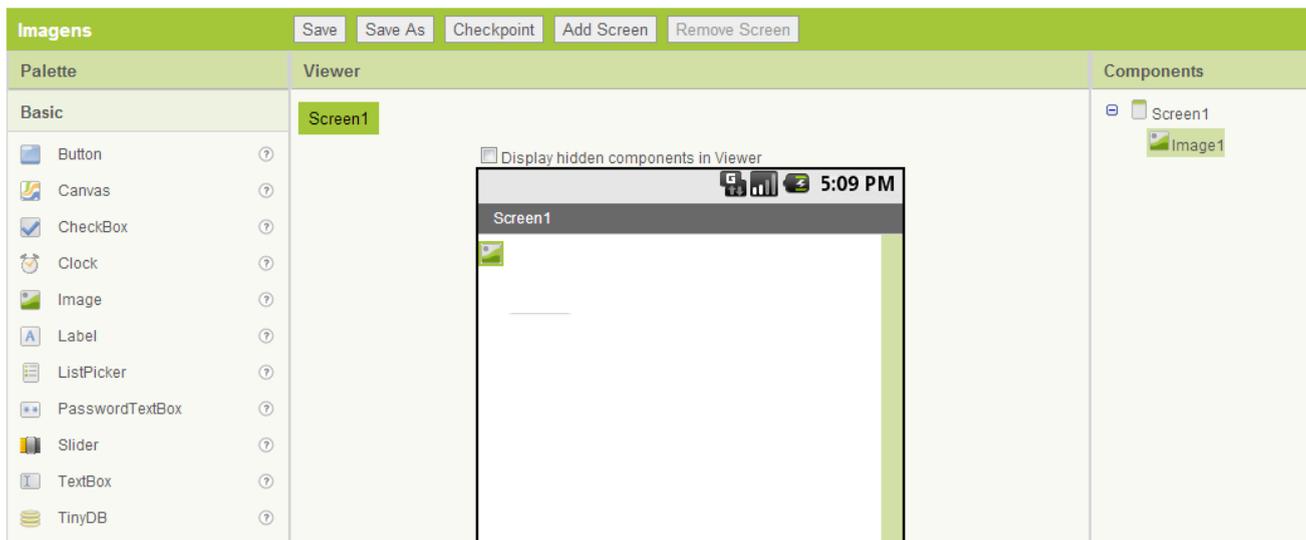
O uso adequado das fontes de letra pode fazer com que nosso aplicativo seja muito atraente para os usuários ou não. Para que seja assim, devemos prestar bastante atenção a seu desenho.

### 3. Uso de imagens

Outro aspecto fundamental na hora de melhorar o desenho de nosso aplicativo é incluir imagens, de forma que possamos transmitir de uma maneira visual o que queremos dizer.

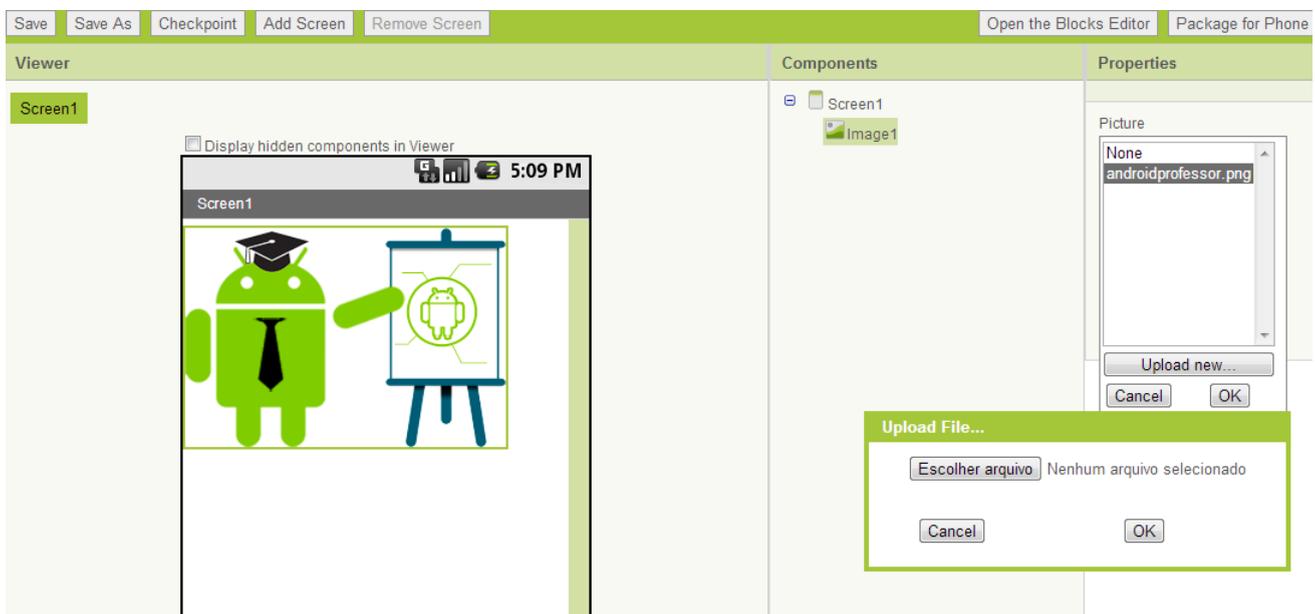
Para explicar como trabalhar com imagens no AppInventor, vamos criar um novo projeto, ao qual chamaremos “Imagens”.

Uma vez criado o projeto, procuramos na paleta de componentes “Image” e o arrastamos ao nossa Viewer. Este componente é o mais comum para trabalhar imagens no AppInventor já que nos permite carregar uma imagem diretamente e definir seu tamanho.



Depois carregaremos a imagem e vamos à parte inferior de “Componentes” no site do AppInventor e clicamos em “Add”. Escolhemos qualquer imagem de nosso PC.

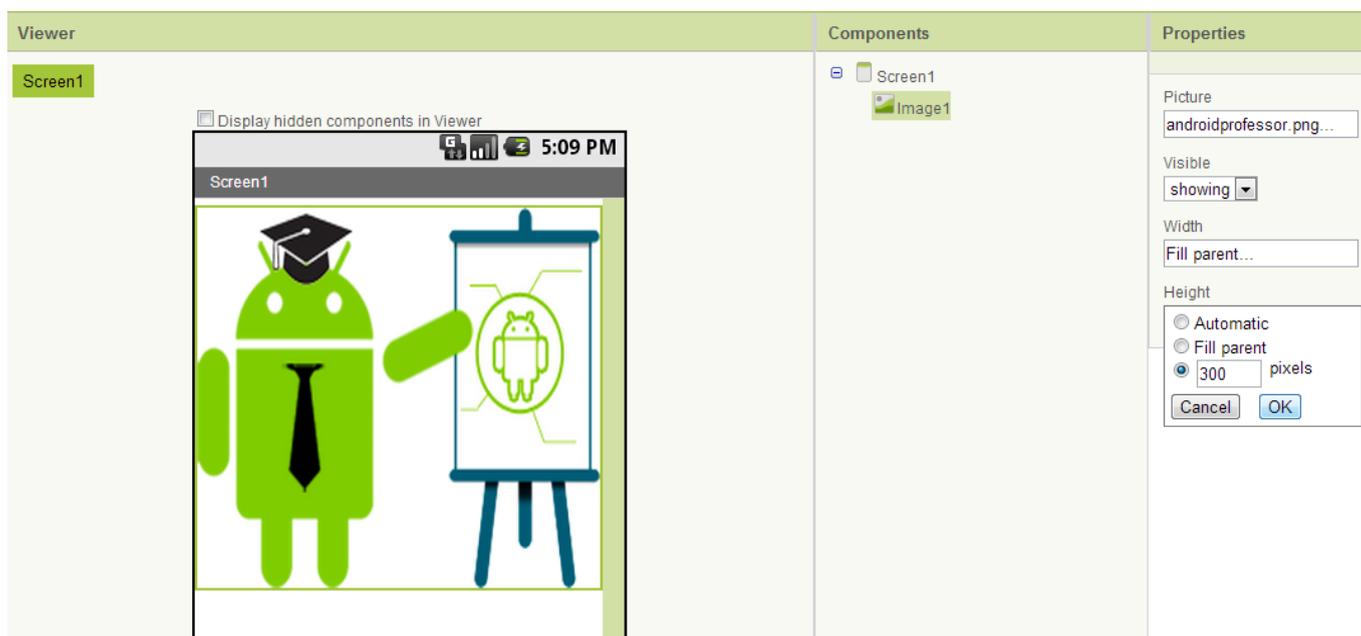
Agora, selecionamos o componente “Image” que arrastamos para a viewer. Em “Properties” vemos um item chamado “Picture” que serve para definir que imagem queremos mostrar. Clicamos nele e selecionamos a imagem que acabamos de carregar.



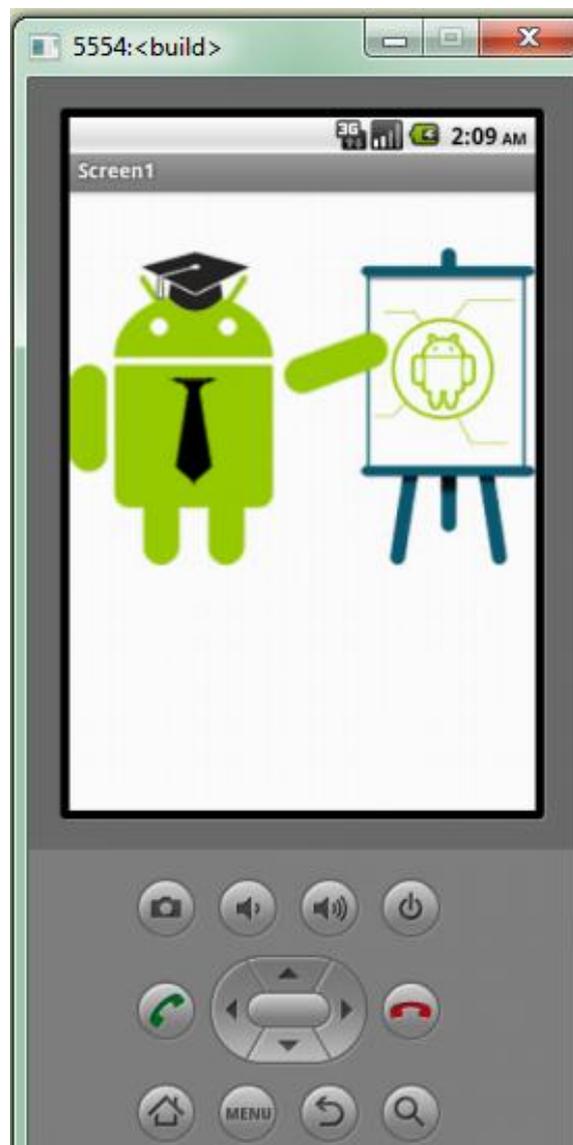
Como vemos, na tela da viewer, aparece nossa imagem.

Agora vamos alterar o tamanho da imagem. Vamos selecioná-la e vemos que em “Properties” temos os controles de tamanho Width(Largura) e Height (Altura). Clicamos na opção Width e vemos que há 3 opções: “Automatic”, ”FillParent” e um campo para introduzir o tamanho em pixels. O campo Automatic mostra a imagem em tamanho real. Em meu exemplo, as dimensões são: 266 x 183.

O campo FillParent, faz com que o tamanho do componente “Image” seja o mesmo que o do componente que o contém. Se selecionarmos a opção “FillParent”, aparecerá o componente do mesmo tamanho da tela. Também podemos definir o tamanho exato de nossa imagem. Por exemplo, nossa imagem é de 100x100, mas se quisermos que tenha o tamanho de 200x200 só temos que modificar o tamanho nos campos Height e Width. O único que temos que levar em conta é que o novo tamanho mantenha a proporção da imagem original. Isto é, se a imagem é de 100x100 não é possível deformá-la estabelecendo como valores 300x50. A imagem sempre deve manter a proporção, por isso é possível estabelecer como tamanhos, por exemplo: 50x50, 200x200 ou 320x320. Repare o resultado abaixo e compare com a imagem anterior:



Como podemos observar, o tamanho do componente, na cor verde, ajusta-se à tela, enquanto a imagem parece inalterada. Contudo, no emulador o tamanho da imagem foi alterado. Quando trabalharmos o aspecto de desenho devemos trabalhar sempre com o emulador já que às vezes o que o site de desenho mostra não é o que finalmente é visto no emulador.



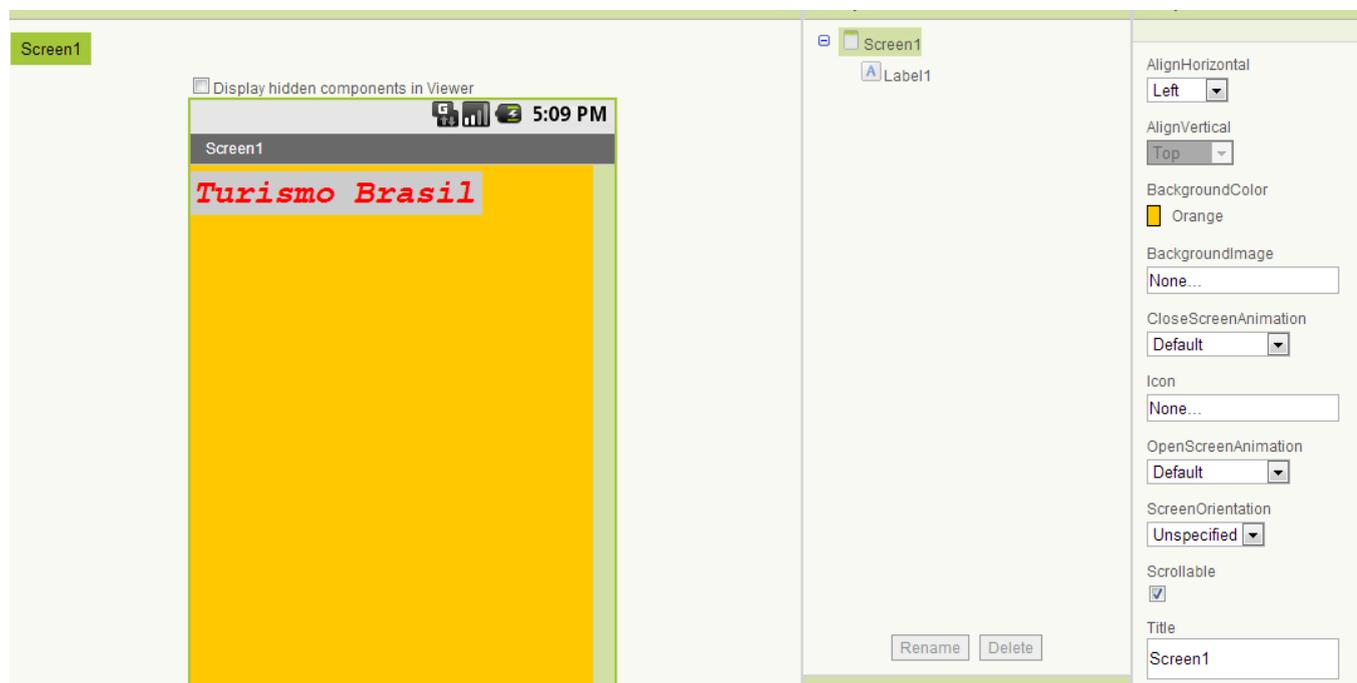
## 4. Alterar fundo de tela

Definir o fundo de tela do aplicativo é mais importante do que pode parecer. Com um fundo de tela adequado podemos realçar zonas do aplicativo, criar contrastes, em resumo, torná-lo mais atraente. Mas, corremos o risco de torná-lo menos humanizado se não utilizarmos bem as cores. Devemos utilizar cores que façam contraste entre si para diferenciar uma parte da outra. Ou seja, que o fundo faça contraste com os botões e elementos que temos. Se colocarmos um fundo cinza comuns botões cinzentos, o usuário não será capaz de visualizá-los.

Vamos retornar ao nosso ProjetoBrasil:

Clique em MyProjects e escolha ProjetoBrasil, clicando duas vezes sobre seu nome. Ele retornará a este projeto.

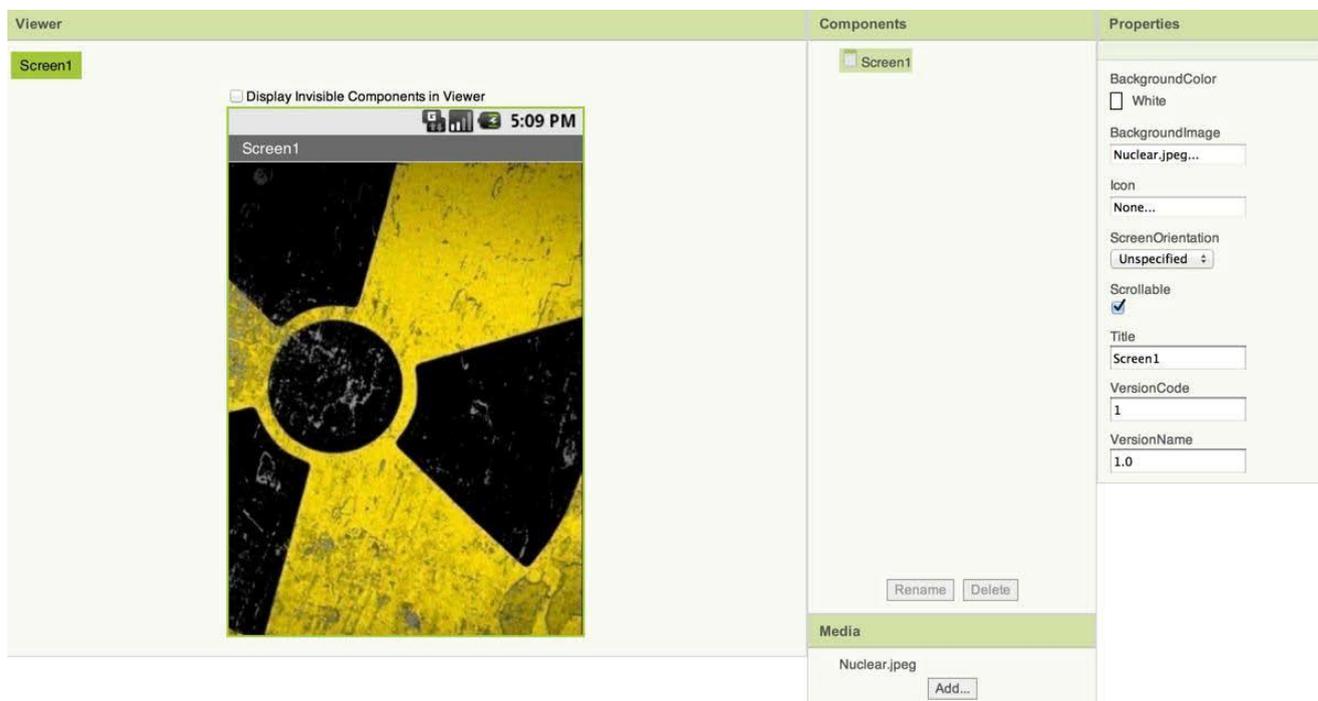
Para alterar o fundo de tela, podemos utilizar dois métodos. O primeiro é definir uma cor de fundo. Indo em Components, clicaremos em “Screen1”, e alteramos a propriedade “BackgroundColor”. Por padrão, a tela aparece em branco, mas podemos modificá-la como quisermos. Por exemplo, selecionamos a cor Orange (Laranja).



O segundo método consiste em atribuir uma imagem. Carregamos no nosso projeto uma imagem clicando em “Add”. Uma vez carregada, vamos a Properties e localizamos “BackgroundImage”.



Ao clicar, podemos selecionar a imagem, neste caso a que você deseja de seu computador.



O uso de imagens de fundo também dá grande personalidade ao aplicativo, oferecendo-lhe um estilo diferente. Contudo, as cores da imagem de fundo devem ser diferentes da cor do texto para que seja visualizado corretamente.

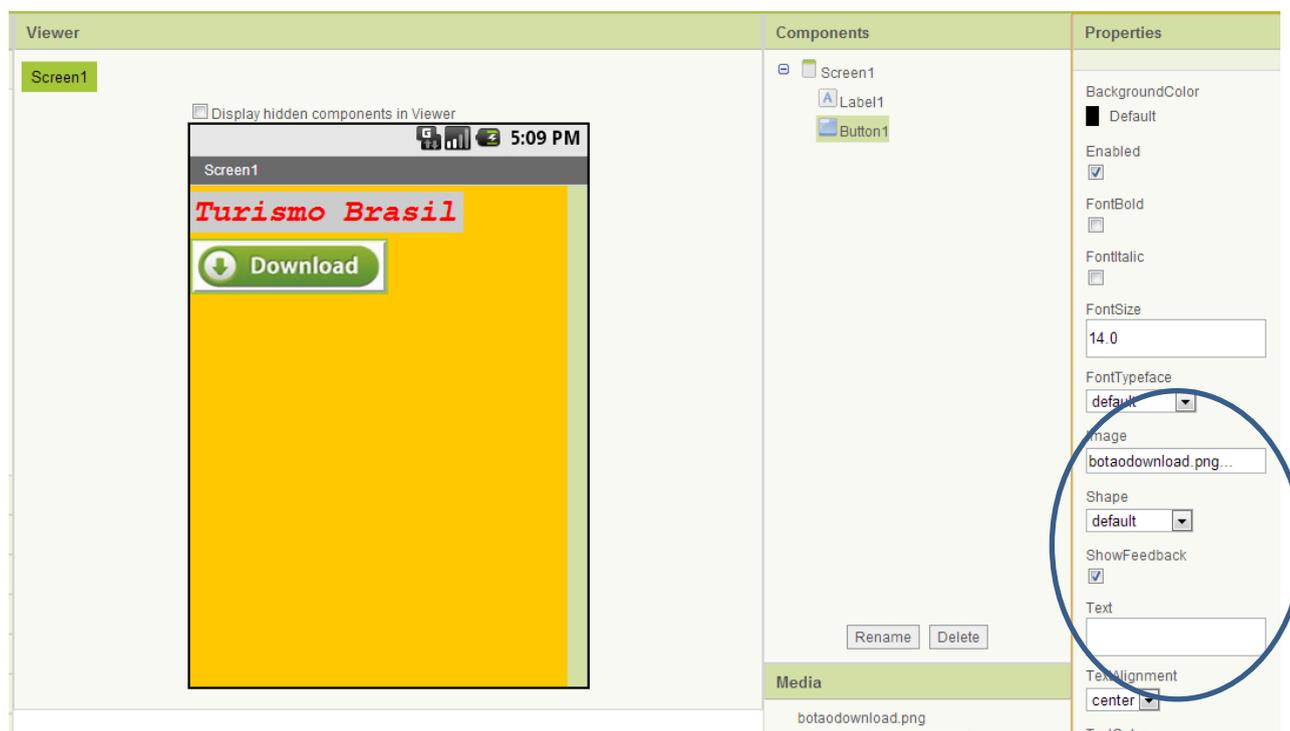
## 5. Botões personalizados

Outra opção para personalizar nossa interface é através dos botões. A maioria dos aplicativos do mercado não utiliza os botões que aparecem por padrão porque os botões personalizados dão aos aplicativos um toque mais atraente e personalizado.

Para criar botões personalizados, arrastamos o componente “Button” a Viewer e carregamos a imagem do botão personalizado, como fizemos antes.

Iremos aproveitar o projeto que já está aberto (ProjetoBrasil) para testarmos os botões personalizados. Insira um botão na área Viewer.

Clicando no botão, vemos em “Properties” ou item “Image”, aqui escolhemos nosso botão, de nome “botaodownload.jpg”. Lembre-se de apagar o texto que encontra-se em Text, esta caixa deve ficar vazia, em branco.



Para criar botões personalizados, devemos levar em conta certas regras de estilo para que o aplicativo tenha um aspecto profissional:

Em **primeiro lugar**, se a imagem que servirá de botão tiver título, este deve ser claro, em letras que contrastem com os outros botões e que seja visto facilmente.

Em **segundo lugar**, o botão deve contrastar com o fundo se deve distinguir perfeitamente sem que seja necessário procurá-lo na tela.

**Por último**, o botão deve ter o tamanho adequado para que qualquer usuário clique sem complicações. Alguns usuários têm os dedos largos, não faça botões muito pequenos.

## 6. A organização dos elementos na tela

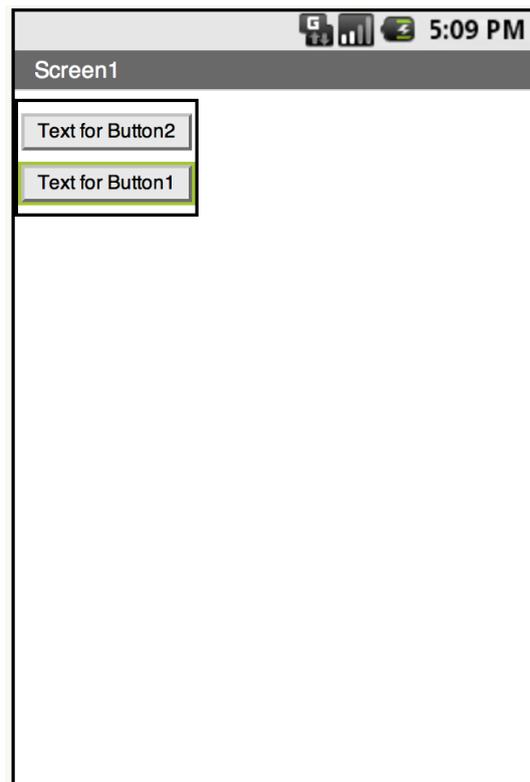
Um fator importante, quanto ao desenho, é a colocação dos elementos na tela. Isto é, vamos organizar os botões, as imagens e os outros componentes. Por padrão, no AppInventor os componentes que incluímos são colocados um debaixo do outro. Mas, em algumas ocasiões, podemos optar por outro tipo de organização. Para isso, o AppInventor dispõe de uma seção chamada “ScreenArrangement” (Organização em tela) que dispõe de três componentes para criar desenhos com a organização que desejarmos.

O primeiro destes componentes é o HorizontalArrangement, que serve para alinhar componentes um ao lado do outro, ou seja, na horizontal. Vamos utilizá-lo quando queiramos colocar, por exemplo, vários botões um ao lado do outro. Não há limite quanto ao número de componentes a adicionar, o único limite é a própria tela. Se você adicionar muitos, os que estiverem na parte direita não poderão ser vistos completamente.

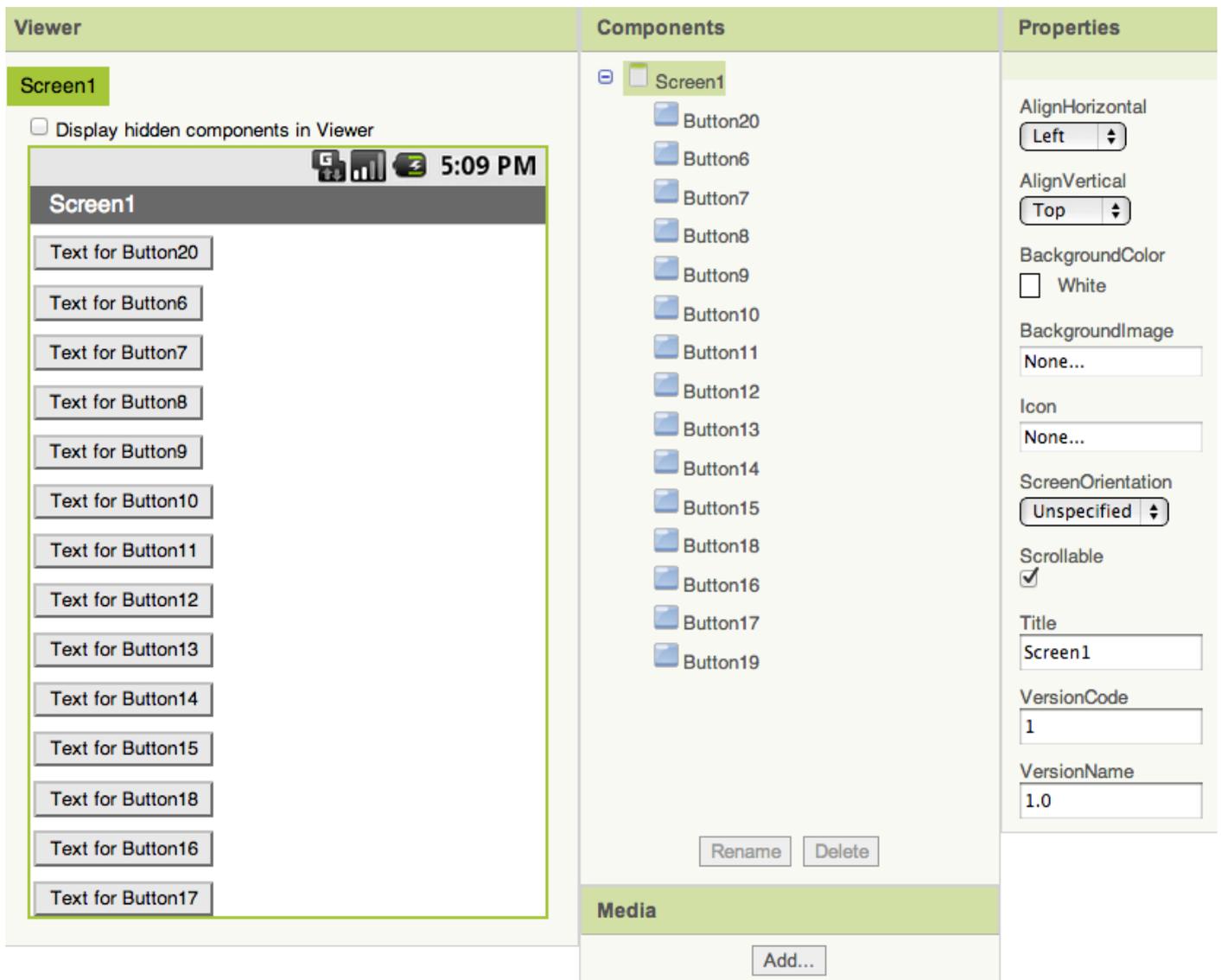
Nota: Para testarmos estes recursos, solicito que abra um novo projeto com o nome de Testealinhamento, e insira dois botões.



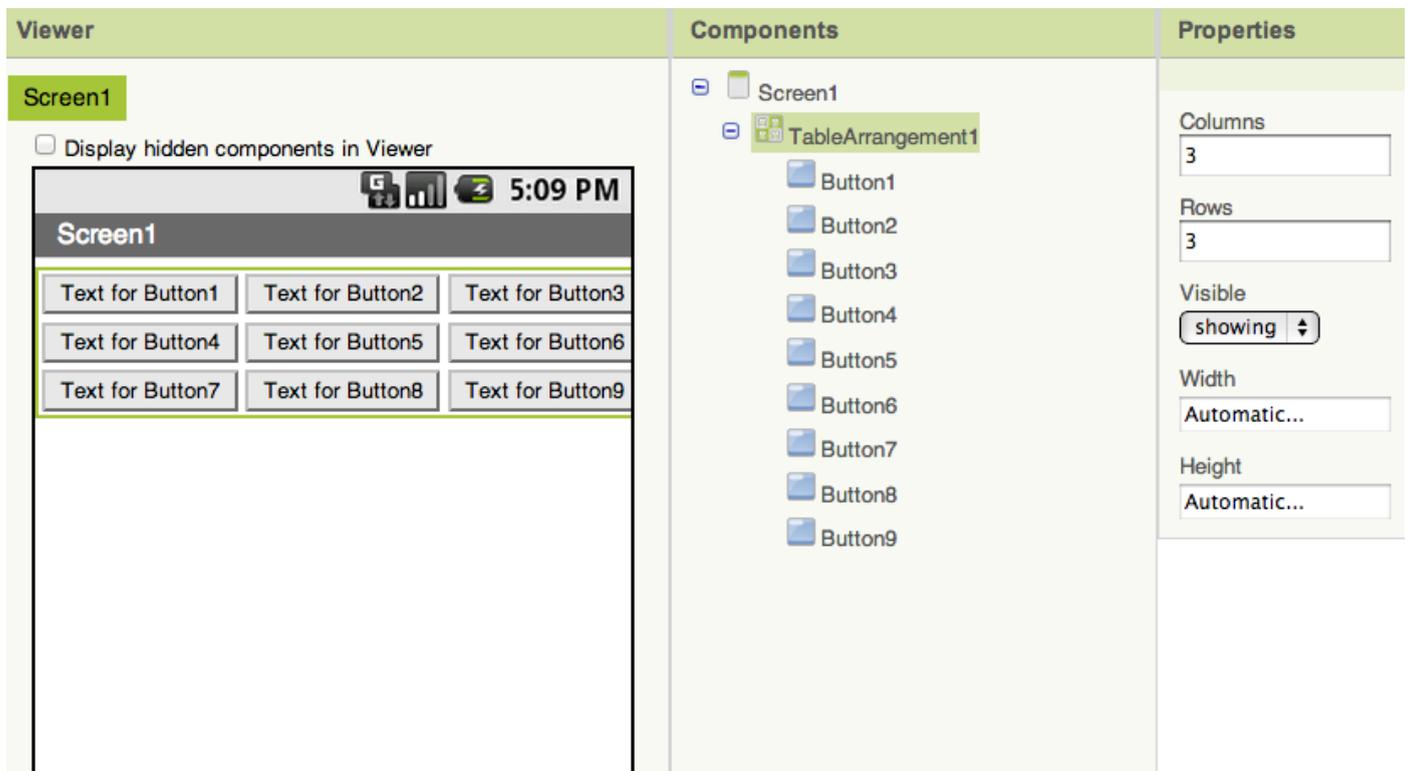
O próximo componente, é o VerticalArrangement. Serve para alinhar componentes um debaixo do outro, isto é, na vertical.



Também não há limite de componentes. O único condicionante é o tamanho da tela. Mas, podemos adicionar os elementos que quisermos mesmo que ultrapassem o tamanho da tela, se permitirmos a navegação Scroll. Ou seja, se permitimos que o usuário possa ir para cima ou para baixo para ver todos os componentes.



Por último, encontramos o componente TableArrangement que serve para alinhar componentes em forma de quadrícula. Tem a peculiaridade de que permite definir quantas colunas e filas terá graças às propriedades “Columns” e “Rows”. Por padrão, este componente vem definido com 2 filas e duas colunas. Mas é possível modificar estes valores definindo, por exemplo, 3 colunas e 3 filas.



## 7. Alterar o ícone do aplicativo

Para completar a formação sobre a criação de interfaces, falta apenas falar do ícone, algo essencial em qualquer aplicativo.

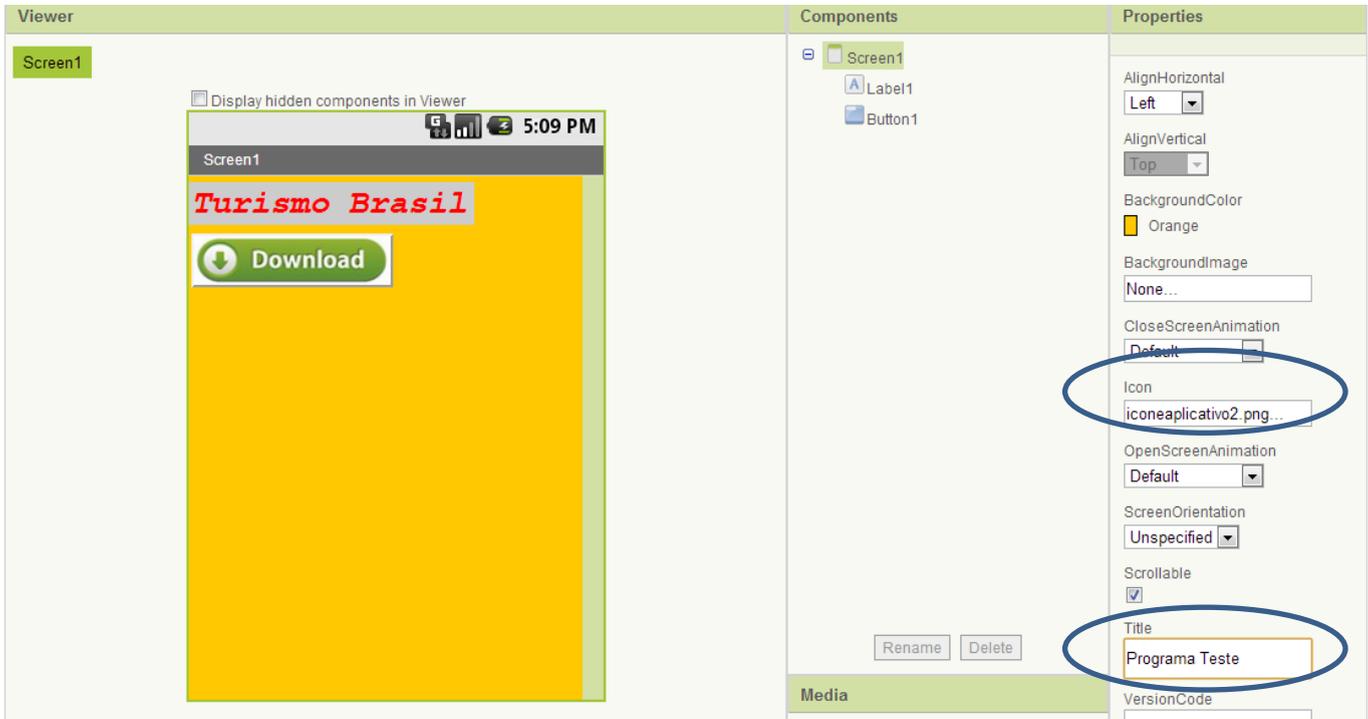
O ícone é tão importante que merece a pena investir tempo em criar um atraente. Deve representar o que o aplicativo faz e, além disso, deve ser desenhado pensando no público a quem vai dirigido.

Se o público-alvo é do ambiente empresarial, o ícone deve ser sóbrio e sério, mas se o público-alvo é adolescente, deve ser divertido e chamativo.

Os ícones no App Inventor devem ser imagens quadradas de 48 x 48 pixels.

Carregamos os ícones em nosso aplicativo como uma imagem qualquer, vamos a Properties e no item "Icon", introduzimos nosso ícone, chamado "íconeaplicativo.jpg".

Aproveite também para colocar um nome em seu aplicativo, escrevendo em Title (canto inferior direito), personalizando-o mais ainda.



## 8. Melhorias no desenho do aplicativo turístico.

Para finalizar, melhoraremos a interface do aplicativo turístico que fizemos na Unidade 2 incluindo funcionalidades. Se você se lembra, tínhamos uma interface simples. Abra o projeto para continuar trabalhando com ele.

o desenhar aplicativos, em primeiro lugar, devemos centralizar as ideias do aplicativo para que tenha uma funcionalidade concreta. Por exemplo, em nosso aplicativo turístico podemos mostrar como está evoluindo a Bolsa, mas, esta funcionalidade não é interessante para o usuário; não é o que está procurando. Nesse caso, é preferível fazer um aplicativo turístico e um aplicativo de Bolsa. O usuário deve encontrar conteúdos referentes ao nome do aplicativo.

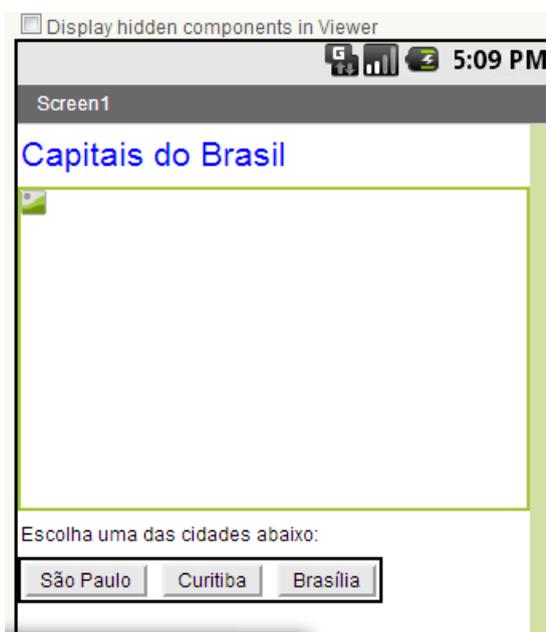
Como vemos, cada elemento aparece em uma linha, esteticamente não é atraente. Para ter os 3 botões agrupados em uma única linha, arrastamos um elemento `horizontalArrangement` da paleta para a `viewer`. Lembre-se que `HorizontalArrangement` está dentro de `ScreenArrangement` na coluna da esquerda.



Depois, arrastamos os botões da paleta para `horizontalArrangement` de tal forma que fique uma o lado do outro.



Agora vamos definir o tamanho da imagem dos monumentos. Queremos que a imagem ocupe de largura toda a tela para vê-la o maior possível. Alteramos o tamanho do componente “Image” para que ocupe toda a tela. Selecionamos o componente e introduzimos os parâmetros “Width” na opção “FillParent e “Height”, com 200 Pixels.”.



Obtemos, assim, uma interface mais organizada e clara. Contudo, deveríamos fazer com que ao entrar no aplicativo, “Image” tenha alguma imagem e não fique em branco.

Selecionamos o componente “Image” e em “Properties” clicamos no item “Image”. Por exemplo, uma imagem que aprecie que esteja em seu computador.

Podemos ainda, colocar um ícone para o aplicativo, um nome, e personalizar os botões, recursos simples que podem melhorar a aparência de seu aplicativo, e tornando-o bem diferente do projeto original.