

# MIT App Inventor 2: Introdução

Saulo Popov Zambiasi  
*saulopz@gmail.com*

16 de março de 2020

## 1 Introdução

O MIT App Inventor 2 [MI2] é uma ferramenta web para a criação de aplicativos para dispositivos móveis com Sistema Operacional Android. A ferramenta, Google App Inventor, foi originalmente desenvolvida pela Google que descontinuou o projeto mas liberou como *open source*, permitindo que outras empresas pudessem seguir com seu desenvolvimento. O MIT (*Massachusetts Institute of Technology*) decidiu abraçar o projeto, gerando uma segunda versão com o nome de MIT App Inventor 2.

Inicialmente, a ferramenta da Google era composta por três módulos, o módulo de *Design* que era acessado via navegador web, o módulo de código (*Blocks*) que era um programa java em comunicação com o módulo web e um emulador. Desde que o MIT assumiu o projeto, muitas melhorias têm sido feitas. Na versão atual, os módulos *Design* e *Blocks* se encontram na ferramenta web. Ainda existe um emulador (não muito bom), mas há um aplicativo na Google Play chamado **MI2 Companion** para fazer a integração do aplicativo em um dispositivo móvel com a ferramenta web, permitindo *debug*, instalação, e testes.

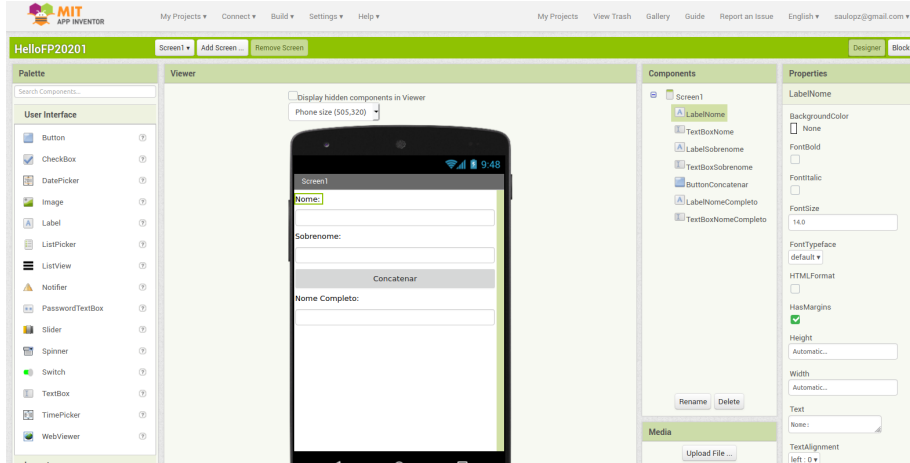
## 2 Interface

A interface do MIT App Inventor 2 é separada em duas partes principais, a parte de *Design* (Figura 1) e a parte de programação, ou *Blocks*, (Figura 2).

A interface do *Design* é composta por 4 colunas principais, a primeira coluna é a Paleta de componentes que podem ser utilizados para compor o aplicativo, a segunda coluna é a visualização do aplicativo, a terceira são os componentes utilizados e sua estrutura hierárquica e a quarta coluna são as propriedades dos componentes.

A interface do *Blocks* é composta por duas colunas principais. Na primeira coluna se encontram os componentes (ou *Blocks*) que podem ser utilizados para criar o programa, isso inclui alguns componentes básicos (*Built-in*) e os componentes referentes aos elementos utilizados para criar o *design* do aplicativo. A segunda coluna é a visualização do programa. Quando um componente é

Figura 1: Interface do MIT App Inventor 2: *Design*



selecionado, abre uma aba na coluna da visualização para poder escolher a ação que se quer utilizar.

### 3 Programa *Hello world*

É preciso você ter uma conta na Google para poder utilizar o MIT App Inventor 2, então antes de começar a utilizar, crie uma conta na Google caso você ainda não tenha.

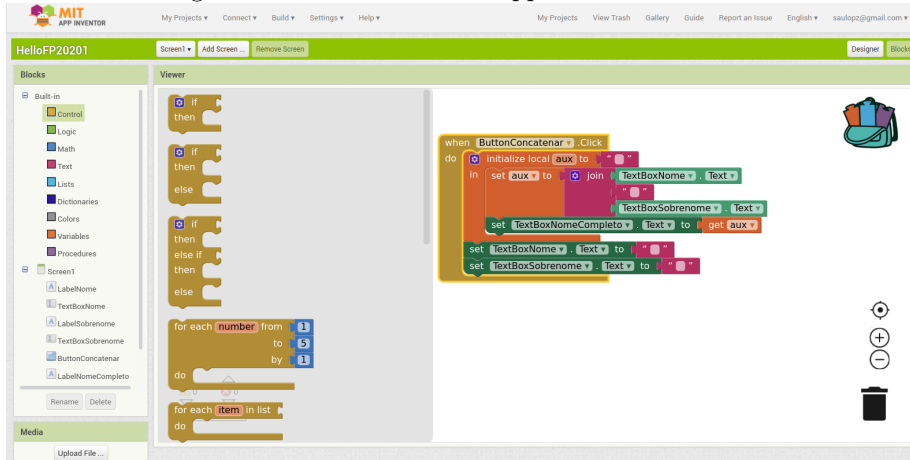
Vamos apresentar aqui o passo a passo da criação de um programa *Hello World*), apesar de ir um pouco além disso. Vamos colocar alguns componentes no *Design* para poder fazer eles interagirem.

**Primeiro**, crie um projeto. Então vá no menu superior, em *My Projects* e selecione *Start new project*, dê o nome dele de **HelloWorld**, por exemplo.

A ferramenta vai abrir a parte do *Design*, então vamos começar a inserir alguns elementos.

1. Inserir um **Label** (renomeie para LabelNome) e altere as seguintes propriedades:
  - Text: Nome
2. **TextBox** (TextBoxNome):
  - Width: Fill parent...
  - Hint: Digite seu nome
3. **Label** (LabelSobrenome):

Figura 2: Interface do MIT App Inventor 2: *Blocks*



- Text: Sobrenome
4. **TextBox** (TextBoxSobrenome):
    - Width: Fill parent...
    - Hint: Digite seu sobrenome
  5. **Button** (ButtonConcatenar):
    - Text: Concatenar
    - Width: Fill parent...
  6. **Label** (LabelNomeCompleto):
    - Text: Nome Completo
  7. **TextBox** (TextBoxNomeCompleto):
    - Width: Fill parent...
    - Hint: Aparecerá aqui.

O *design* ficará conforme mostrado na Figura 1. Depois clique no botão **Design**, localizado próximo ao canto superior direito da ferramenta.

Para começar nossa programação, vamos colocar uma ação no evento do botão. Para isso, selecionamos em **Blocks** o componente chamado **ButtonConcatenar** e várias peças tipo quebra cabeça irão aparecer, encontramos a peça **When ButtonConcatenar.Click** e arrastamos ela para a área *Viewer*

Poderíamos fazer de forma bastante simples, fazendo com que o campo texto do nome completo recebe a concatenação direta do nome, com sobrenome e

com um espaço em branco entre eles, mas vamos complicar um pouquinho nosso exemplo para vermos um exemplo de criação e utilização de variável local.

Para criar uma variável local, vá em *Built-in*, clique em *Variables* e arraste a primeira peça escrita **initialize local name to** para dentro da peça do botão. Renomeie name para **aux**.

Precisamos inicializar essa variável com um valor, faremos isso inserindo um texto vazio. Então vá em *Built-in*, clique em *Text* e arraste a primeira pecinha para o encaixe da variável **aux**. Não mexa no conteúdo dessa pecinha de texto, ela é um texto vazio.

Agora faremos a concatenação dos campos de textos e colocaremos o resultado na variável **aux**. Então passe o mouse em cima no nome **aux** que está na pecinha da variável, irá aparecer duas pecinhas, arraste a pecinha set e coloque a dentro da pecinha da variável.

**Observe** que a variável **aux**, como sendo variável local, só será visível e utilizável pelas pecinhas que estão dentro dela. Para que você possa ter mais variáveis locais, junto com essa que criamos, basta clicar no ícone de ferramenta da peça da variável e ali é possível inserir mais variáveis locais.

Vamos inserir uma peça **join** para fazer a concatenação. Para isso, vá em *Built-in*, clique em *Text* e arraste a peça **join** para o encaixe do **set aux** que acabamos de colocar. Contudo, vamos precisar de três espaços e a peça só tem dois. Vai ficar Nome + espaço em branco + Sobrenome. Então clique na ferramentazinha do **join** e adicione mais um elemento, nosso **join** vai ficar com três encaixes.

Agora vamos colocar os conteúdos dos encaixes do **join**. Na coluna *Blocks* da nossa interface, vá em *Screen1*, clique no **TextBoxNome** e arraste a pecinha **get Text**, veja que não tem a palavra **get**, mas tem uma saliência para encaixar.

Depois coloque no segundo encaixe um espaço em branco. Basta copiar e colar a peça de texto que já tem no código, encaixar essa nova peça no **join**, clicar dentro dela e digitar apenas um espaço em branco.

Para o último elemento, copie e cole a primeira peça encaixada no **join**, apenas altere o nome dela de **TextBoxNome** para **TextBoxNomeCompleto** e coloque ela no terceiro encaixe do **join**.

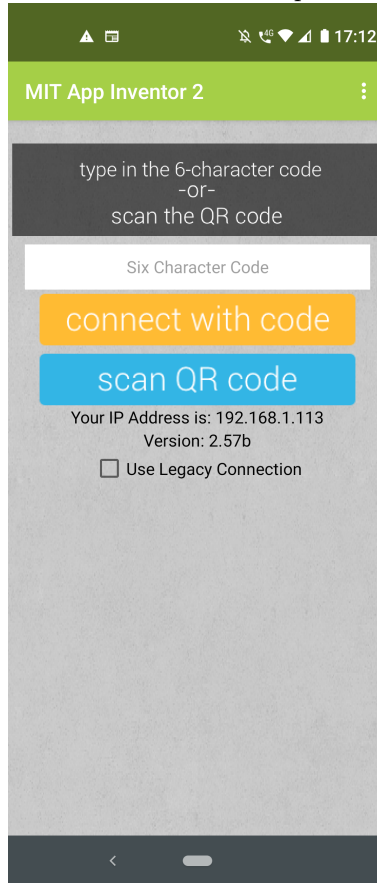
Depois, seguindo esse padrão, pegamos o **set TextNomeCompleto** e colocamos um **get aux** da nossa variável ali no encaixe.

Opcionalmente, coloque um set nos campos Nome e Sobrenome com texto vazio para *resetar* nossos campos.

### 3.1 Execução

Para executar, baixe o aplicativo **AI Companion** (Figura ??fig:aicompanion) da *Google Play Store* no seu Android e o execute. Na ferramenta web, vá no menu superior, clique em **Connect** e selecione **AI Companion**, irá aparecer um código de barras na tela. No aplicativo do seu dispositivo móvel clique no botão **scan QR code**, aponte para o código na tela e o aplicativo irá rodar no seu Android.

Figura 3: Interface do AI Companion: *Blocks*



Usando esse método, as alterações feitas na ferramenta são sincronizadas com seu aplicativo em tempo de execução, sendo bastante útil para debug. Contudo, é necessário que o computador e o dispositivo móvel estejam conectados na mesma rede wifi.

Para instalar, ao invés de apenas executar, vá em **Build** e selecione **App (provide QR code for .apk)**, use qualquer leitor de QR Code e baixe e instale o apk no seu Android.

## Referências

[MI2] *With MIT App Inventor anyone can build apps with global impact.*  
<<https://appinventor.mit.edu/>>, acessado em Março 2020.