



Desenvolvendo aplicativos Android facilmente com o MIT App Inventor

Dia 1. Conhecendo o ambiente e seus principais componentes

PROF. LEANDRO BORDIGNON

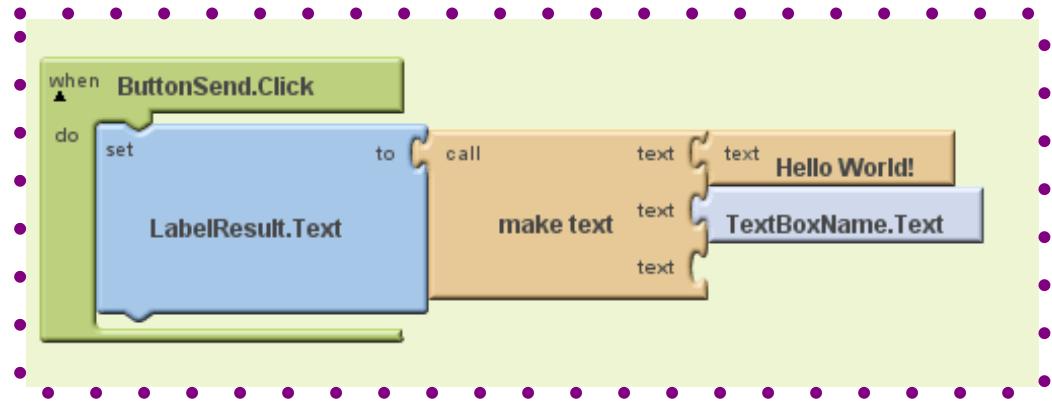


O que é o App Inventor ?





Um ambiente de desenvolvimento na Web, originalmente criado no Google Labs, mas atualmente pertence ao MIT Labs. Tem como objetivo facilitar o desenvolvimento de Aplicativos para Android.





O Desenvolvimento de aplicativos Android requer um conhecimento da plataforma Java, do paradigma orientado a Objetos. Tornando difícil o desenvolvimento de um App por não desenvolvedores.



O App Inventor permite
desenvolver aplicativos Android
sem o conhecimento destes
conceitos.





Por que é tão fácil?

- **Nenhuma sintaxe.** A linguagem de blocos elimina a necessidade de lembrar e digitar código
- **Tudo está bem na frente de vocês.** Componentes e funções estão em gavetas. Basta encontrar, arrastar e soltar.
- **Componentes de alto nível.** O app inventor tem uma grande biblioteca.
- **Plug-in.** Você não pode fazer coisas que não fazem sentido.
- **Concreto.** Menos abstrato do que muitas linguagens.
- **Na Web,** caso tenha um celular Android, não vai precisar instalar nada no seu computador.

O que podemos criar?



- Aplicativos educacionais
 - Com vídeos, imagens, texto-para-voz
- Aplicativos de localização
- Jogos
- Controladores de Robô
- Aplicativos para a Web
- Aplicativos pessoais



O que não podemos criar?

- App Inventor não é um substituto para a linguagem de programação de aplicativos para o Android.
- Não é uma ferramenta para criar jogos e aplicativos **comerciais**.
- Não é possível criar jogos como Angry Birds ou aplicativos complexos como o Firefox Mobile, e as aplicações criadas estão **longe de ser bonitas**. Elas são, no entanto, funcionais.

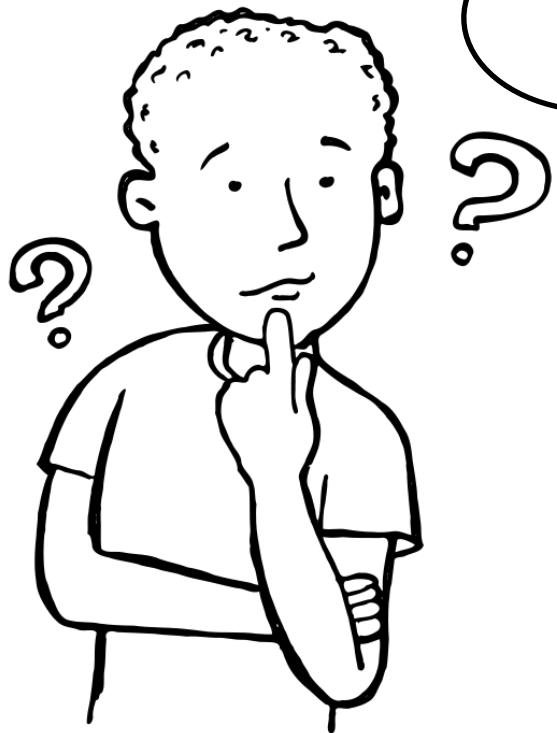
Até o final da aula de
hoje, seremos capazes
de fazer um app simples
que calcula o IMC



A screenshot of an Android application titled "IMC - App". The screen shows fields for "Altura" (Height) containing "1.68" and "Peso" (Weight) containing "64", which is highlighted with an orange border. A "Calcular" (Calculate) button is below the weight field, and the result "22,67574" is displayed below it. At the bottom is a numeric keyboard with digits from 0 to 9, symbols like @, #, \$, %, &, -, +, (,), =, \, <, *, ', :, ;, !, ?, and a checkmark button.



OK. E como começo ?



Primeiro projeto



1. Crie uma conta de e-mail do gmail (utilize o navegador Firefox ou Chrome).
2. Entre na sua conta do gmail.
3. Adicione uma nova aba no navegador.
4. Digite o seguinte endereço:
 - <http://appinventor.mit.edu/explore/>

Primeiro projeto



Clique em "Create"

The screenshot shows the MIT App Inventor website at appinventor.mit.edu/explore/. The main heading is "Apps in the World" with subtext "Creative ways people are using App Inventor" and categories "Education - Communities - Health - Environment - Entrepreneurship". A large blue button labeled "View Stories" is visible. To the right, there's a photograph of a person using a smartphone. Below the main heading, there are three sections: "Get Started" (with a green icon of a flag and a "Start" button), "Create" (with an orange icon of a smartphone and a "Create" button), and "Tutorials" (with a purple icon of a lightbulb and a "Tutorials" button). A red speech bubble points from the text "Clique em 'Create'" to the "Create" section.

appinventor.mit.edu/explore/

MIT App Inventor

Home ▾ Blog ▾ Support ▾ Create

Follow Us: [f](#) [t](#) [y](#) [g](#) [h](#) Google™ Custom Sea [q](#)

Apps in the World

Creative ways people are using App Inventor

Education - Communities - Health
Environment - Entrepreneurship

[View Stories](#)



Get Started

Follow these simple steps to build your first app.

[Start](#)

Create

Design and program your own apps using MIT App Inventor.

[Create](#)

Tutorials

Step-by-step guides show you how to build all kinds of apps.

[Tutorials](#)



Você pode acessar o App Inventor diretamente, navegando ao endereço <http://ai2.appinventor.mit.edu/>

Primeiro projeto



Seleciona
seu e-mail

Google accounts

The application **MIT AppInventor Version 2** is requesting permission to access your Google Account.

Please select an account that you would like to use.

- carlafaria@ifma.edu.br
- etec@ifma.edu.br
- carlaslz@gmail.com
- redeetecbrasil@ifma.edu.br

Google is not affiliated with the contents of **MIT AppInventor Version 2** or its owners. If you sign in, Google will share your email address with **MIT** other personal information.

[Allow](#) [No thanks](#) [Sign in to another account](#)

Clique em
“Allow”

Primeiro projeto



Welcome to App Inventor!

Welcome to MIT App Inventor 2

New Release (done)
May 26, 2014: DatePicker Incompatibility
[More Information](#)

Got an Android phone or tablet? Find out how to
[Set up and connect an Android device](#).

Don't have an Android device? Find out how to
[Set up and run the Android emulator](#).

(Emulator and USB connections are currently for Mac
and Windows only. Support for Linux is *coming
soon!*)

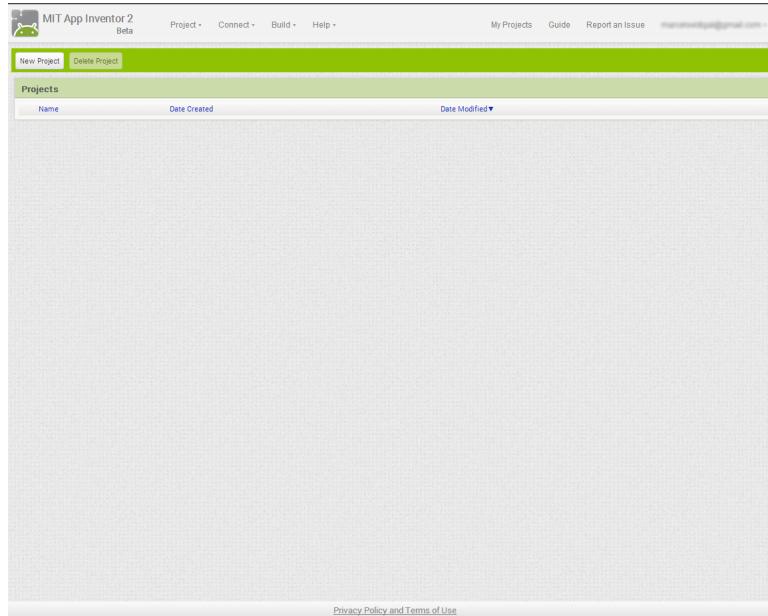
(Support for Internet Explorer is *coming soon!*)

[Continue](#)

Clique em
“Continue”

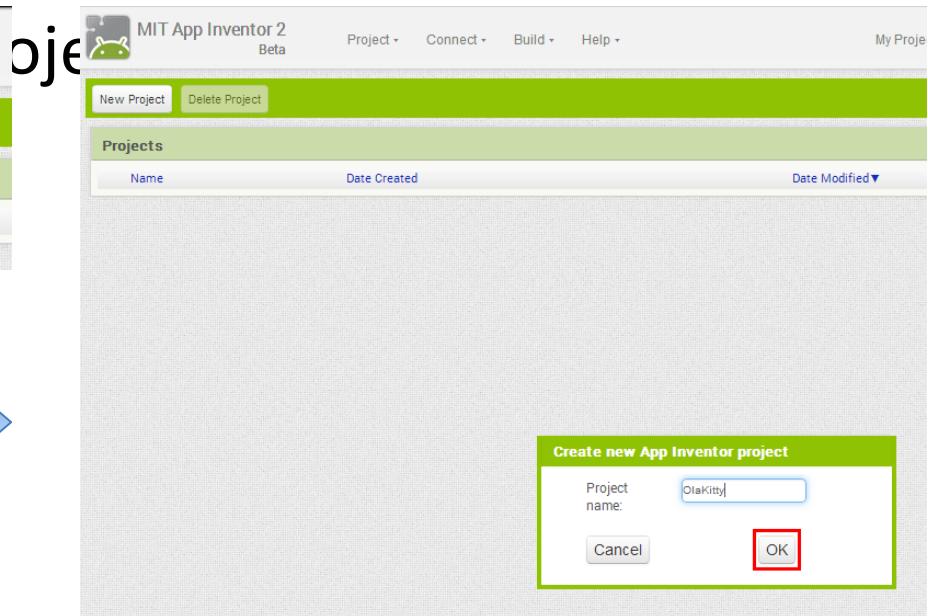
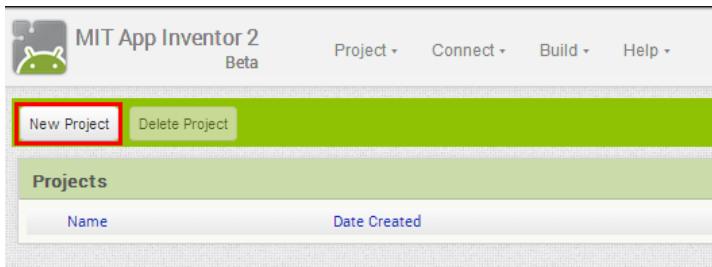
Primeiro projeto

Se for a primeira vez que usa o App Inventor,
você verá a página de Projetos



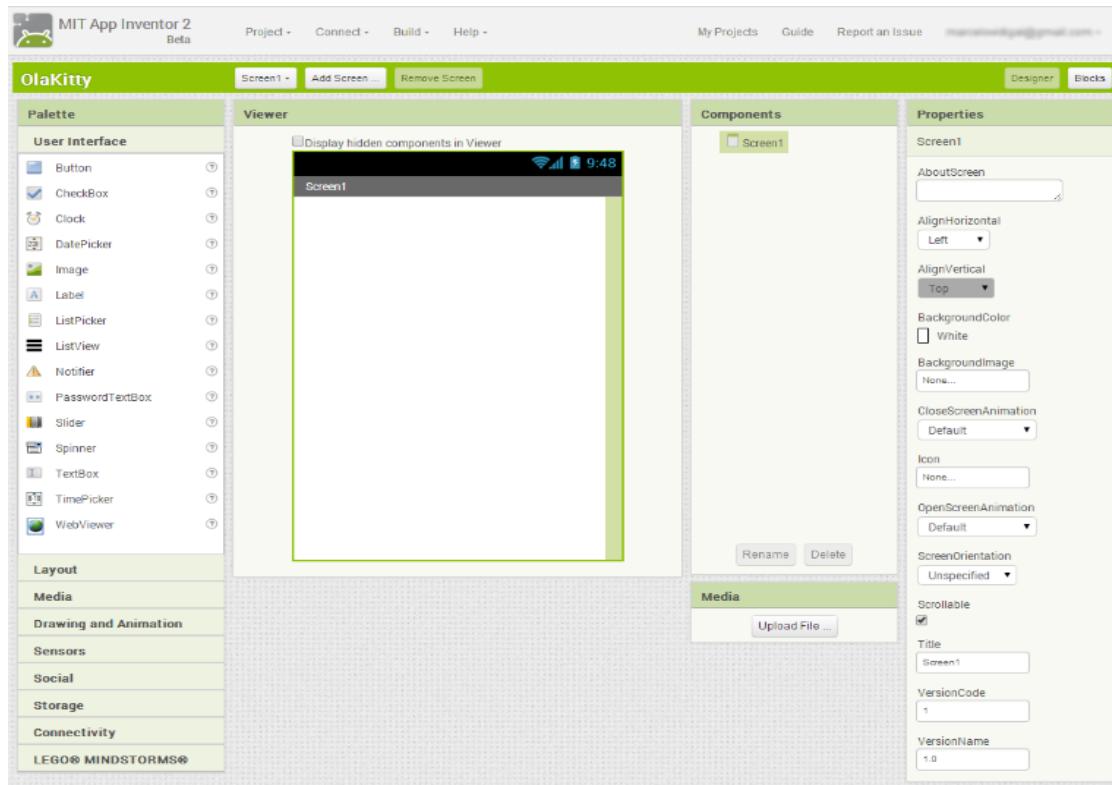
Primeiro projeto

- Para criar um projeto, clique em “New Project” na parte superior esquerda da página



Primeiro projeto

- A primeira janela que se abre é o *Component Designer*



Component Designer

Component designer

Dividida
em 4
colunas

The screenshot shows the MIT App Inventor 2 component designer interface. The interface is divided into four main vertical columns:

- Palette (Left Column):** Contains a list of UI components under "User Interface". Components listed include: Button, CheckBox, Clock, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, TextBox, TimePicker, and WebViewer.
- Viewer (Second Column):** Displays a mobile phone screen simulation titled "Screen1". The screen is currently empty. There is a checkbox labeled "Display hidden components in Viewer" which is unchecked.
- Components (Third Column):** Shows a list of components. In this view, there is one item: "Screen1".
- Properties (Right Column):** Provides settings for the selected component ("Screen1"). The properties shown are:
 - AboutScreen: A text input field containing "Screen1".
 - AlignHorizontal: A dropdown menu set to "Left".
 - AlignVertical: A dropdown menu set to "Top".
 - BackgroundColor: A color swatch set to "White".
 - BackgroundImage: A dropdown menu set to "None...".
 - CloseScreenAnimation: A dropdown menu set to "Default".
 - Icon: A dropdown menu set to "None...".
 - OpenScreenAnimation: A dropdown menu set to "Default".
 - ScreenOrientation: A dropdown menu set to "Unspecified".

Component designer

The screenshot shows the MIT App Inventor 2 component designer interface. A red box highlights the central viewer area, which displays a mobile phone screen representation. The screen shows a white background with the text "Screen1" and a green vertical bar on the right side. The interface includes a palette on the left with various component icons, a components list on the right, and properties settings on the far right.

Palette (User Interface)

- Button
- CheckBox
- Clock
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- TextBox
- TimePicker
- WebViewer

Viewer

Display hidden components in Viewer

Screen1

9:48

Components

Screen1

Properties

Screen1

AboutScreen

AlignHorizontal: Left

AlignVertical: Top

BackgroundColor: White

BackgroundImage: None...

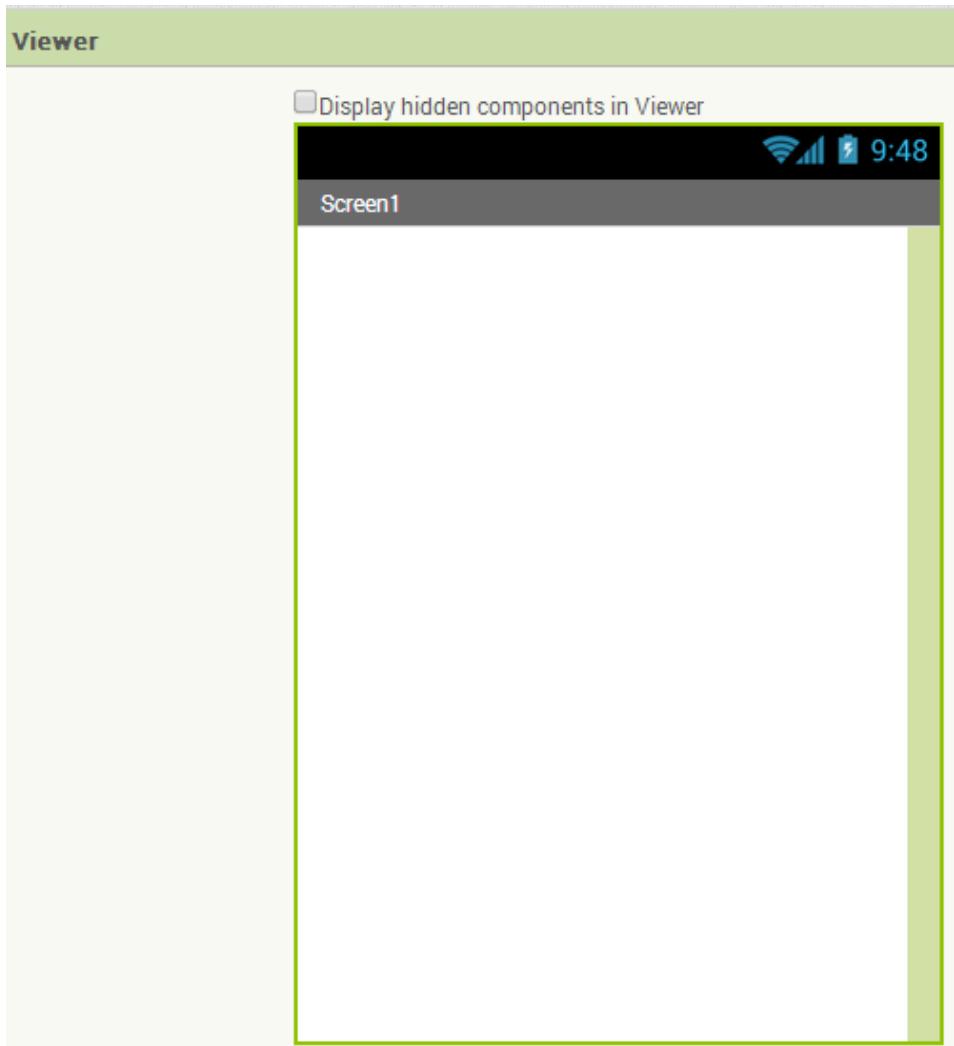
CloseScreenAnimation: Default

Icon: None...

OpenScreenAnimation: Default

ScreenOrientation: Unspecified

Component designer - Viewer



O "Viewer" (Visualizador) permite o usuário organizar cada um de seus objetos, montando o aplicativo como ele deve ser. Uma janela de exibição simula a tela de um smartphone com o sistema operacional Android.

Component designer - Palette

The screenshot shows the MIT App Inventor 2 Component Designer interface. A red box highlights the 'Palette' section on the left side of the screen.

Header: ai2.appinventor.mit.edu/#6694109360160768, MIT App Inventor 2 Beta, Project, Connect, Build, Help, My Projects, Guide, Report an Issue, carlafaria@ifma.edu.br

Toolbar: Add Screen ..., Remove Screen, Designer, Blocks

Palette (highlighted by red box):

- User Interface
 - Button
 - CheckBox
 - Clock
 - DatePicker
 - Image
 - Label
 - ListPicker
 - ListView
 - Notifier
 - PasswordTextBox
 - Slider
 - Spinner
 - TextBox
 - TimePicker
 - WebViewer
- Layout

Viewer: Screen1

Components: Screen1

Properties:

- Screen1
- AboutScreen
- AlignHorizontal: Left
- AlignVertical: Top
- BackgroundColor: White
- BackgroundImage: None...
- CloseScreenAnimation: Default
- Icon: None...
- OpenScreenAnimation: Default
- ScreenOrientation: Unspecified

Palette

User Interface

	Button	
	CheckBox	
	Clock	
	DatePicker	
	Image	
	Label	
	ListPicker	
	ListView	
	Notifier	
	PasswordTextBox	
	Slider	
	Spinner	
	TextBox	
	TimePicker	
	WebViewer	

Component designer - Palette

Na “Palette” (Paleta), é onde ficam todos os componentes utilizáveis num aplicativo. Esta paleta é dividida em seções para facilitar a localização dos componentes, que vão dos básicos (botões, imagens e textos)

Para utilizar um desses componentes basta clicar sobre ele e arrastar para cima da coluna, chamada “Viewer” (Visualizador).

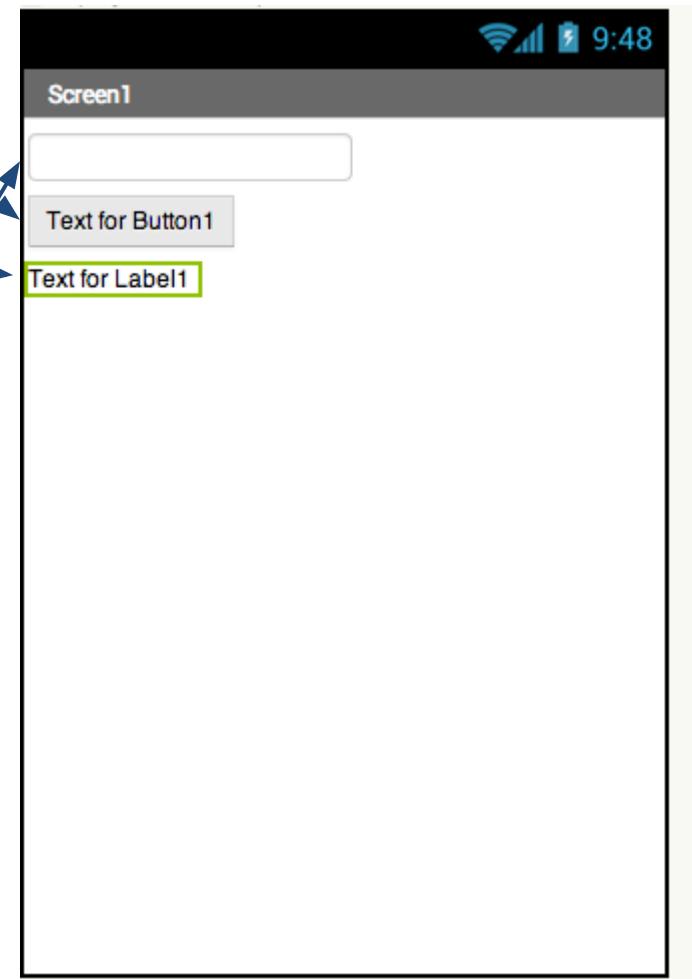
Nem todos os componentes da paleta são visíveis, por exemplo os utilizados para manipular e acessar recursos do celular, como web, sensores ...

Palette

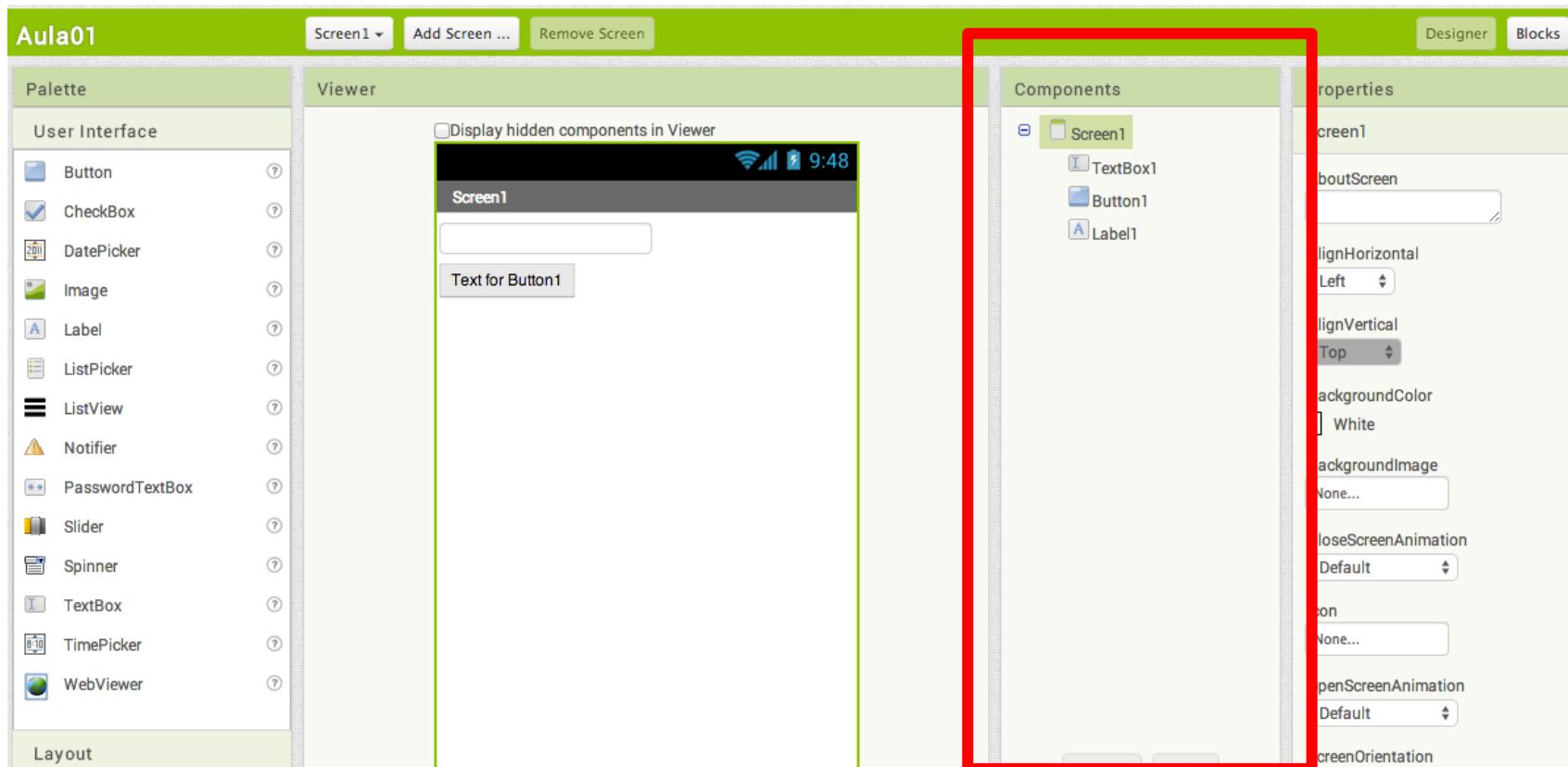
User Interface

-  Button
-  CheckBox
-  Clock
-  DatePicker
-  Image
-  Label
-  ListPicker
-  ListView
-  Notifier
-  PasswordTextBox
-  Slider
-  Spinner
-  TextBox
-  TimePicker
-  WebViewer

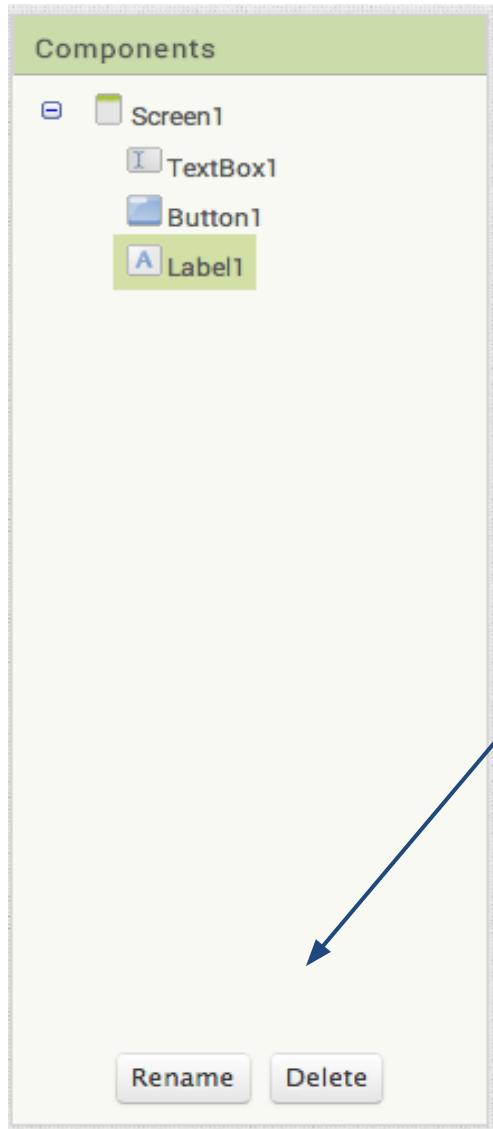
Atividade: Arraste os seguinte componentes para o visualizador: textBox, Button e Label



Component designer - Components



Component designer - Components



Nesta coluna ficam armazenados todos os itens adicionados, sejam eles visíveis ou não na tela do programa.

Feito a atividade anterior, teríamos três componentes, agrupados na tela (screen1).

Aqui podemos também renomear ou deletar os componentes

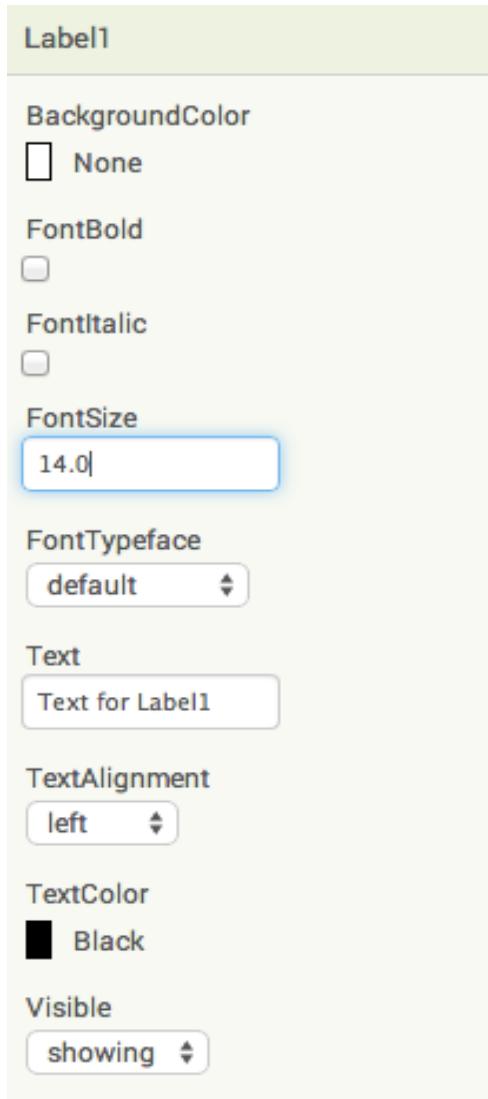
Clicar sobre qualquer um dos itens da lista na coluna “Components” permite que você possa editar seus detalhes na quarta coluna, chamada de “Properties” (**Propriedades**).

Component designer - Properties

The screenshot shows the Component designer interface for a project named "Aula01". The interface is divided into several panels:

- Palette**: A sidebar containing a "User Interface" section with icons for various components like Button, CheckBox, DatePicker, etc., and a "Layout" section.
- Viewer**: A central panel displaying a mobile screen preview titled "Screen1". The screen shows a white area with a placeholder text "Text for Button1".
- Components**: A panel on the right showing a tree structure of components: Screen1 (which contains TextBox1, Button1, and Label1).
- Properties**: A panel on the far right with a red border, listing properties for the selected component "Screen1". These properties include:
 - AboutScreen
 - AlignHorizontal: Left
 - AlignVertical: Top
 - BackgroundColor: White
 - BackgroundImage: None...
 - CloseScreenAnimation: Default
 - Icon: None...
 - OpenScreenAnimation: Default
 - ScreenOrientation

Component designer - Properties

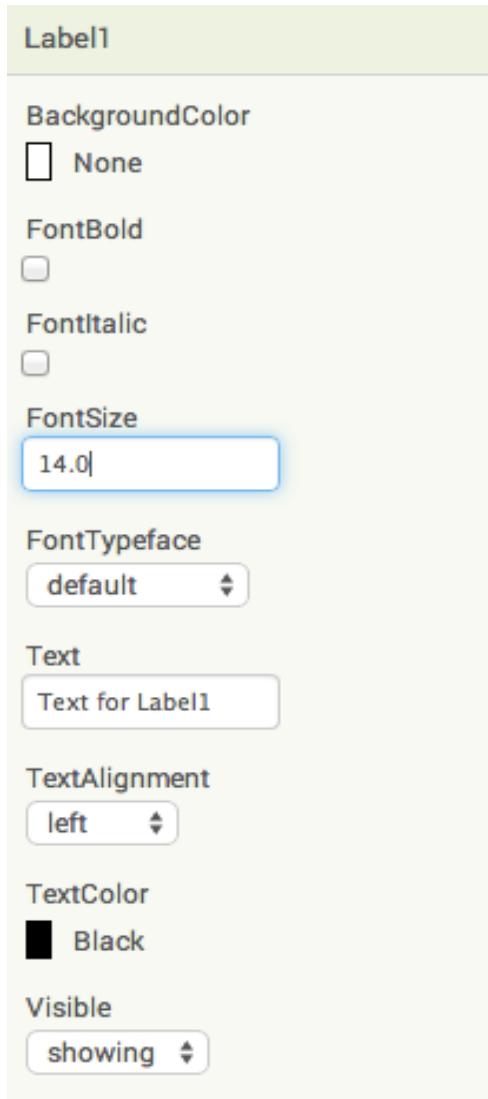


Propriedades são tipicamente associadas com componentes em um aplicativo, como Button, TextBox e Canvas .

Para um aplicativo, cada componente é completamente definido por um conjunto de propriedades

Ela permite definir os tamanhos e conteúdos dos textos de botões e caixas de informação, tamanho das imagens, cores de fundo e largura e altura de objetos.

Component designer - Properties



Atividade: click no componente Label1 e altere suas seguintes propriedades:
Aumente sua fonte (fontsize)
Apague o texto na propriedade text.

Propriedades e ações

As **propriedades** de diversos componentes de um aplicativo podem ser alteradas no momento da execução.

Para isso existe **ações** que mudam as propriedades dos objetos, como o texto, a cor, o tamanho

A simple line drawing of a young boy with short, wavy hair. He is wearing a light-colored, short-sleeved shirt. His right hand is raised to his chin, with his index finger touching his temple, a classic pose for someone deep in thought. Two small, black question marks are floating near his head, one on each side, further emphasizing his疑惑 (confusion).

Onde encontramos estas
ações ?



Você encontrará as ações no block designer, é onde toda programação ocorre.

Block Designer

Clique em
“Blocks”

Aula01

Screen1 ▾ Add Screen ... Remove Screen

Designer Blocks

Palette

User Interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- TextBox
- TimePicker
- WebViewer

Layout

Viewer

Display hidden components in Viewer

Screen1

Text for Button1

Components

- Screen1
 - TextBox1
 - Button1
 - Label1

Properties

Screen1

AboutScreen

AlignHorizontal Left

AlignVertical Top

BackgroundColor White

BackgroundImage None...

CloseScreenAnimation Default

Icon None...

OpenScreenAnimation Default

ScreenOrientation

Block Designer

Dividida
em 2
colunas

The screenshot shows the Block Designer application window titled "Aula01". The top navigation bar includes "Screen1", "Add Screen ...", "Remove Screen", "Designer" (which is selected), and "Blocks". The left sidebar, titled "Blocks", lists categories: Built-in (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), Screen1 (TextBox1, Button1, Label1), and Any component. The main area is titled "Viewer" and contains a blank workspace. At the bottom of the workspace are two warning icons (yellow triangle with exclamation mark and red triangle with exclamation mark) both set to 0, and a "Show Warnings" button. A green trash can icon is located in the bottom right corner of the workspace.

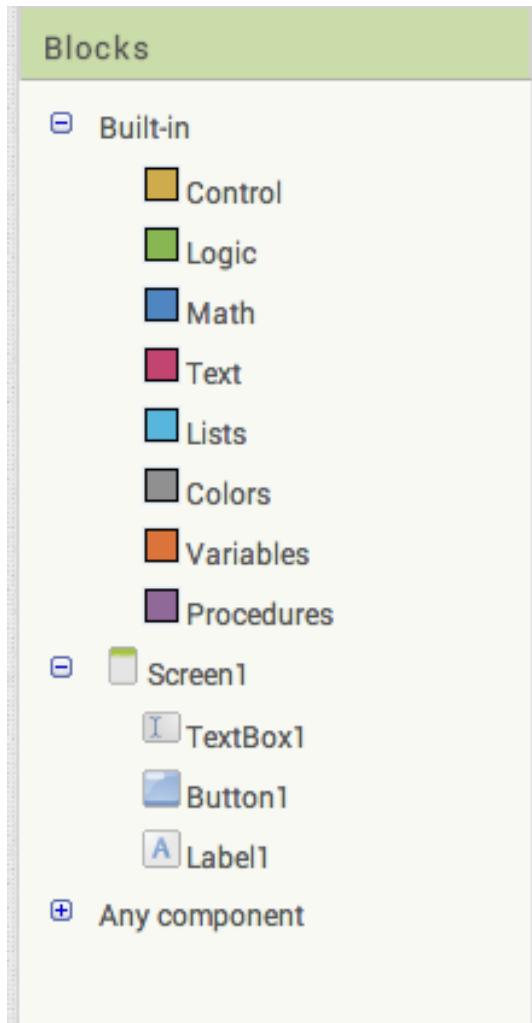
Block Designer - Blocks

The screenshot shows the 'Block Designer' application window titled 'Aula01'. The top menu bar includes 'Screen1', 'Add Screen ...', 'Remove Screen', 'Designer' (which is selected), and 'Blocks'. The left sidebar is labeled 'Blocks' and contains a tree view of components:

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Colors
 - Variables
 - Procedures
- Screen1
 - TextBox1
 - Button1
 - Label1
- Any component

A red box highlights the 'Blocks' sidebar. The main area is labeled 'Viewer' and is currently empty. At the bottom of the viewer area, there are two warning icons (yellow triangle with exclamation mark and red triangle with exclamation mark) both set to '0', and a 'Show Warnings' button.

Block Designer - Blocks



Aqui estão dispostos os diversos blocos utilizados para a programação. Em “Built-in” (Internos) blocos para comandos mais gerais, como operações logicas, matemáticas e de controle. Em “Screen1” (Tela1) tem todos os objetos que você inseriu em seu programa. As ações para cada objeto serão encontradas aqui.

Block Designer - Blocks

The image shows the Block Designer interface. On the left, the 'Blocks' sidebar lists categories like Built-in, Screen1, and Any component, with specific blocks like Control, Logic, and Label1 highlighted. On the right, the 'Viewer' pane displays a stack of green Scratch-style blocks for a script:

- Label1 . BackgroundColor
- set Label1 . BackgroundColor to [red color]
- Label1 . FontSize
- set Label1 . FontSize to [16px]
- Label1 . Height
- set Label1 . Height to [100px]
- Label1 . Text
- set Label1 . Text to [Hello World!]
- Label1 . TextColor
- set Label1 . TextColor to [blue color]

Two red arrows point from the text "Ação que altera o tamanho da fonte de Label1" to the 'FontSize' block and from the text "Ação que altera o texto mostrado em label1" to the 'Text' block.

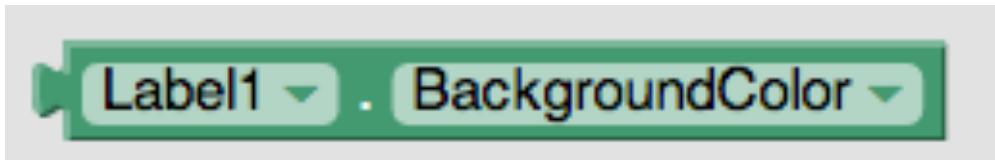
Ação que altera o tamanho da fonte de Label1

Ação que altera o texto mostrado em label1

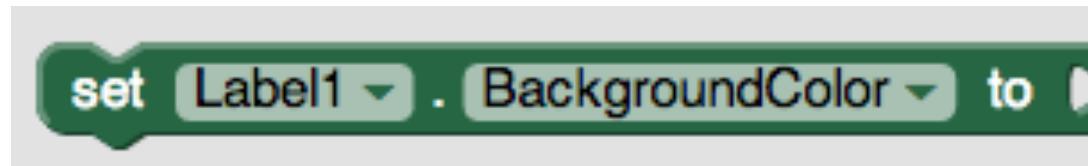
Block Designer - Blocks

Em computação é comum chamar as ações que retorna uma propriedade de getters, e setters as ações que alteram uma propriedade.

Exemplo de um *getter*



Exemplo de um *setter*





Ok. Mas como estas ações
são inicializadas ?



Além das ações, os objetos tem também eventos. Um aplicativo basicamente executa ações quando ocorre algum evento.

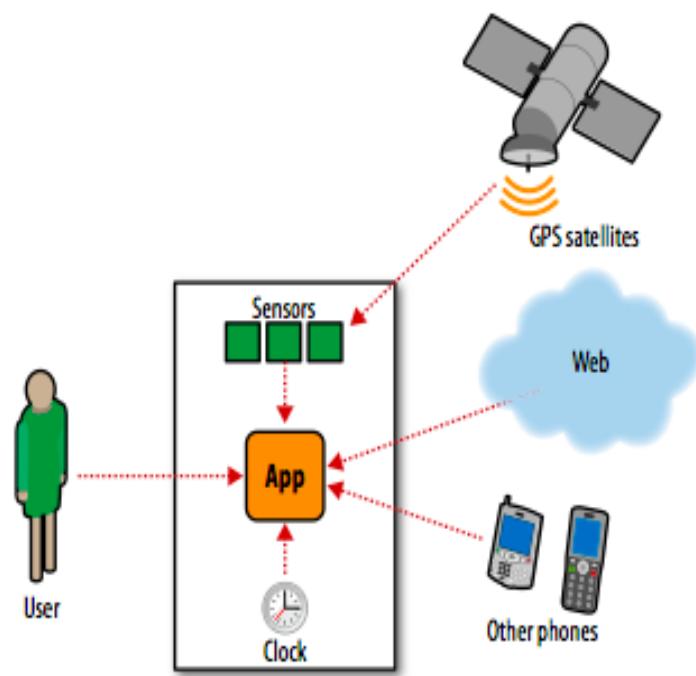
Eventos

- O comportamento de um aplicativo é definido por um conjunto de manipuladores de eventos
- Cada manipulador de eventos executa funções específicas em resposta a um acontecimento particular

Eventos

Um click em um botão é um exemplo de um evento iniciado pelo usuário. Contudo, existem outros tipos de eventos:

1. Timer events,
2. Sensor events,
3. Phone events,
4. Animation events,
5. Web events,
6. App (Screen) launch events.



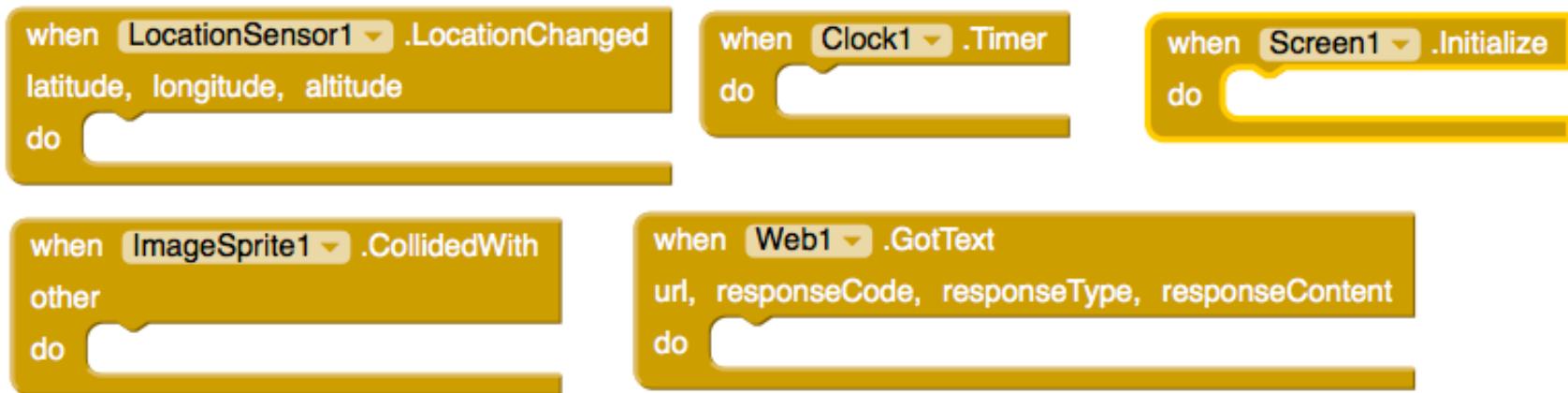
Conceito Chave

eventos + ações

Quando ocorrer um determinado evento,
execute uma dada ação.

Eventos + ações

Exemplo de eventos



Exemplo de ações





Ok. Já posso tentar
começar a programar.



App é bem simples. O usuário vai digitar o seu nome (ex. Joao) na text box, clicar no botão e então será mostrado a seguinte mensagem:
Olá Joao.

Voltando ao editor de blocos ...

Aula01 Screen1 ▾ Add Screen ... Remove Screen Designer Blocks

Blocks Viewer

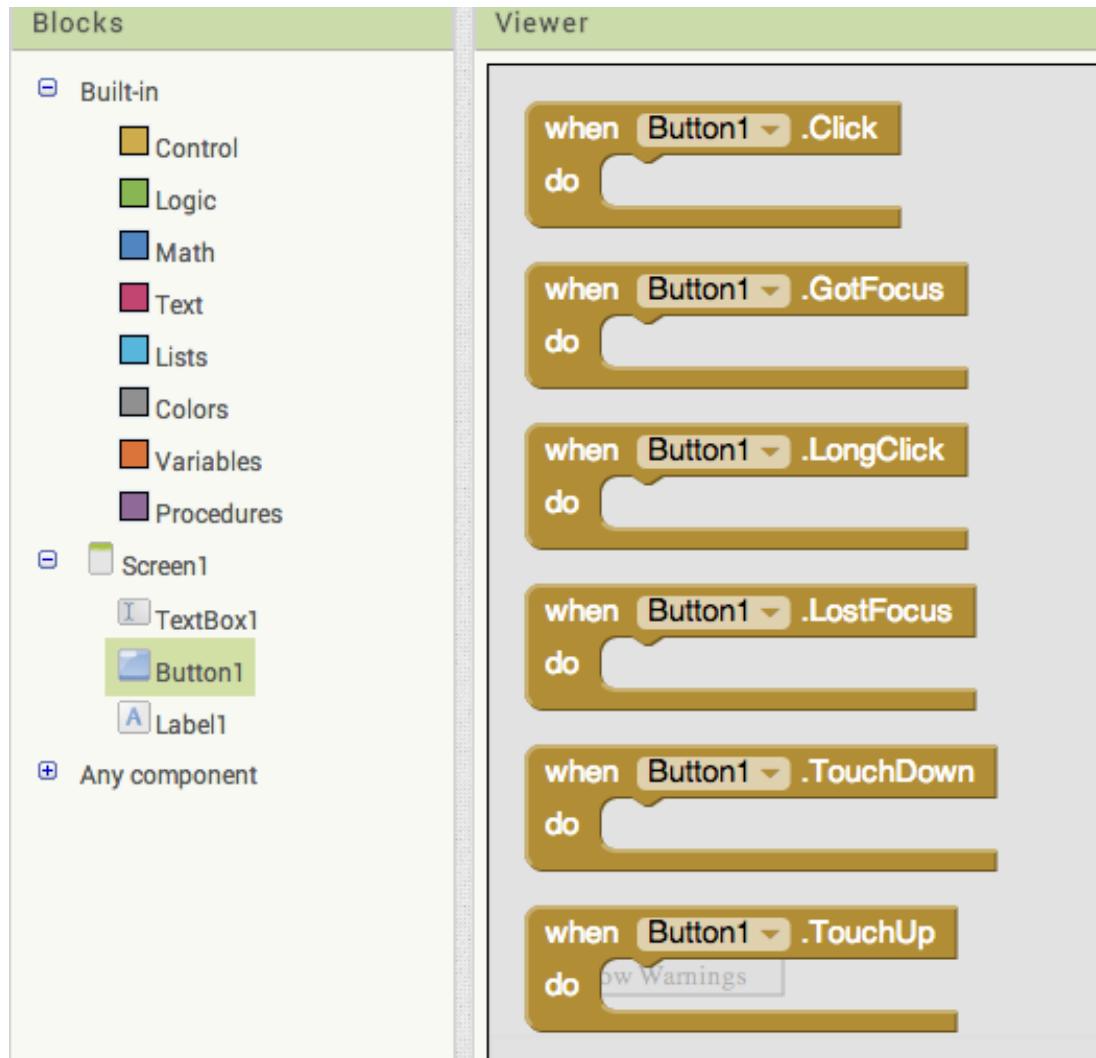
- ⊕ Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Colors
 - Variables
 - Procedures
- ⊕ Screen1
 - TextBox1
 - Button1
 - Label1
- ⊕ Any component

⚠ 0 ⚡ 0

Show Warnings

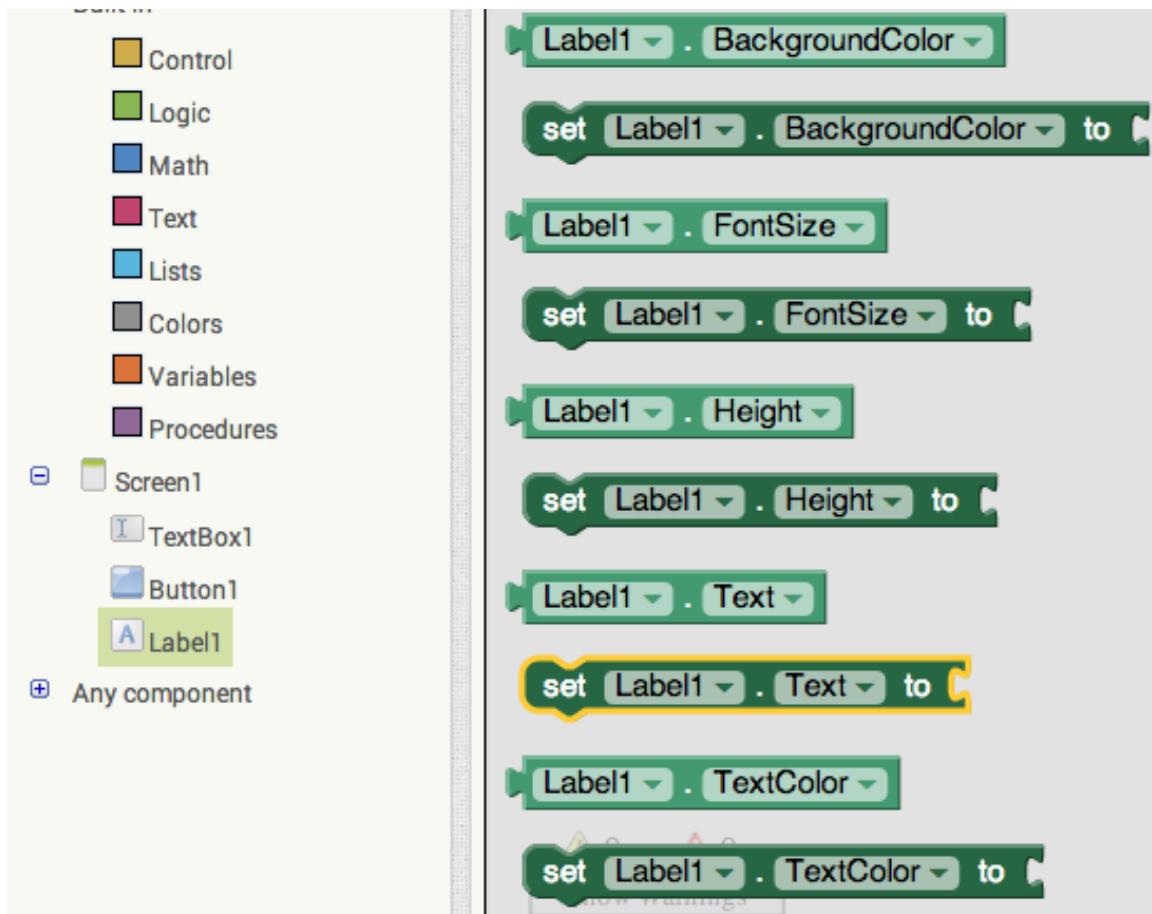


Programando o comportamento

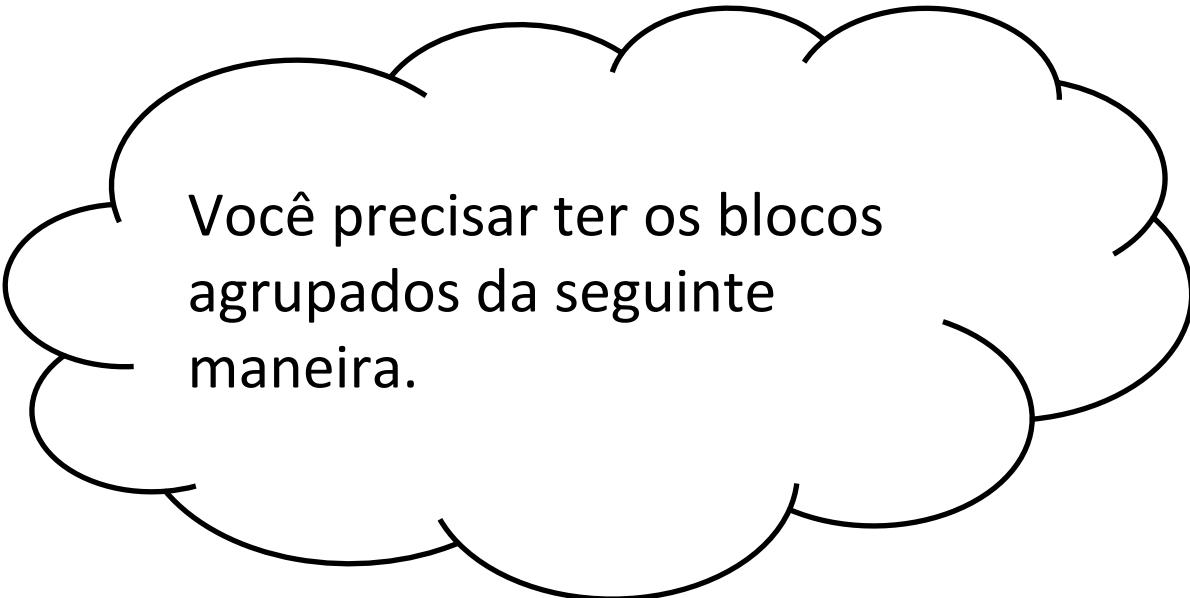


1. Seleciona o componente Button1.
2. Arraste o evento “When Button1. click” para o visualizador de blocos.

Programando o comportamento



3. Seleciona o componente Label.
4. Arraste a ação “set Label1.text” para o visualizador de blocos dentro do evento anterior.



Você precisar ter os blocos
agrupados da seguinte
maneira.



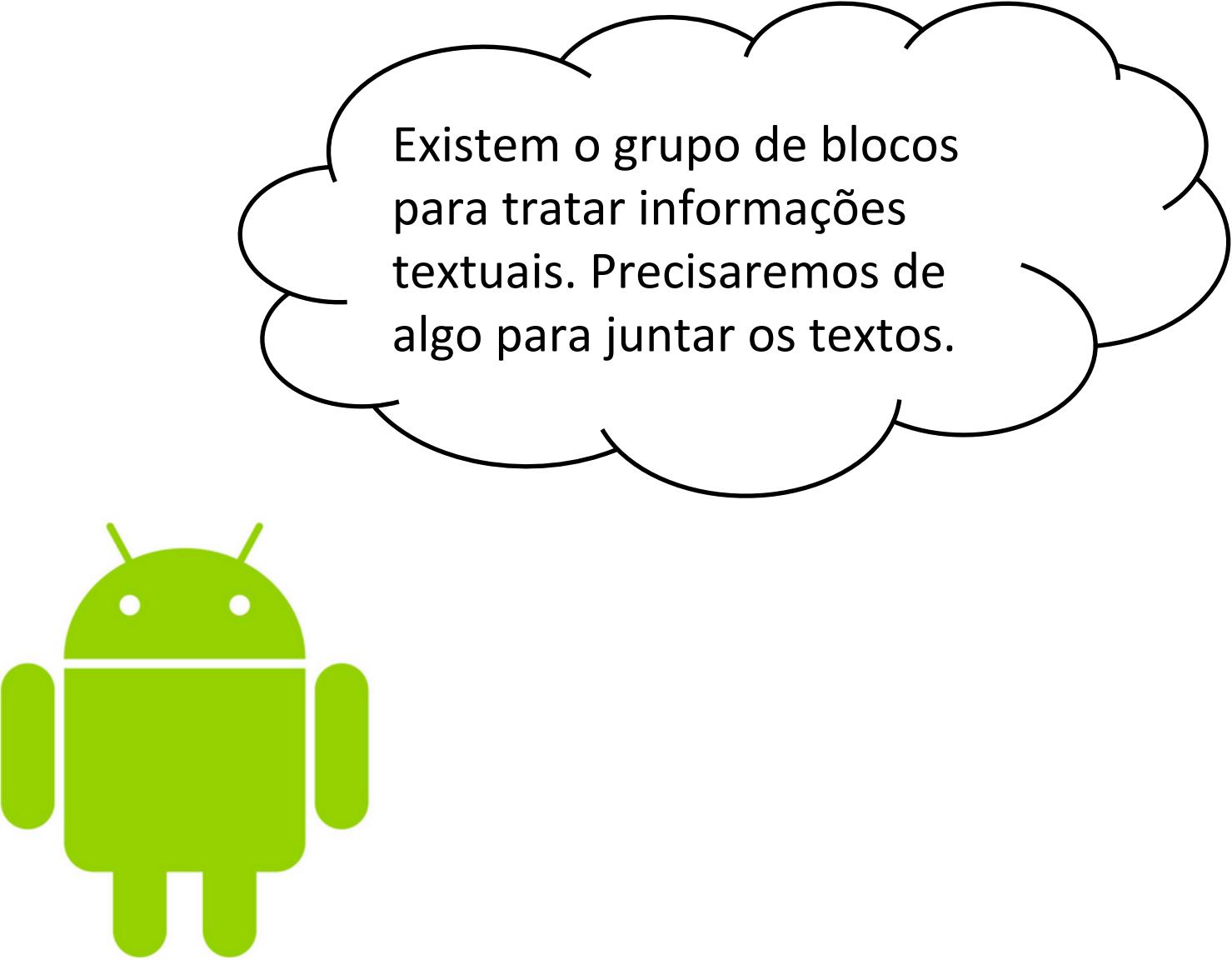
```
when Button1 .Click
do [set Label1 .Text to ]
```



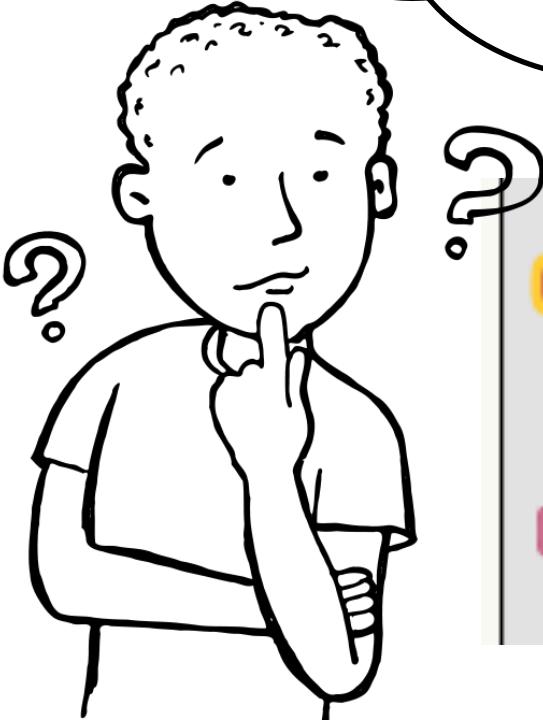
A mensagem a ser exibida
em label1 deve ser Ola + o
nome digitado.



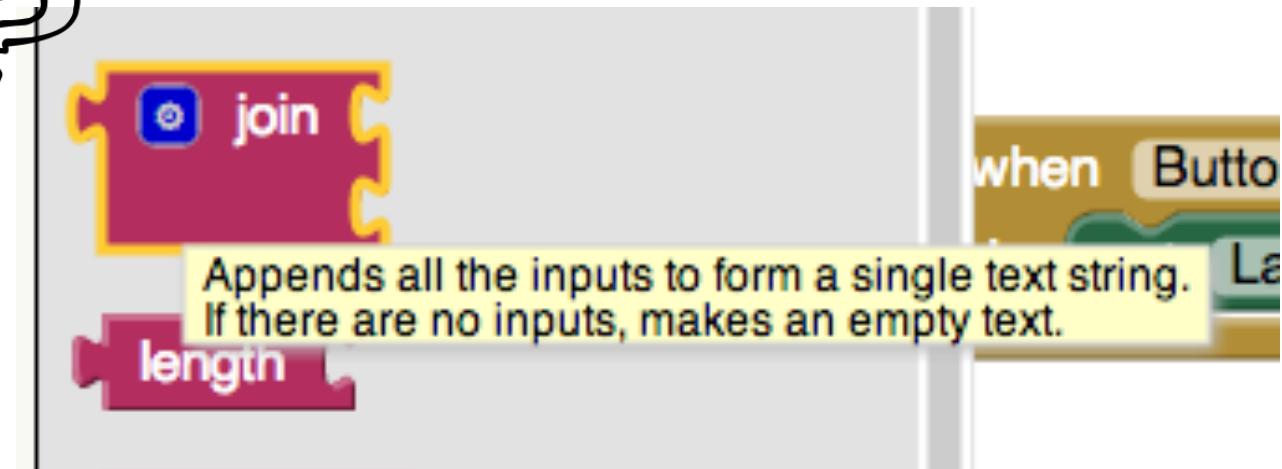
Ou seja, precisaremos
juntar estes dois textos.



Existem o grupo de blocos para tratar informações textuais. Precisaremos de algo para juntar os textos.



Só pode ser esse bloco
Join. Ao apontar para ele
mostrou a seguinte
mensagem.

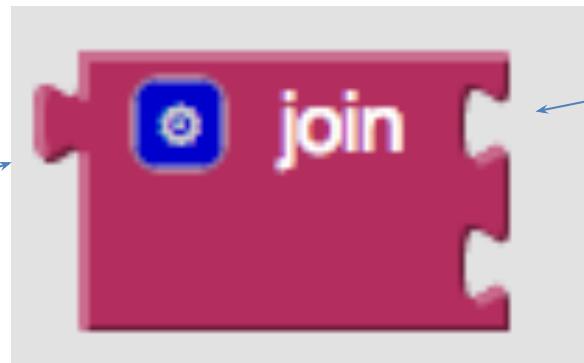




Isso mesmo, esta é uma boa forma para explorar e conhecer o App Inventor.

Anatomia de um bloco

Sai o texto
concatenado



Entra dois textos
nesta
extremidade.

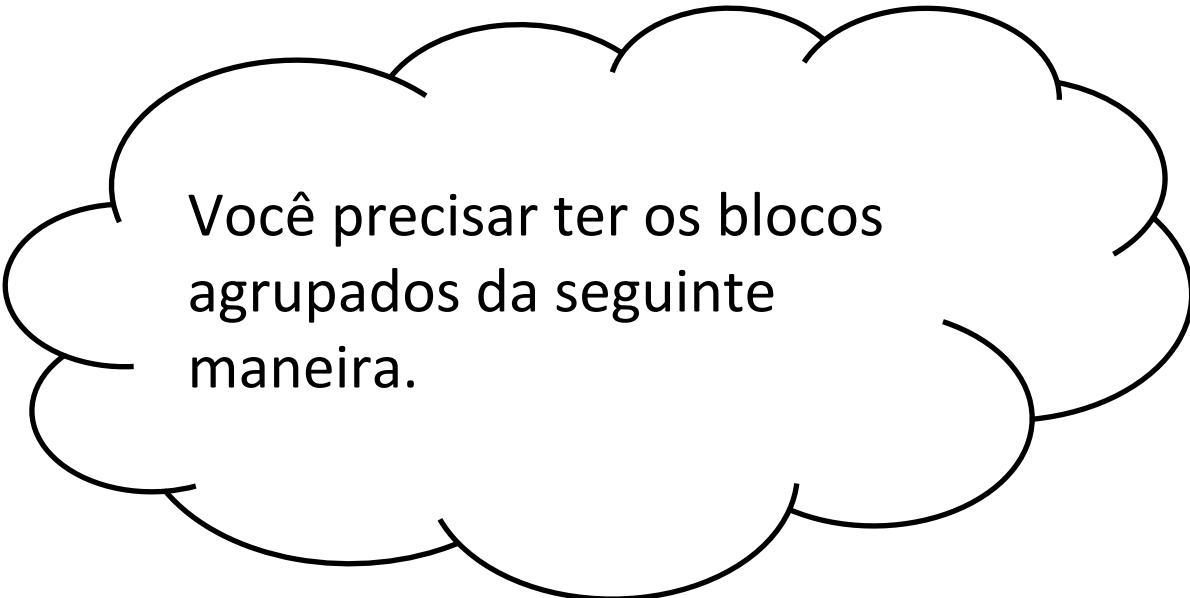
Programando o comportamento

The image shows the Scratch interface. On the left is the 'Blocks' palette, which is divided into sections: 'Built-in' (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), 'Screen1' (TextBox1, Button1, Label1), and 'Any component'. The 'Text' section is currently selected, indicated by a green background. On the right is the 'Viewer' stage area, which displays several text-related blocks:

- "()"
- join
- length
- is empty
- compare texts < ▾
- trim
- upcase ▾
- starts at text ▾

A blue arrow points from the 'join' block in the viewer towards the 'Text' section in the blocks palette.

5. Seleciona o grupo de blocos Text.
6. Arraste a ação Join para o visualizador de blocos. A saída da Join será a entrada do “Set Label1. Text”.



Você precisar ter os blocos
agrupados da seguinte
maneira.



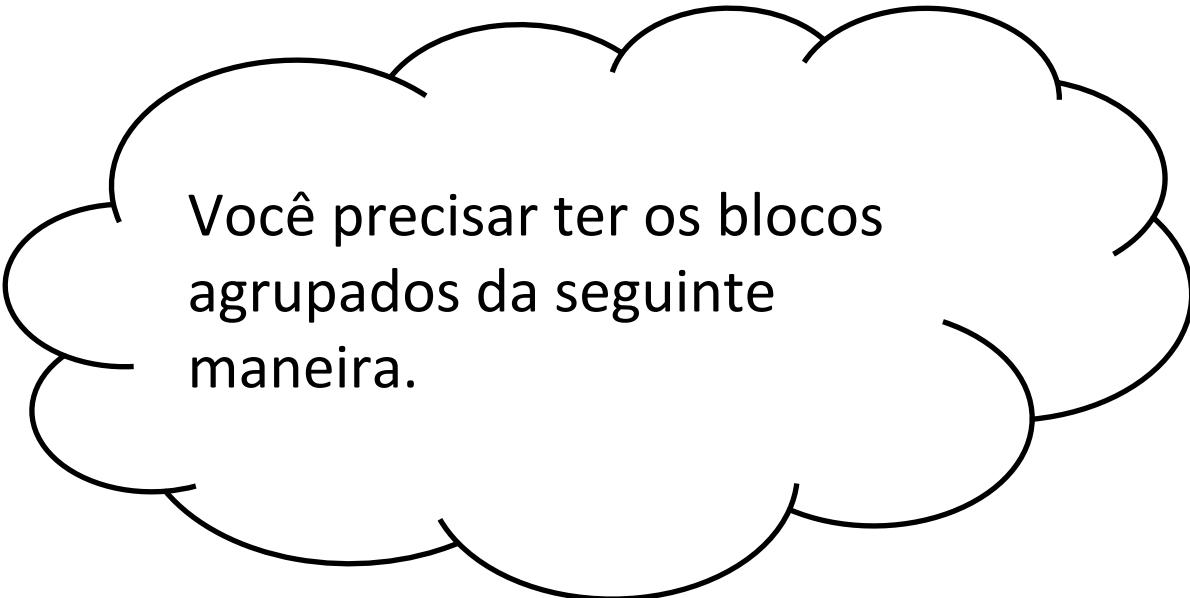
Programando o comportamento

The image shows the Scratch interface. On the left is the 'Blocks' palette, which is divided into sections: 'Built-in' (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), 'Screen1' (TextBox1, Button1, Label1), and 'Any component'. On the right is the 'Viewer' stage area, which contains several text-related blocks:

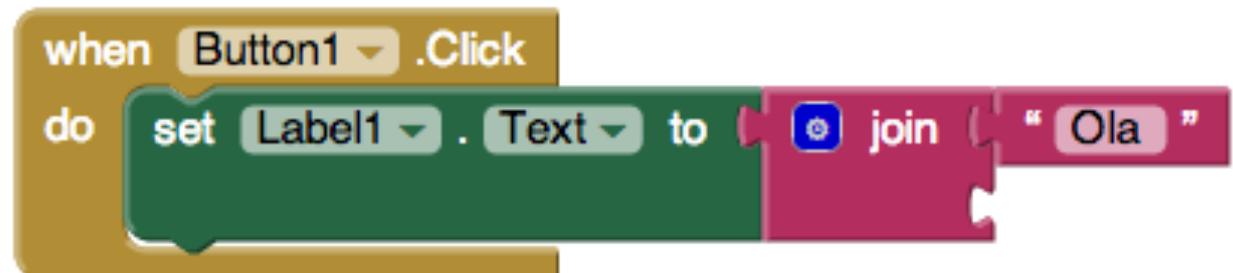
- " " (text block)
- join [] (text block)
- length [] (text block)
- is empty [] (text block)
- compare texts [] < [] (text block)
- trim [] (text block)
- upcase [] (text block)
- starts at [] text [] (text block)

A red arrow points from the text "7. Seleciona o grupo de blocos Text." to the "Text" block in the palette.

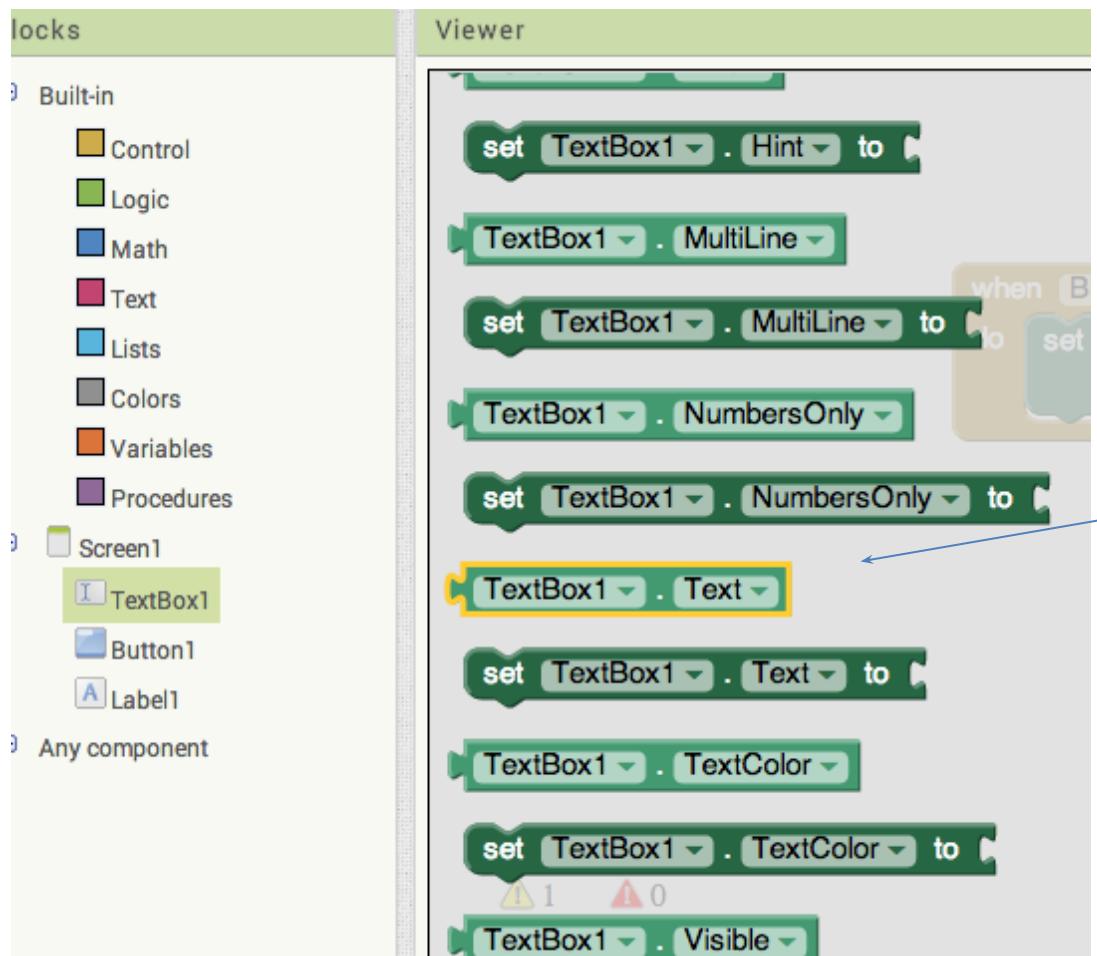
7. Seleciona o grupo de blocos Text.
8. Arraste o bloco que cria um texto.
9. Escreva a mensagem “Ola”



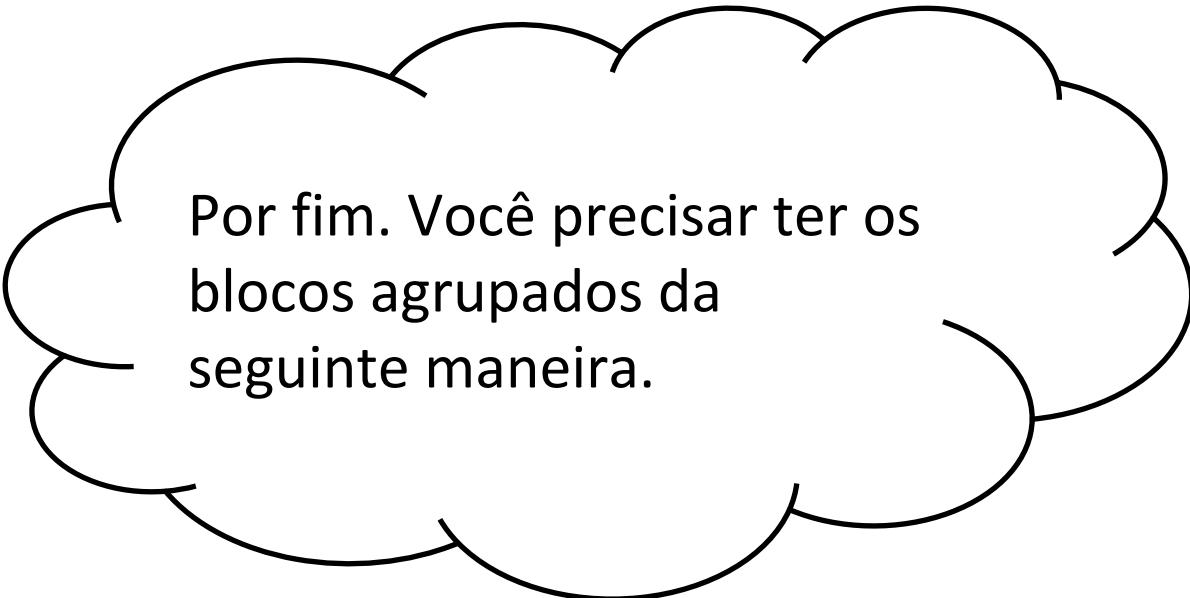
Você precisar ter os blocos agrupados da seguinte maneira.



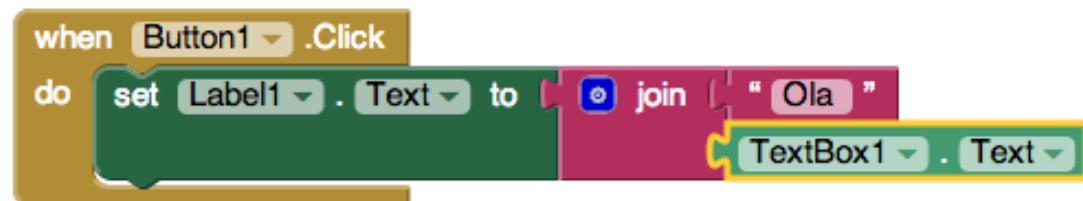
Programando o comportamento



- 10 . Seleciona o componente TextBox
11. Arraste o bloco que retorna o texto para dentro do visualizado, como entrada do bloco Join.

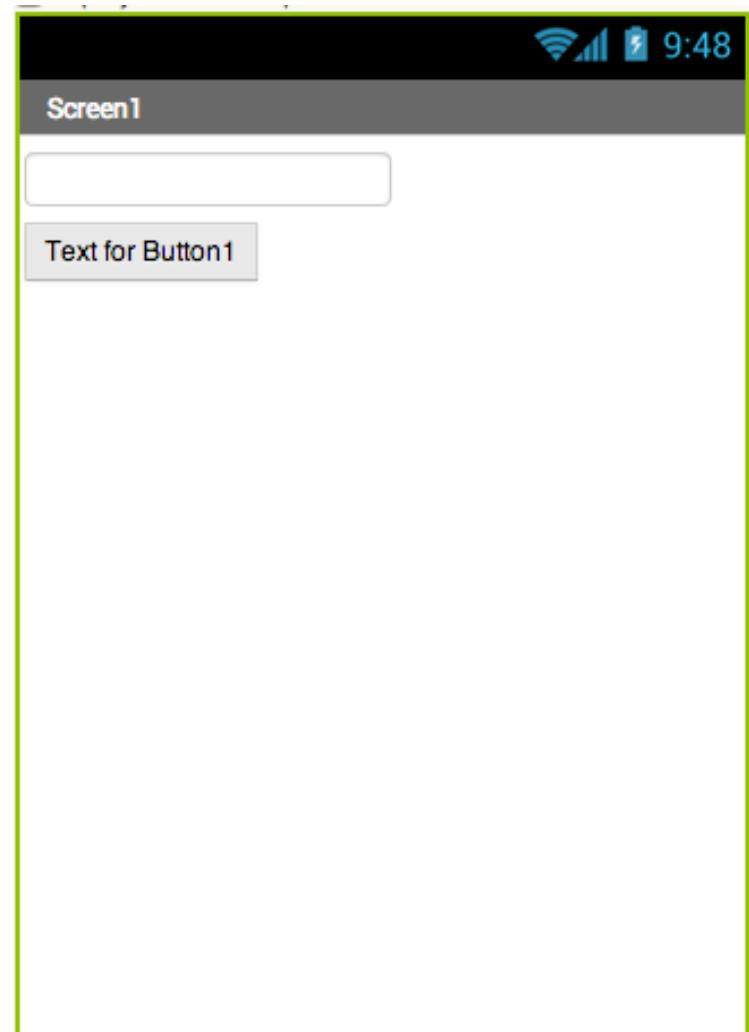


Por fim. Você precisar ter os blocos agrupados da seguinte maneira.



```
when [Button1].Click
do [set [Label1.Text] to [join ["Olá " [TextBox1.Text]]]]
```

Relembrando a tela do
nossa aplicativo. Vamos
testar ?



Instalem o MIT AI2 Companion em seu smartphone



MIT AI2 Companion

MIT Center for Mobile Learning - 5 de julho de 2014
Educação

Instalado

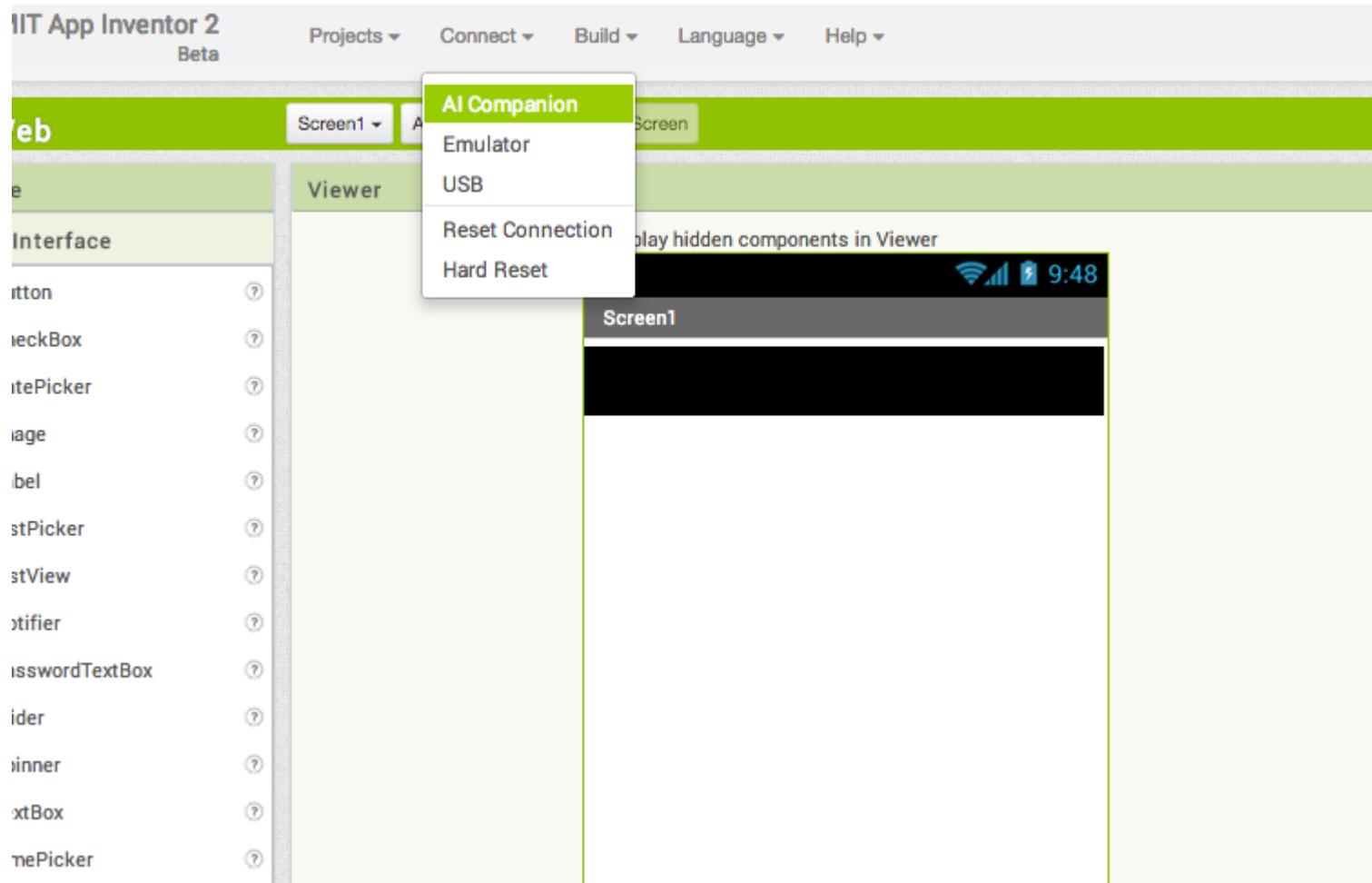
Este aplicativo é compatível com todos os seus dispositivos.

★★★★★ (4.530)

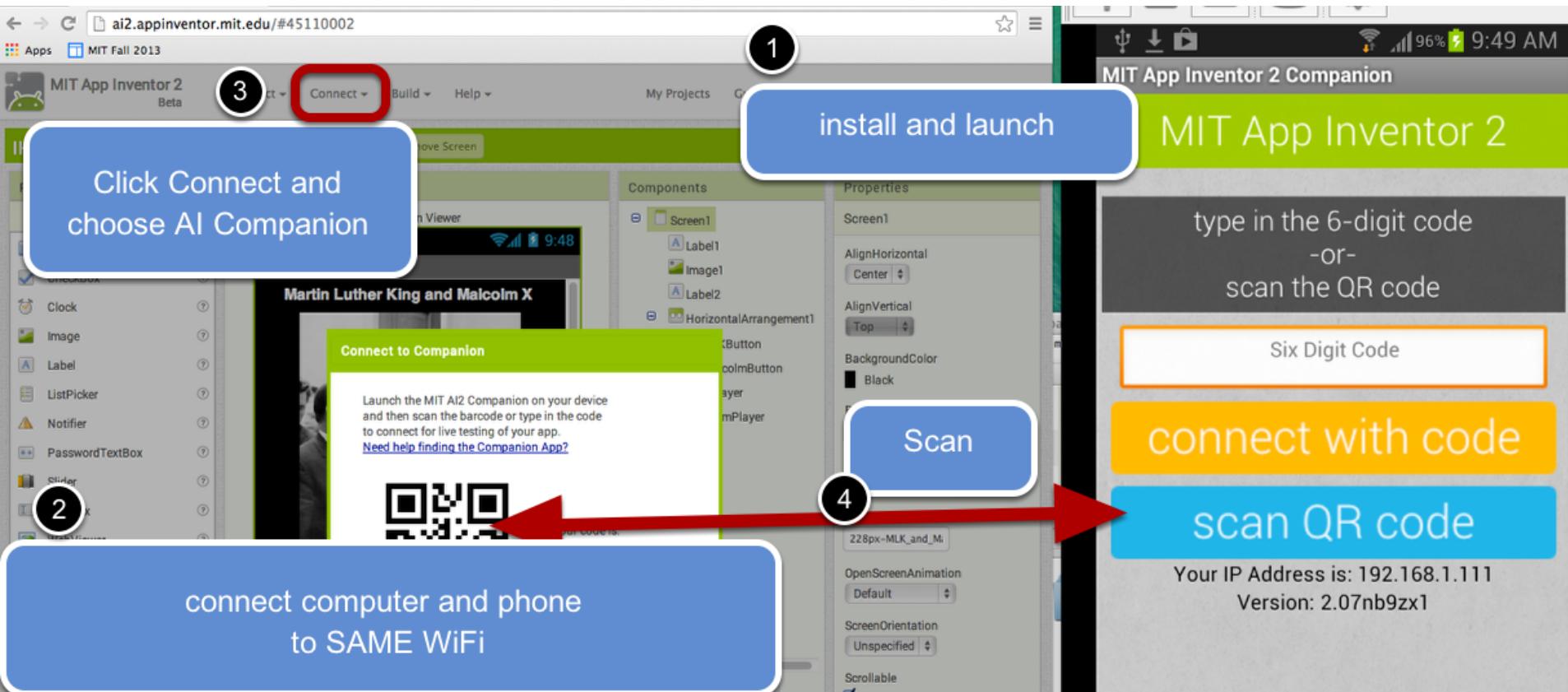
[g+1 +665 Recomende isto no Google](#)

The screenshot shows the app's main screen with three connection methods: 'Six Digit Code', 'connect with code', and 'scan QR code'. Below these, it displays the IP address '192.168.1.107' and version '2.11a2zx1'. To the right, there is a smaller screen showing a ginger cat with the text 'Pet the Kitty!' and a note about loading via the MIT AI2 Companion.

Testando usando o MIT App Inventor 2 Companion



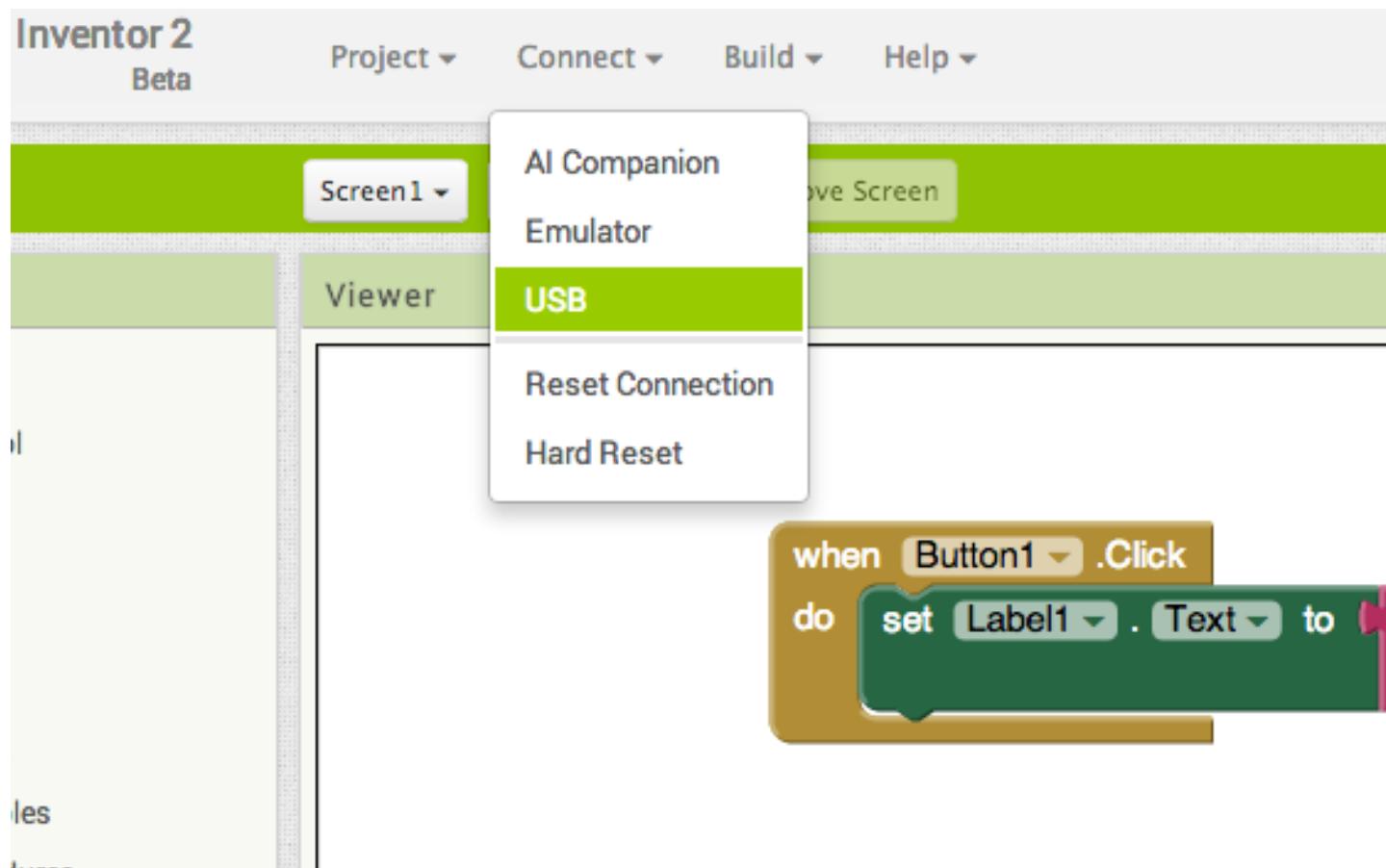
Testando usando o MIT App Inventor 2 Companion



<http://www.appinventor.org/appInventor2Changes>

ou ...

Conekte seu smartphone android a porta usb do computador, e seleciona Connect e depois em USB.

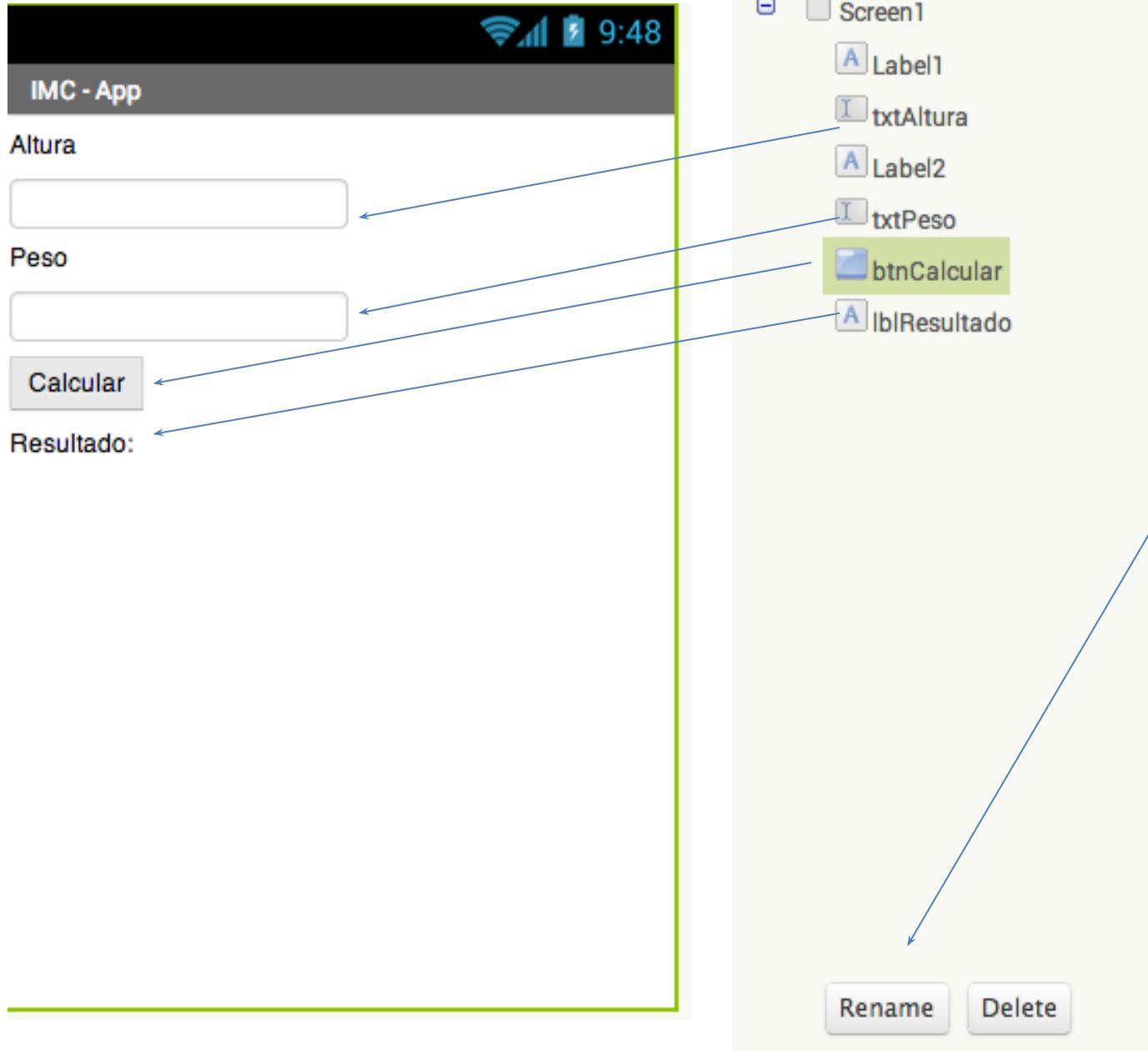


Atividade guiada

Criar um aplicativo para calcular o IMC (Índice de Massa Corpórea).

1. Crie um novo projeto chamado IMC-APP
2. Monte a interface do aplicativo como a seguir

Atividade guiada



Lembre-se de renomear os componentes.

Atividade guiada

The image shows a Scratch script editor interface. On the left is the 'Blocks' palette, which is divided into sections: 'Built-in' (Control, Logic, Math, Text, Lists, Colors, Variables, Procedures), 'Screen1' (Label1, txtAltura, Label2, txtPeso, btnCalcular, lblResultado), and 'Any component'. On the right is the 'Viewer' pane, which displays a script consisting of the following blocks:

- A blue control block: `0`.
- A blue control block: `[+ -] = []`.
- A blue control block: `[+ -] + []`.
- A blue control block: `[+ -] - []`.
- A blue control block: `[+ -] × []`.
- A yellow control block: `[+ -] / []`. Two blue arrows point from the text 'Para fazer o cálculo do IMC basta dividir seu peso em quilogramas pela altura ao quadrado (em metros). Então precisaremos dos blocos para dividir e multiplicar.' to this block.
- A blue control block: `[^]`.
- A blue control block: `random integer from [1] to [100]`.
- A blue control block: `random fraction`.

Para fazer o cálculo do IMC basta dividir seu peso em quilogramas pela altura ao quadrado (em metros). Então precisaremos dos blocos para dividir e multiplicar.

Atividade guiada

Os blocos devem estar como na figura abaixo.



The image shows a screenshot of a mobile application titled "IMC - App". At the top, there is a status bar with various icons including signal strength, battery level (76%), and the time (20:30). Below the status bar, the app title "IMC - App" is displayed. The main interface consists of several input fields and a button:

- Altura:** An input field containing the value "1.68".
- Peso:** An input field containing the value "64", which is highlighted with a red border.
- Calcular:** A grey button labeled "Calcular".
- Result:** The calculated result "22,67574" is displayed below the button.

At the bottom of the screen is a numeric keyboard:

- Top row: Numbers 1 through 0.
- Second row: Special characters @, #, \$, %, &, -, +, (,).
- Third row: Mathematical operators = \ < *, " ', : ; ! ? and a delete key (X).
- Bottom row: ABC, -, /, a blank gray button, , ., and a green checkmark button.

Na próxima aula, veremos como tomar decisões. Iremos incrementar nosso app de IMC, para dar as seguintes mensagens.



Cálculo IMC Situação

- | | |
|-------------------|--|
| Abaixo de 18,5 | Você está abaixo do peso ideal |
| Entre 18,5 e 24,9 | Parabéns — você está em seu peso normal! |
| Entre 25,0 e 29,9 | Você está acima de seu peso (sobre peso) |
| Entre 30,0 e 34,9 | Obesidade grau I |
| Entre 35,0 e 39,9 | Obesidade grau II |
| 40,0 e acima | Obesidade grau III |

The screenshot shows a mobile application interface for calculating BMI (IMC). At the top, there are icons for battery, signal, and time (18:15). The title bar says "IMC - App". Below it, there are two input fields: "Altura" with the value "1.68" and "Peso" with the value "64", which is highlighted with an orange border. A "Calcular" button is below the inputs. A message at the bottom says "Parabéns – você está em seu peso normal!" (Congratulations – you are in your normal weight!). At the bottom, there is a numeric keypad with digits 0-9, symbols (@, #, \$, %, &, -, +, (,)), and other special characters (=, \, <, *, ", ', :, ;, !, ?, x). There are also buttons for "ABC", a decimal point, and a checkmark.

Referências

- The App Inventor Course-in-a-Box (AI2 Version) - <http://www.appinventor.org/course-in-a-box2>