CAPÍTULO 3

UMA SIMPLES CALCULADORA

Após ter realizado um tour de reconhecimento do ambiente do App Inventor, aprenderemos neste capítulo a declarar variáveis, utilizar botões para controlar a solicitação do usuário, realizar pequenos cálculos e exibir o resultado.

Passo a passo, você verá a explicação e a demonstração de como criar uma simples calculadora que realizará as quatro operações básicas da matemática – soma, subtração, multiplicação e divisão. Apesar de ser um aplicativo bem simples, servirá para demonstrar como utilizar algumas das seções do ambiente do App Inventor, apresentadas anteriormente, que usaremos nos próximos capítulos.

Iniciaremos criando um novo projeto. Para isso, clique no menu superior do App Inventor, na opção `Projects` , e depois em `Start new Project` . A figura a seguir exibe o local para a criação de um novo projeto. Sempre que desejarmos criar um novo projeto, será através deste caminho.

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

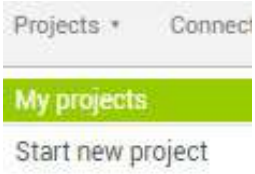

Figura 3.1: Criando um novo projeto

Após o clique, surgirá a tela `Create new App Inventor Project`, conforme a figura a seguir, onde daremos um nome para o projeto que será desenvolvido. Digite a palavra `Operacoes` em `Project Name` e, na sequência, confirme clicando no botão `OK`.

O nome `Operacoes` foi escrito sem acentos propositalmente, pois o App Inventor não permite a utilização de letras com acentuação, espaços em branco ou caracteres especiais.

> Baixe as imagens que serão utilizadas neste projeto, conforme informado no capítulo *Introdução*, em http://nelfabbri.com/appinventor/appinventor.zip.

Figura 3.2: Criando um novo projeto

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Surgirá na sua tela a `Screen1` , local no qual colocaremos os componentes visuais de nosso aplicativo. Esses componentes servirão para dar interatividade com o usuário, e aos poucos vamos discutindo sobre eles. A figura a seguir exibe a `Screen1` pronta para a inserção dos componentes.

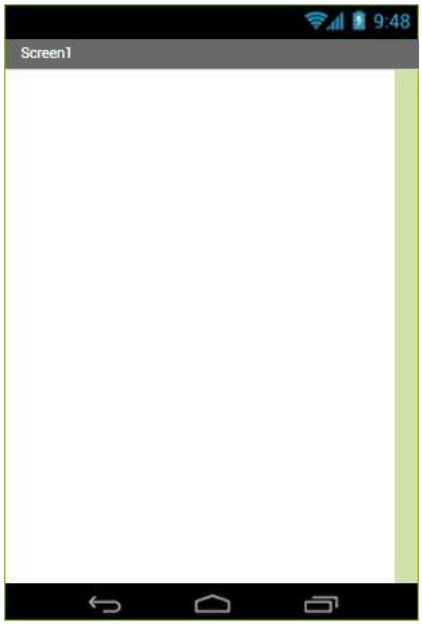

Figura 3.3: Tela do projeto em branco

3.1 DESENVOLVENDO O LAYOUT DO APLICATIVO

Antes de inserir as funções que darão interatividade ao nosso aplicativo, necessitamos desenvolver o seu layout.

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Quando usamos o termo *layout*, estamos nos referindo ao visual que nosso aplicativo terá, ou seja, onde ficarão disponibilizados os botões, as caixas de textos, as imagens e todos os outros componentes que poderemos utilizar na criação da aparência do app.

Começaremos a inserir uma imagem de fundo na `Screen1`. Necessitamos realizar um upload da imagem, ou seja, enviar a que desejamos exibir para o ambiente do App Inventor. Selecione a propriedade `BackgroundImage` na barra `Properties`, clicando no espaço imediatamente inferior à propriedade, para exibir a opção para enviar a sua imagem.

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

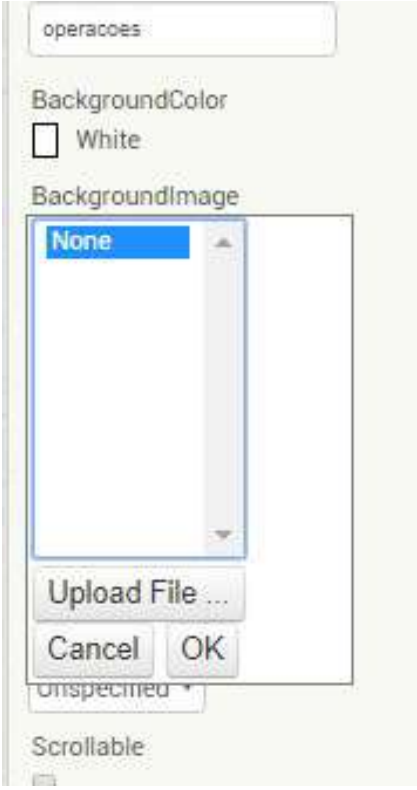

Figura 3.4: Propriedade BackgroundImage

Para enviar um arquivo, clique no botão `Upload File...` , assim será exibida a janela onde você escolherá a imagem para o envio. Utilize o arquivo `quadro.png` , que você já deve ter baixado, ou outra imagem qualquer de sua livre escolha. Sugiro usar imagens com o formato `.png` , pois este tem uma melhor qualidade de compressão de arquivos, sem perder a qualidade nem as suas cores.

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Figura 3.5: Selecionando o arquivo para upload

Abra o local onde você baixou os arquivos do projeto e escolha o `quadro.png` , então clique no botão `OK` para finalizar o envio. Aguarde o carregamento e, ao término, a sua `Screen1` deverá ter tal aparência:

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Figura 3.6: Screen1 após o envio da imagem

Vamos alterar algumas outras propriedades da `Screen1` . O nome que demos no momento da criação do projeto, `Operacoes` , não é necessariamente o que será exibido embaixo do ícone após a instalação no dispositivo. O nome de exibição pode ser alterado, e é exatamente isso que vamos fazer agora.

Localize na barra de `Properties` a opção `AppName` e digite: **Operações**. Nesse local, podemos inserir textos com acentuação, espaços e qualquer outro tipo de caractere.

Uma outra mudança que realizaremos na barra de propriedades da `Screen1` é o nome que será exibido na barra de título do aplicativo. Observe na imagem a seguir como está essa exibição antes da mudança.

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Figura 3.7: Barra de título original

Para alterar o nome da barra de título, vá à barra de `Properties` , localize a opção `Title` e altere o texto de `Screen1` para **As 4 operações**, pois este será o novo nome que exibiremos na barra de título de nosso app. A imagem a seguir exibe a tela de desenvolvimento com essa alteração.

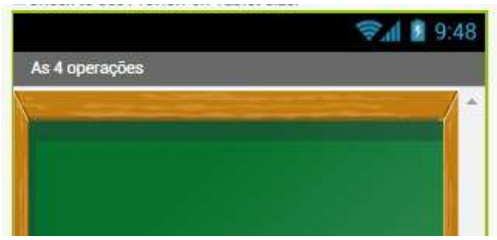

Figura 3.8: Barra de título modificada

DICA

Não é necessário modificar todas as propriedades de cada componente, alteraremos somente aquelas necessárias para cada aplicativo desenvolvido neste livro.

Dando sequência ao desenvolvimento do layout de nossa calculadora, inseriremos os componentes para montar o visual do

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

app. Vamos colocar uma mensagem na tela informando o que o aplicativo será capaz de realizar: **AS 4 OPERAÇÕES**.

> Ao inserir o primeiro componente, ele sempre será posicionado na parte superior esquerda da sua `Screen1` .

Para essa mensagem não se sobrepor a margem da lousa da imagem usada como fundo da `Screen1` , necessitamos de um componente para organizar a visualização da mensagem. Os componentes da guia `Layout` têm a finalidade de apenas organizar o designer dos demais componentes.

Vamos inserir um componente que realiza essa organização. Para isso, clique na guia `Layout` para exibir todos os componentes referentes à organização:

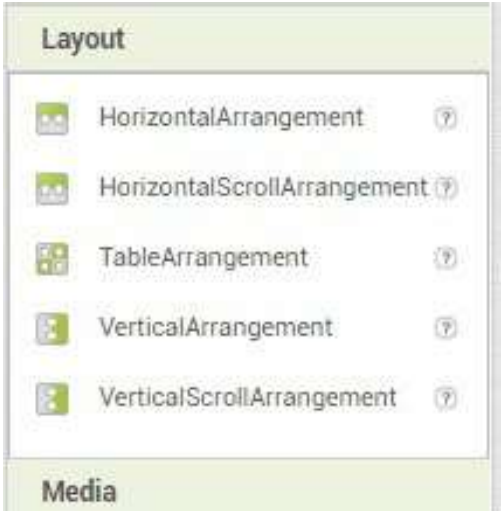

Figura 3.9: Guia Layout e seus componentes

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

O primeiro componente que usaremos é o `HorizontalArrangement`. Ele é capaz de organizar horizontalmente um ou mais componentes em seu interior. Sem ele, não conseguiríamos colocar dois componentes um ao lado do outro.

Para inserir um componente na `Screen1`, clique sobre ele, mantenha pressionado e arraste-o para a área da `Screen1`. Veja na figura a seguir como ela ficará após a inserção do `HorizontalArrangement`. Não se preocupe com o resultado deste componente, pois vamos alterar as suas propriedades e você verá como ele ficará.

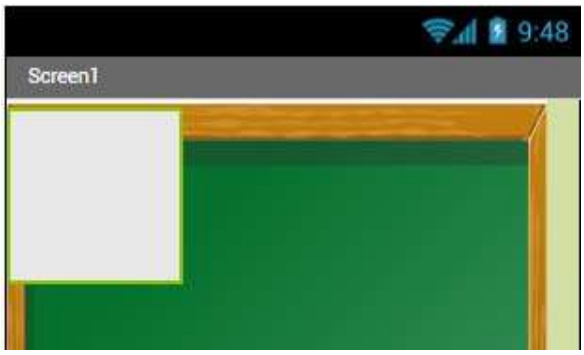

Figura 3.10: HorizontalArrangement

Altere as propriedades do componente `HorizontalArrangement`, conforme a tabela:

Propriedade	Alterar valor para	Observação
AlignVertical	Bottom:3	Indica que todos os componentes que estarão dentro do HorizontalArrangement serão alinhados na sua parte inferior.

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

BackgroundColor	None	Altera a cor de fundo do componente para transparente.
Height	80 pixels	Ajusta a altura do componente em exatos 80 pixels.
Width	Fill Parent	Ajusta o componente para a largura máxima da tela do dispositivo.

Utilizamos a propriedade `AlignVertical` para alinhar o texto na parte inferior da `HorizontalArrangement`, pois assim o texto que ainda vamos inserir ficará abaixo da margem da lousa que utilizamos de fundo. Alteramos para transparente a propriedade `BackgroundColor`, para que a imagem de fundo também seja exibida dentro do `HorizontalArrangement`.

Em `Height`, alteramos a altura do componente para `80 pixels` para forçar a exibição do texto abaixo da margem da lousa. E na propriedade `Width`, para o componente ocupar toda a largura da tela, selecionamos a opção `Fill Parent`.

Veja na figura a seguir como ficará a sua tela após a configuração das propriedades do `HorizontalArrangement`.

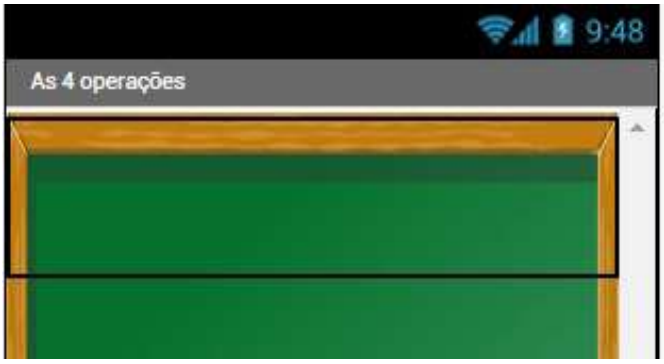

Figura 3.11: HorizontalArrangement configurado

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Dentro do `HorizontalArrangement`, queremos exibir a mensagem **As 4 operações**, que será o título do nosso aplicativo. Existe um componente específico para essa exibição, a `Label`. Vamos então inseri-la em nossa `Screen1`.

Clique na `Pallete`, e depois na opção `User Interface` para exibir todos os componentes com que o usuário poderá interagir. A figura a seguir exibe a guia `User Interface` expandida.

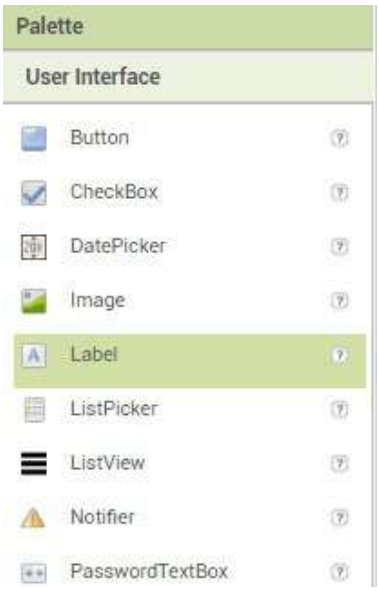

Figura 3.12: Guia User Interface

Selecione o componente `Label`, clique sobre ele e arraste para dentro do componente `HorizontalArrangement` na sua `Screen1`. A figura a seguir exibe como deverá ficar a sua tela de designer.

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

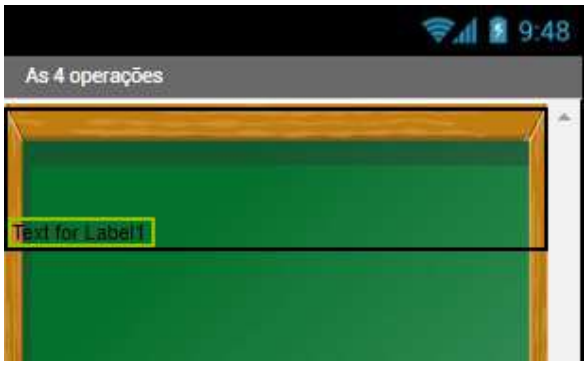

Figura 3.13: Tela com a Label inserida

Altere as propriedades do componente `Label` , conforme orientações da tabela:

Propriedade	Alterar valor para	Observação
FontBold	Marcar	Ativa a opção de negrito.
FontSize	25	Altera o tamanho da fonte.
Width	Fill Parent	Ajusta o componente para a largura máxima da tela do dispositivo.
Text	AS 4 OPERAÇÕES	Será o texto que será exibido na Label.
TextAlignment	center:1	Alinhamento do texto dentro da label.
TextColor	White	Cor de exibição do texto da label.

Como queremos dar um destaque ao texto **AS 4 OPERAÇÕES** que o leitor já deve ter digitado na propriedade `Text` , alteramos a que o deixa em negrito (`FontBold`), e também mudamos o tamanho da fonte através da propriedade `FontSize` , que foi modificada para o tamanho 25. Como desejamos que a `Label` ocupe todo o espaço da linha em que se encontra, alteramos a

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

propriedade `Width` para `Fill Parent` .

Para centralizar o texto em todo o espaço da `Label` , a propriedade `TextAlign` foi alterada para a opção `center:1` . E como a imagem de fundo do app é uma lousa verde, achamos interessante alterar a cor da `Label` para branco e, por essa razão, na propriedade `TextColor` , selecionamos `white` .

Após as configurações citadas, sua `Screen1` deverá ter a aparência:

Figura 3.14: Label configurada

Continuando o desenvolvimento do layout, necessitamos preparar o espaço para o usuário digitar um número. Como teremos na mesma linha uma mensagem indicando o que se deve digitar e uma caixa de texto para a digitação do número, obrigatoriamente teremos de inserir mais um `HorizontalArrangement` da guia `Layout` . Insira-o abaixo do `HorizontalArrangement` que contém a informação **AS 4 OPERAÇÕES** e vamos realizar as configurações conforme a tabela seguinte:

	Alterar	

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Propriedade	valor para	Observação
AlignHorizontal	Center:3	Indica que todos os componentes que estarão dentro do HorizontalArrangement serão alinhados ao centro.
AlignVertical	Center:2	Indica que todos os componentes que estarão dentro do HorizontalArrangement serão alinhados ao centro.
BackgroundColor	None	Altera a cor de fundo do componente para transparente.
Width	Fill Parent	Ajusta o componente para a largura máxima da tela do dispositivo.

Conforme vimos anteriormente, a propriedade `AlignHorizontal` é utilizada para alinhar os componentes em seu interior e, como agora queremos deixá-los centralizados, usamos a opção `Center:3`. A propriedade `AlignVertical` alinha os componentes de maneira vertical, e então selecionamos a opção `Center:2`.

Alteramos a propriedade `BackgroundColor` para transparente pelo mesmo motivo da anterior, ou seja, para que possa ser exibida a imagem de fundo também dentro do `HorizontalArrangement`. Com a propriedade `Width`, selecionamos a opção `Fill Parent` para que o componente ocupe toda a largura da tela.

O usuário deverá informar um número para o aplicativo poder realizar os cálculos. O objeto que recebe a digitação de valores é o componente `TextBox`. Volte para a guia `User Interface` da `Pallete` e insira uma `Label` e uma `TextBox` no interior da `HorizontalArrangement` que acabamos de configurar.

Vamos configurar as seguintes propriedades desta forma:

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Componente	Propriedade	Alterar valor para	Observação
Label	FontBold	Marcar	Ativa a opção de negrito.
	FontSize	16	Altera o tamanho da fonte.
	Text	Número	O texto que será exibido na Label.
	TextColor	White	Cor de exibição do texto da Label.
TextBox	Hint	Apagar o texto	Exibe um texto antes da digitação na TextBox.
	NumbersOnly	Marcar	Quando marcada, ativa o teclado numérico do seu dispositivo.

Para a `Label`, alteramos o texto de exibição para **Número**, pois assim o usuário do app saberá o que deverá digitar. Deixamos o texto em negrito e alteramos o tamanho da fonte para 16, utilizando respectivamente as propriedades `FontBold` e `FontSize`. Alteramos também a cor na propriedade `TextColor` para `White` para seguir com a mesma cor já utilizada anteriormente.

Para o componente `TextBox` que receberá o número digitado pelo usuário, apagamos os valores da propriedade `Hint`, pois não é necessária a exibição de nenhuma informação adicional, já que temos uma `Label` que indica o que deverá ser digitado em seu interior. Como o esperado na digitação é apenas números, devemos marcar a opção `NumbersOnly`, pois ela ativa um teclado que exibe apenas números durante a digitação, evitando assim que o usuário digite letras ou outros valores que não sejam números e que causariam erros na execução do app.

Veja na imagem a seguir a `Screen1` após as configurações realizadas na `Label` e na `TextBox`.

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Figura 3.15: Label e TextBox configuradas

Para facilitar a identificação dos componentes, vamos renomear os que usaremos durante a programação. Em vez de chamarmos de `Textbox1` , trocaremos o seu nome para `Txt_N1` , fazendo uma referência ao fato de essa `TextBox` representar o primeiro número.

Para renomear, clique sobre o nome `Textbox1` na área `Components` para selecioná-lo e, logo após, clique sobre o botão `Rename` . A figura a seguir exibe o botão para renomear um componente.

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

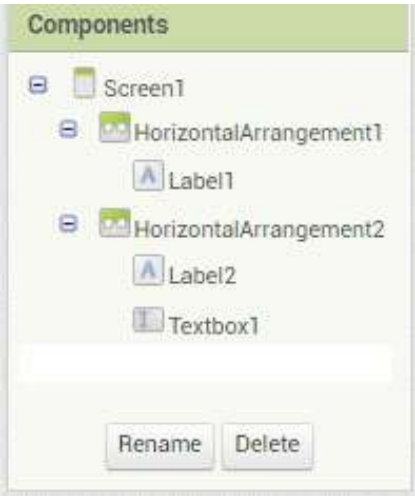

Figura 3.16: Área para renomear componentes

Após o clique, a janela `Rename Component` será exibida, conforme vemos a seguir. Altere a opção `New Name` para o novo nome que daremos ao componente `Txt_N1`, e clique no botão `OK` para confirmar.

Figura 3.17: Renomeando o componente

Repita o procedimento visto anteriormente, inserindo um novo `HorizontalArrangement`, uma `Label` e outra `TextBox`, para permitir a digitação de um segundo número, não se

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

esquecendo de renomear a sua `TextBox` para `Txt_N2` . A figura seguinte exibe como deverá ficar o layout de seu app após as configurações realizadas.

Figura 3.18: Layout com as duas TextBox

Para dar um espaçamento entre os componentes já inseridos e os botões das operações matemáticas que adicionaremos mais adiante, precisamos utilizar um novo `HorizontalArrangement` . Este deverá ser inserido abaixo do `HorizontalArrangement` que contém o segundo número para a digitação.

Como sua única função é realizar essa separação, ele deverá ficará vazio sem nenhum outro componente em seu interior. Porém teremos de alterar algumas de suas propriedades: `Height` para `40` pixels, pois assim será inserido um bom espaçamento para o layout do app. Na propriedade `Width` , altere para `Fill Parent` .

A figura a seguir demonstra como ficará sua tela até este momento:

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Figura 3.19: HorizontalArrangement servindo de espaçador

Precisamos de mais um `HorizontalArrangement` para organizar os quatro botões que realizarão as operações matemáticas de adição, subtração, multiplicação e divisão. Selecione na `Pallete` a guia `Layout` e arraste-a para a área de designer do aplicativo logo abaixo do `HorizontalArrangement` que utilizamos para realizar um espaçamento entre os componentes. Realize as seguintes alterações nas suas propriedades, conforme a tabela a seguir:

Propriedade	Alterar valor para
AlignHorizontal	Center:3
Width	Fill Parent

Como gostaríamos de centralizar nessa linha todos os futuros componentes que vamos inserir lado a lado em seu interior,

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

alteramos a propriedade `AlignHorizontal` para `Center:3` . Novamente na propriedade `Width` , foi selecionada a opção que deixa o componente utilizando todo o tamanho da linha: `Fill Parent` .

Acesse a `Pallete` e, na guia `User Interface` , selecione o componente `Button` e arraste-o para dentro do `HorizontalArrangement` . Esse primeiro componente `Button` , quando pressionado, executará as ações para realizar a adição dos números que vamos programar ainda neste capítulo.

Faça as seguintes configurações no botão, conforme a tabela:

Propriedade	Alterar valor para
FontSize	30
Height	50 pixels
Width	40 pixels
Text	+

Neste componente `Button` , foi alterado o tamanho da fonte para 30 na propriedade `FontSize` , para dar uma maior visibilidade ao comando que ele executará quando pressionado. Alteramos também o tamanho do botão para que fique com um layout mais proporcional ao restante do app. Para isso, na propriedade `Height` que altera a altura do botão, escrevemos `50` pixels e, para a sua largura (`Width`), escrevemos `40` pixels. E por fim, alteramos o texto de exibição do `Button` para `+` na propriedade `Text` .

Renomeie o botão para `Btn_Soma` conforme vimos anteriormente. Insira mais três botões ao lado direito do

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

`Btn_Soma` , ainda dentro da `HorizontalArrangement` , para realizar a subtração, multiplicação e divisão, e realize as configurações descritas na tabela a seguir:

Componente	Propriedade	Alterar valor para
Button2	FontSize	30
	Height	50 pixels
	Width	40 pixels
	Text	-
	Rename	Btn_Sub
Button3	FontSize	30
	Height	50 pixels
	Width	40 pixels
	Text	X
	Rename	Btn_Mult
Button4	FontSize	30
	Height	50 pixels
	Width	40 pixels
	Text	/
	Rename	Btn_Div

Da mesma maneira e com os mesmos objetivos que realizamos as configurações com o `Btn_Soma` , deveremos alterar as propriedades dos botões de **subtração**, **multiplicação** e **divisão** descritos na tabela anterior.

Veja na figura a seguir como ficará o design após a inserção e configuração dos botões.

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

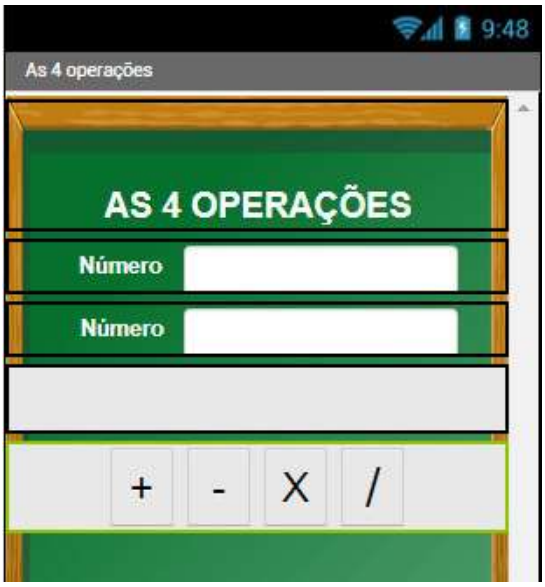

Figura 3.20: Exibição dos botões de cálculo

Finalmente, necessitamos de um espaço para exibirmos o resultado da conta escolhida pelo usuário. Como dissemos no começo deste capítulo, o componente responsável pela exibição de valores é o `Label` , porém, para a sua organização visual, devemos inserir mais um `HorizontalArrangement` logo abaixo do `HorizontalArrangement` que contém os botões com as operações matemáticas, e vamos alterar as suas propriedades, conforme a tabela a seguir.

Propriedade	Alterar valor para
AlignHorizontal	Center:3
AlignVertical	Center:2
Height	50 pixels
Width	Fill Parent

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Alteramos as propriedades `AlignHorizontal` e `AlignVertical` para `Center`, para deixar os componentes que ficarão em seu interior centralizados. Deixamos também, por uma questão de layout, a sua altura em `50` pixels, alterando a propriedade `Height`. Já na propriedade `Width`, alteramos a sua largura para o comprimento total, selecionando a opção `Fill Parent`.

Dentro deste último `HorizontalArrangement`, insira duas `Labels`, em que a primeira exibirá a mensagem `Resultado` e a segunda exibirá o resultado do cálculo obtido. Veja na tabela a seguir as propriedades que deveremos alterar.

Componente	Propriedade	Alterar valor para
Label4	FontSize	20
	Text	Resultado:
Label5	FontSize	20
	Text	0
	Rename	Lbl_Resultado

Na primeira `Label`, que exibirá a mensagem que foi alterada na propriedade `Text` para **Resultado**, alteramos o tamanho da sua fonte para 20. Ela é apenas informativa, ou seja, somente exibirá a palavra **Resultado**.

A segunda `Label` que foi inserida servirá para exibir o resultado do cálculo matemático. Alteramos também o tamanho da fonte para 20 em `FontSize`, e definimos a propriedade `Text` para o valor **0**, pois assim que o aplicativo for inicializado, o número **0** já estará sendo exibido na tela do app. Renomeamos essa `Label` para `Lbl_Resultado` para uma melhor identificação

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

desse componente no momento em que estivermos programando as funções de cada botão e quisermos exibir o resultado do cálculo.

A figura a seguir exibe a tela final do aplicativo:

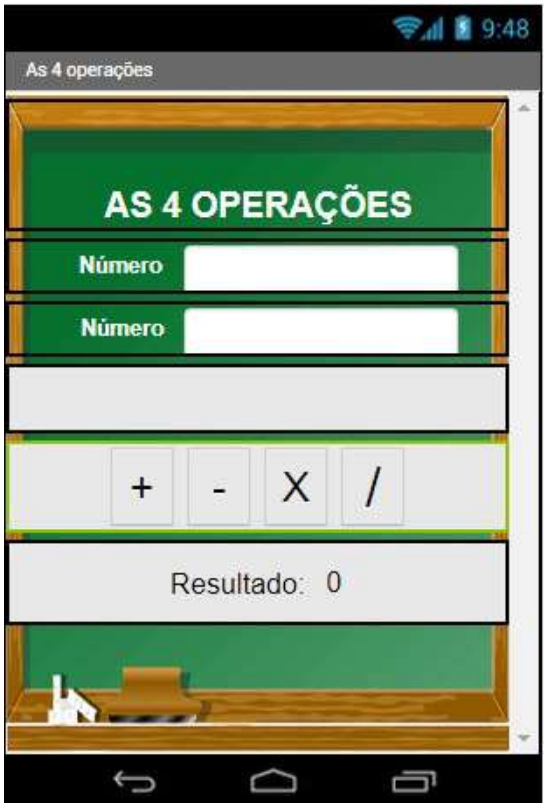

Figura 3.21: Layout completo do aplicativo

Nesse momento, finalizamos a criação do layout, já inserimos todos os componentes e configuramos as propriedades necessárias. Nosso próximo passo será preparar a interatividade do usuário com o aplicativo.

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

3.2 ACESSANDO A ÁREA DE PROGRAMAÇÃO

A área de programação, ou simplesmente a área dos blocos, é o local onde faremos a atribuição das funções para cada um dos componentes. Assim como existem várias propriedades e não modificamos todas, ocorrerá o mesmo com os componentes, pois vamos atribuir as funções, isto é, programar apenas os blocos necessários.

Para acessar a área de blocos, clique no botão `Blocks` no canto superior direito. A figura a seguir exibe o botão de acesso à área de blocos.

Figura 3.22: Acessando a área de blocos

Os campos `Txt_N1` e `Txt_N2` são os locais onde o usuário vai digitar os números para efetuarmos os cálculos. Logo, é necessário armazenar os valores digitados nessas `TextBoxs` em variáveis.

Começaremos a criar as variáveis que vamos usar no aplicativo. Precisaremos de duas variáveis para armazenar os valores digitados nas `TextBoxs` : `Txt_N1` e `Txt_N2` . Vamos conhecer qual é o procedimento para isso.

Do lado esquerdo da tela do App Inventor, existe uma guia chamada de `Blocks` , e nela temos uma seção chamada de `Built-in` , conforme demonstra a figura adiante. Dentro dela, existem algumas opções que vamos conhecer assim que surgir a necessidade. Mas para este momento, clique na opção `Variables`

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

para abrir uma outra guia com todas as opções das variáveis.

O que são as variáveis?

São espaços reservados na memória do computador para guardar informações que serão usadas em outros momentos durante a execução do aplicativo.

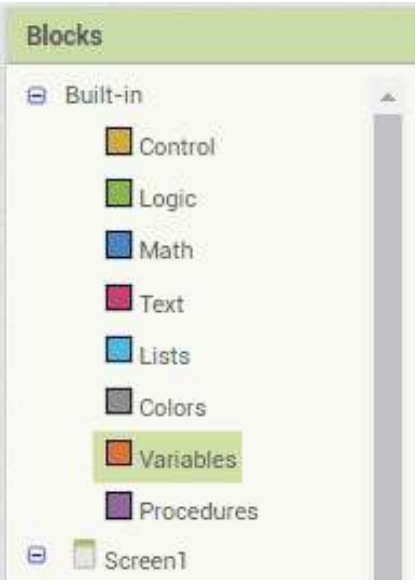

Figura 3.23: Guia Built-in

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Figura 3.24: Opção Variables selecionada

Como necessitamos criar duas variáveis, clique na opção `Initialize global name to` para ela ser inserida na área `Viewer`, e repita o processo para inserir um segundo bloco de variável. A figura a seguir exibe os dois blocos das variáveis:

Figura 3.25: Variáveis inseridas

Sempre que criar uma variável, torna-se necessário definir também o tipo de informação que ela vai armazenar. No exercício atual, vamos armazenar os **números** que o usuário digitará nas `TextBoxs`, por isso a nossa variável será do tipo **numérica**. Para isso, deveremos inserir um bloco com o valor `0` (zero) e encaixar

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

na sequência `Initialize global name to` .

> Variável do tipo numérica apenas armazena números. Encaixar um bloco com o valor `0` (zero) na criação da variável torna-a do tipo numérica.

Vá até a seção `Built-in` e verifique que existe uma opção chamada `Math` . Nela estão todas as opções matemáticas disponíveis no App Inventor:

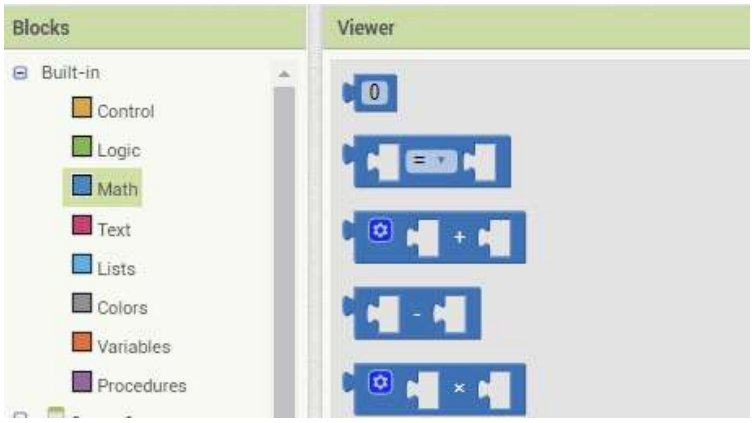

Figura 3.26: A guia Math

Como queremos criar variáveis numéricas para armazenar os valores digitados pelo usuário, necessitamos inserir um bloco que indique o **número zero**. Clique e arraste-o para encaixar na variável. Veja na próxima imagem como ela ficará. Repita esse procedimento para a criação de uma segunda variável numérica.

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Figura 3.27: Variável numérica

Toda variável necessita de um nome próprio que nos ajudará na sua identificação durante o restante do desenvolvimento do aplicativo. Vamos trocar os nomes para `numero1` e `numero2`, pois são bem melhores do que os nomes de identificação sugeridos (`name` e `name2`).

Para renomear uma variável, basta clicar sobre o nome que consta no bloco da variável `name` e digitar o novo (`numero1`). Repita esse procedimento para a segunda variável renomeando-a para `numero2`. A figura a seguir exibe as suas variáveis numéricas finalizadas.

Figura 3.28: Variáveis numéricas finalizadas

3.3 PROGRAMANDO UM BOTÃO

Para dar funcionalidade para cada um dos botões que inserimos na tela de designer, necessitamos utilizar um bloco que represente a ação que o usuário realizou. Essa ação é o clique que o botão receberá. Existe um bloco que representa essa ação, o `When Btn_Soma.click`.

Para localizá-lo, vá até a guia `Blocks` na lateral esquerda da tela e localize o nome que demos ao botão que realiza a soma, ou

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

seja, o `Btn_Soma` . Ao selecioná-lo, surgirá um complemento de tela ao lado direito com todas as possibilidades de ações que o botão poderá sofrer. A figura a seguir exibe a tela após a seleção do bloco `Btn_Soma` .

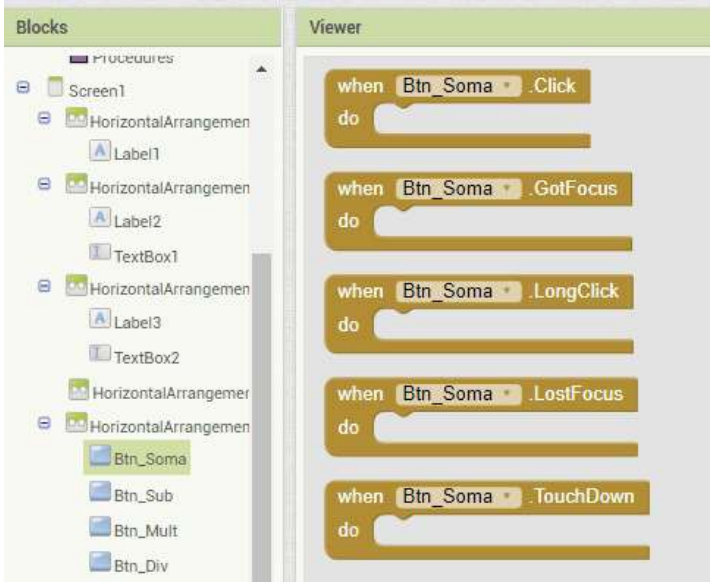

Figura 3.29: Opções do Btn_Soma

Como queremos programar as funções de soma, teremos de inserir um bloco que represente **quando o Btn_Soma for clicado**. Localize o bloco `When Btn_Soma.click` e arraste-o para a área `Viewer` . A próxima imagem exibe o `Btn_Soma` nessa área.

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

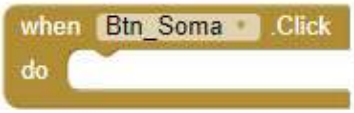

Figura 3.30: Btn_Soma

Agora, as variáveis que foram criadas e definidas como `numero1` e `numero2` deverão receber os valores digitados e contidos nas `TextBoxs` de nomes `Txt_N1` e `Txt_N2`.

Para uma variável receber um valor vindo de outro bloco, é preciso usar o comando `set to`. Acesse a área `Built-in` e, na seção `Variables`, clique sobre o bloco `set to` e coloque-o dentro do bloco `When Btn_Soma.click`, pois somente quando o clique no botão acontecer é que a variável receberá um valor. A figura a seguir exibe o bloco da variável dentro do botão.

Figura 3.31: Btn_Soma

Note que o local onde fica visível o nome da variável está vazio. Precisamos indicar o nome da variável que receberá o primeiro valor da `TextBox`. Para isso, clique no espaço logo após o comando `set` e selecione o nome `global numero1`:

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Figura 3.32: Selecionando o nome da variável

Após a seleção, a variável deverá ficar assim:

Figura 3.33: Variável selecionada

A variável `numero1` deverá receber o valor que foi digitado e está na `Txt_N1`. Deveremos inserir um bloco `Txt_N1.Text`, pois é nele que realmente se encontra o valor digitado. Localize na área de `Blocks` o componente `Txt_N1`. Clique sobre ele para expandir a tela com todos os seus possíveis blocos. A figura a seguir demonstra a expansão da janela.

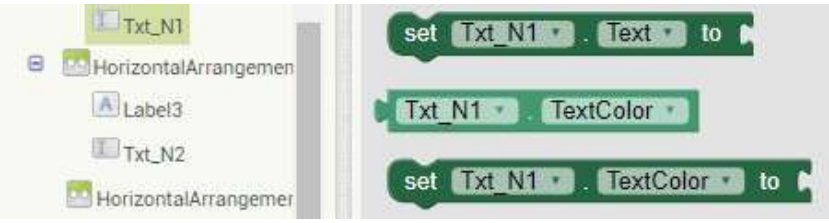

Figura 3.34: Opções de blocos da Txt_N1

Localize a janela em que surgiu o bloco `Txt_N1.Text`. Clique

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

e arraste-o para dentro do bloco da variável `set global numero1 to` . Veja esse procedimento na próxima figura.

Figura 3.35: Opções de blocos da Txt_N1

Repita o procedimento realizado com a variável `numero1` para a variável `numero2` . A figura exibe o bloco com as duas variáveis recebendo os valores:

Figura 3.36: Variáveis recebendo os valores digitados

Já temos os valores informados atribuídos às variáveis. Agora precisamos exibir o resultado da soma das variáveis em uma `Label` . O bloco de uma `Label` para exibição de valores é o `set Lbl.Resultado.Text to` . Localize na seção de blocos o componente `Lbl_Resultado` para expandir os possíveis blocos ao lado direito. A figura a seguir exibe os blocos do `Lbl_Resultado` .

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Figura 3.37: Opções de blocos do Lbl_Resultado

Selecione e arraste o bloco `set Lbl.Resultado.Text to` para dentro do botão `Btn_Soma` :

Figura 3.38: Opções de blocos do Lbl_Resultado

Deveremos realizar a soma antes da exibição. Como já vimos anteriormente, na guia `Built-in` existe a seção `Math` ; selecione-a para expandir e ver as possibilidades de operações matemáticas existentes. Lá você encontrará o bloco que realiza a adição:

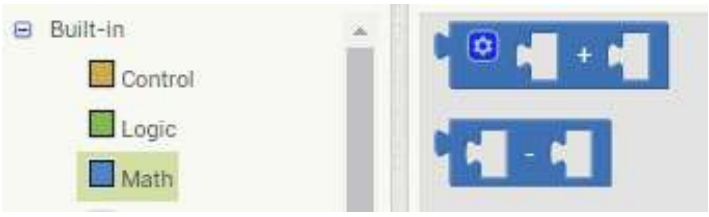

Figura 3.39: Opções de blocos do Lbl_Resultado

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Selecione o bloco de adição e encaixe-o logo após a `Label` que exibirá o resultado, pois o resultado da soma das duas variáveis será exibido na `Lbl_Resultado`. A figura a seguir exibe o bloco de adição encaixado no bloco de exibição do resultado.

Figura 3.40: Bloco de adição posicionado

Note que, dentro do bloco de adição, existem dois espaços em branco para encaixar os blocos das variáveis que serão somadas. Vamos selecionar uma variável para posicionar no primeiro espaço do bloco de adição.

Localize na área `Viewer` o bloco da criação da variável `numero1`. Para indicar a variável que desejamos, posicione o ponteiro do mouse sobre seu nome para ver as possibilidades que surgirão na tela. A figura a seguir exibe essas possibilidades.

Figura 3.41: Possibilidades das variáveis

- `get global numero1` — Indica que usaremos o valor que está armazenado na variável.
- `set global numero1 to` — Indica que estaremos

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

armazenando um novo valor na variável.

Selecione o bloco `get global numero1` e encaixe no primeiro espaço vazio do bloco de adição:

Figura 3.42: Posicionando a variável para somar

Repita o procedimento realizado com a variável `numero1` para a variável `numero2`, e encaixe-a no segundo espaço do bloco de adição. Ao término, o bloco de exibição da soma deverá estar igual ao exibido na figura:

Figura 3.43: Bloco da soma

A próxima figura exibe o bloco do botão `Btn_Soma` finalizado.

Figura 3.44: Btn_Soma finalizado

Desafio

Para os botões de subtração, multiplicação e divisão, o procedimento é basicamente idêntico ao realizado com o botão de soma. Será modificado apenas o bloco da guia `Math` para realizar a operação desejada. Seguindo as orientações deste capítulo, deixo

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

aqui um desafio para o leitor: complete os demais blocos com as operações faltantes.

3.4 TESTANDO O APLICATIVO

Após a criação de todos os botões, chegou a hora de testar seu app. Não se esqueça de primeiro clicar no aplicativo **aiStarter** para habilitá-lo. Como vimos no capítulo anterior, para emular seu app, clique no menu suspenso `Connect` do App Inventor e selecione a opção `Emulator`, então é só aguardar o carregamento do seu aplicativo para testá-lo.

Digite alguns valores nas caixas de texto e clique no botão desejado para fazer a operação matemática. No nosso exemplo, foram digitados os valores 5 e 8, e depois clicado no botão somar, apresentando na `Label` `Lbl_Resultado` a soma 13. A figura a seguir exibe a tela emulada com os valores digitados.

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

Figura 3.45: Teste do aplicativo

3.5 RESUMINDO

Neste capítulo, conhecemos e aprendemos sobre os componentes `Label` e `Button` e suas diversas propriedades de configuração. Nos blocos, aprendemos a declarar as variáveis, utilizar os blocos da guia `Math` e programar os botões para a exibição do resultado da operação matemática selecionada.

No próximo capítulo, vamos criar um aplicativo usando comandos de decisão, com duas telas que nos ajudarão a decidir qual combustível é mais vantajoso para abastecer o seu veículo:

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com

etanol ou gasolina?

E-book gerado especialmente para Carla Fabiane Calixto - carlacalixto16@gmail.com