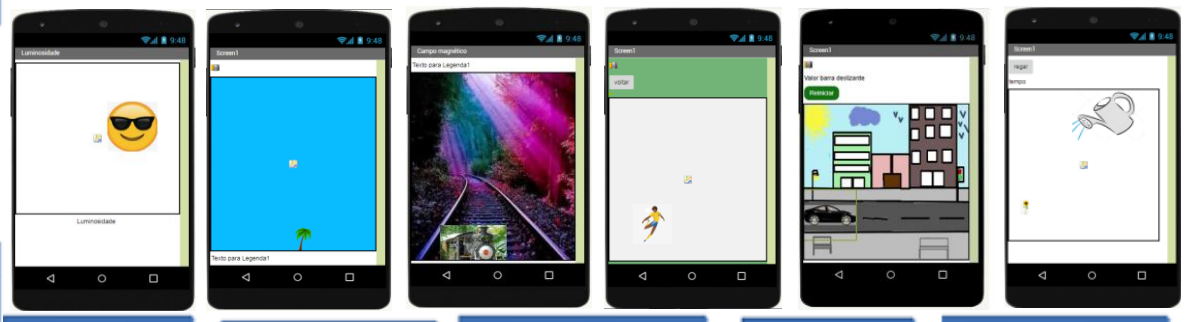
UNIVERSIDADE FEDERAL DO RIO GRANDE – FURG
Programa de Pós-Graduação em Ensino de Ciências Exatas

Atividades interativas com a plataforma App Inventor

Estudando a função de 1º grau através do desenvolvimento de aplicativos para *smartphones*

Profª Adriana Dada de Andrade

Orientador: Prof. Dr. Luciano Silva da Silva



Ficha Catalográfica

A553a Andrade, Adriana Dada de.

Atividades interativas com a plataforma *App Inventor*: estudando a função de 1º grau através do desenvolvimento de aplicativos para *smartphones* [Recurso Eletrônico] / Adriana Dada de Andrade. – [Santo Antônio da Patrulha, RS]: FURG, [2021].

41 f. : il. color.

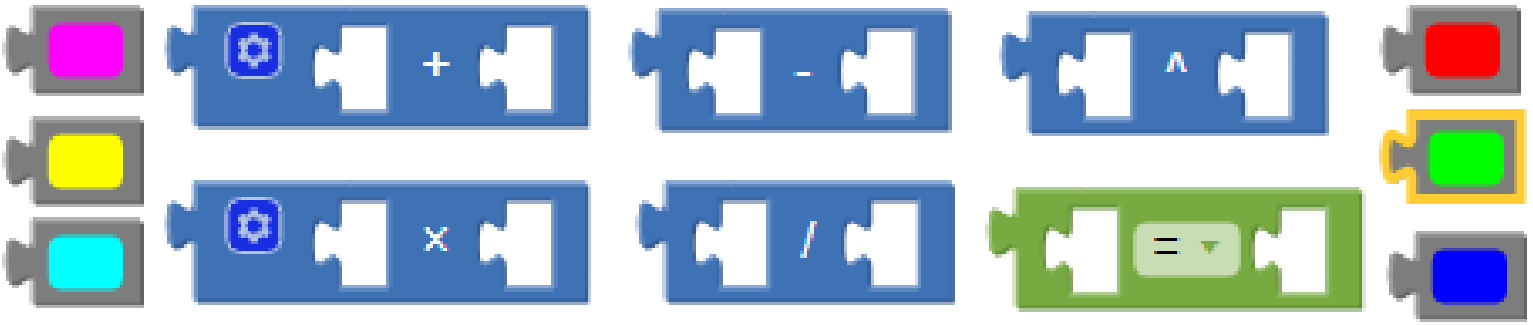
Produto Educacional da Dissertação de mestrado do Programa de Pós-Graduação em Ensino de Ciências Exatas, para obtenção do título de Mestre em Ensino de Ciências Exatas, sob a orientação do Dr. Luciano Silva da Silva.

Disponível em: https://ppgece.furg.br/
http://repositorio.furg.br/

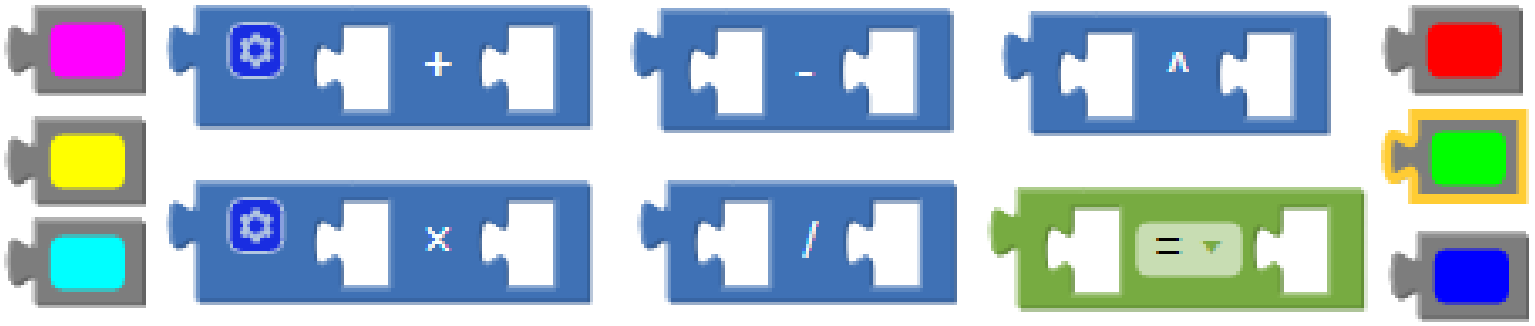
1. Ensino de Matemática 2. *App Inventor* 3. Construcionismo 4. Pensamento Computacional I. Silva, Luciano Silva da II. Título.

CDU 37:51

Catalogação na Fonte: Bibliotecário José Paulo dos Santos CRB 10/2344



Conteúdo



1. Apresentação

O presente guia de atividades teve sua origem no trabalho de Mestrado cujo título é “Desenvolvimento de aplicativos com a plataforma App Inventor: Um complemento para o ensino de funções afins”. Este estudo foi realizado no Programa de Pós-Graduação de Ensino de Ciências Exatas, da Universidade Federal do Rio Grande, e procurou verificar como o uso de recursos tecnológicos digitais pode ajudar a desenvolver a autonomia dos estudantes no ensino de Matemática.

As atividades aqui propostas buscam aproximar os conteúdos da realidade dos estudantes, já que o uso dos *smartphones*, e seus aplicativos em particular, fazem parte do dia a dia destes estudantes. As atividades também têm como finalidade contribuir para a autonomia do estudante, uma vez que ele irá atuar como protagonista do processo, criando aplicativos de maneira intuitiva e personalizada.

Este material foi desenvolvido com base na teoria do Construcionismo de Seymour Papert. Para que essa teoria seja implementada, segundo Papert [3], é imprescindível que as atividades propostas sejam elaboradas de tal forma que o estudante possa construir seu conhecimento a partir da interação com o computador. Com essa perspectiva, foram propostas atividades que permitem que o estudante crie aplicativos de maneira lúdica e intuitiva, por meio da programação através de uma plataforma visual, nos quais os comandos são representados por blocos que se encaixam como peças de Lego.

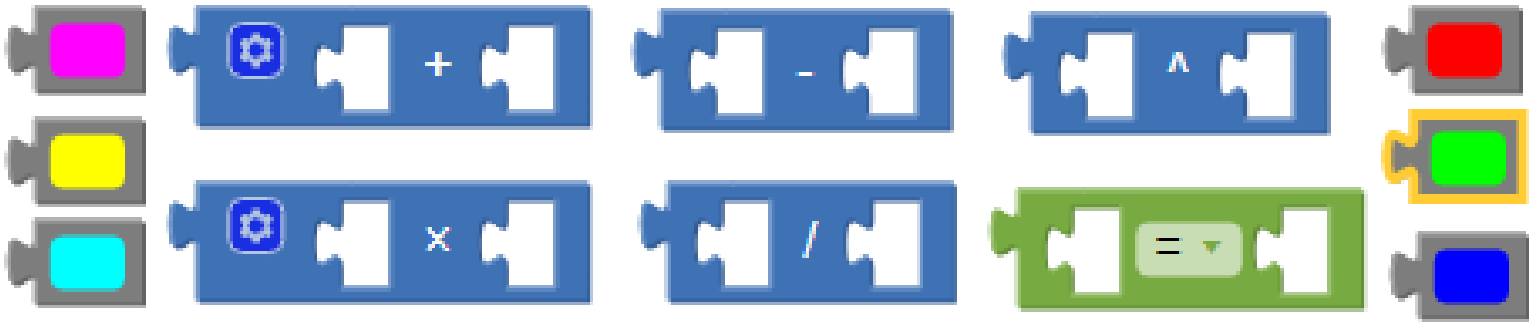
Além disso, este guia de atividades está em consonância com a Base Nacional Comum Curricular (BNCC), pois contribui para o desenvolvimento do pensamento computacional que “envolve as capacidades de compreender, analisar, definir, modelar, resolver, comparar e automatizar problemas e suas soluções, de forma metódica e sistemática, por meio do desenvolvimento de algoritmo” [1], sendo uma das habilidades a serem desenvolvidas nas aulas de Matemática. Ademais, também promove a utilização de tecnologias digitais, que é considerada uma das competências gerais da BNCC.

O conteúdo abordado nas atividades aqui propostas será a Função de 1º grau, envolvendo sistema de coordenadas cartesianas, lei de formação, domínio e contradomínio, função crescente e decrescente, tendo como público-alvo preferencial estudantes do nono ano do Ensino Fundamental.

Para o desenvolvimento das atividades será usada a plataforma App Inventor, que permite a

criação de aplicativos para *smartphones*, sendo necessário o uso de computadores e *smartphones* com acesso à Internet. Também é recomendável que o estudante já tenha uma noção sobre sistema de coordenadas cartesianas e lei da função de 1º grau.

O restante do texto está organizado da seguinte forma: o capítulo 2 apresenta a teoria Construcionista, que constitui a base teórica para o desenvolvimento deste trabalho. No capítulo 3 é apresentada a plataforma App Inventor. Um roteiro para o desenvolvimento de um aplicativo de calculadora simples é apresentado no capítulo 4. No capítulo 5 encontra-se a descrição de uma atividade que envolve o desenvolvimento de uma animação no App Inventor. No capítulo 6 é proposta uma atividade onde os próprios estudantes devem desenvolver um aplicativo. Por fim, o capítulo 7 apresenta as considerações finais.



2. Construcionismo

Nesse capítulo será apresentada a teoria do Construcionismo que foi desenvolvida por Seymour Papert e embasou o desenvolvimento das atividades aqui propostas.

Papert foi professor de Matemática e realizou muitos estudos sobre o uso dos computadores como uma ferramenta para facilitar os processos de ensino e aprendizagem. Ele foi o criador do LOGO na década de 60, um *software* em que o usuário controla movimentos de uma tartaruga na tela através de comandos específicos. A tartaruga funciona como uma caneta que produz traços na tela.

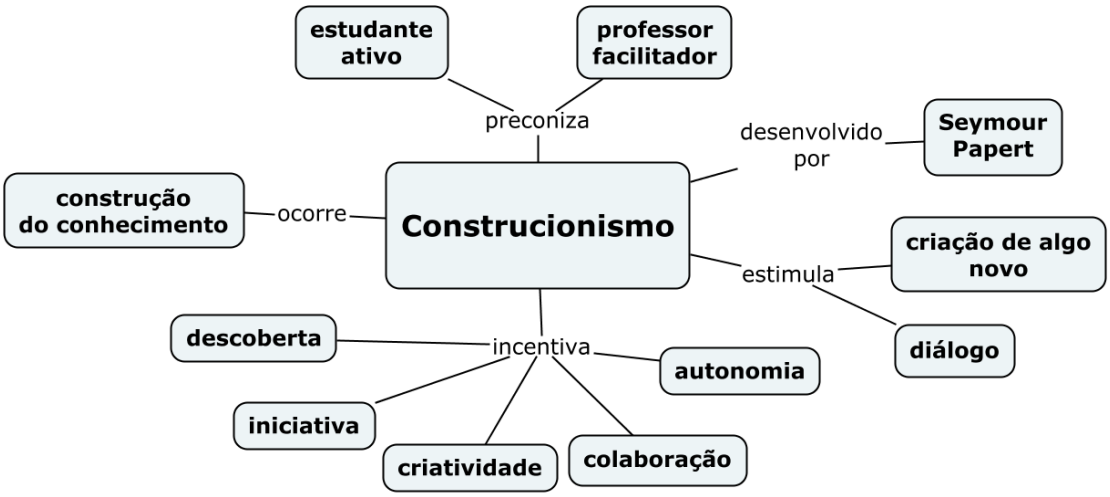
Em seu livro "LOGO: Computadores e Educação" Papert afirma que esse *software* foi desenvolvido para que as crianças aprendessem a programar o computador e construíssem os seus conhecimentos de forma lúdica e interativa. Segundo o autor a "atividade é tão variada, tão rica em descoberta, que mesmo no primeiro dia de programação o estudante pode fazer algo que é novo e excitante para o próprio professor" [2, p. 213] .

O computador é uma ferramenta com potencial para auxiliar nos processos de ensino e aprendizagem, mas segundo Papert [3] é recomendável que ele não seja usado apenas para fornecer informações aos estudantes (paradigma Instrucionista), mas sim em atividades que permitam ao estudante "criar algo novo" e, a partir dessa interação com o computador, tornar os conteúdos mais significativos; com esta meta ele propôs a teoria do Construcionismo.

Segundo a teoria do Construcionismo, o estudante deve ser um sujeito ativo, por consequência a "ênfase da educação deixa de ser a memorização da informação transmitida pelo professor e passa a ser a construção do conhecimento realizada pelo aluno de maneira significativa, sendo o professor o facilitador do processo" [6, p. 18]. Neste contexto, tanto o professor como o estudante desempenham papéis importantes nesses processos, e a autonomia, a iniciativa e a descoberta dos estudantes são incentivadas.

A figura abaixo[1] apresenta algumas características presentes no Construcionismo:

[1] Figura elaborada com base no trabalho de Scheller, Viali e Lahn [5]



Conforme a figura acima ilustra, um ambiente Construcionista também incentiva a colaboração e a criatividade dos estudantes e a aprendizagem acontece "num contexto de descobertas, de experiências e novos contatos motivados pelo diálogo, em um ambiente propício" [5, p. 5]. Assim, a aprendizagem ocorre por meio do uso do computador de maneira dinâmica, estimulando o diálogo com seus colegas e professores.

Ao programar o computador, são disponibilizadas "condições para o aluno descrever a resolução de problemas, usando a linguagem de programação, refletir sobre os resultados obtidos e depurar suas ideias por intermédio da busca de novos conteúdos e novas estratégias" [6]. Assim, muitas ações são realizadas e ocorre o ciclo, descrito pelo autor, como **descrição** → **execução** → **reflexão** → **depuração** → **descrição** (novamente) → ...

Na etapa de **descrição**, o estudante descreve as etapas necessárias para resolver um problema através de uma linguagem computacional. A **execução** ocorre quando o computador executa os comandos e mostra o resultado. Na **reflexão**, o estudante avalia a resposta mostrada pelo computador, verificando se atende às expectativas. No processo de **depuração**, ele busca identificar eventuais erros ou aspectos considerados insatisfatórios na programação, e que podem ser corrigidos ou melhorados.

Ao verificar que existem pontos a serem corrigidos ou melhorados, o estudante inicia novamente o processo de descrição, execução, reflexão e depuração. Este ciclo se repete até que o estudante esteja satisfeito com o resultado obtido. Nesse processo, o erro apresenta um caráter construtivo, pois o aluno aprende com ele.

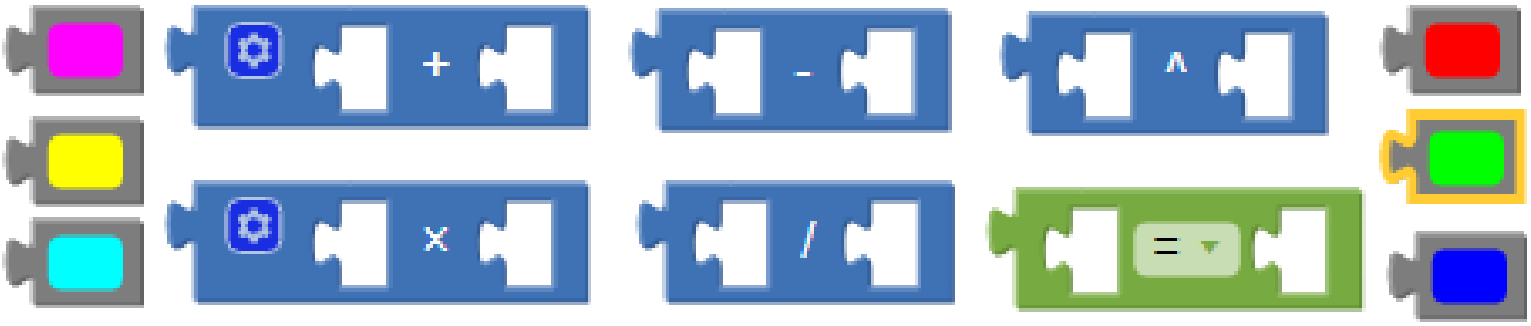
A plataforma App Inventor, utilizada neste trabalho, permite que os estudantes realizem esse ciclo por meio do desenvolvimento de aplicativos. Contudo, Valente [6] ressalta a importância da atuação do professor como mediador nesse processo, para que esse ciclo ocorra de forma satisfatória.

Papert também estabeleceu cinco dimensões que caracterizam os ambientes construcionistas [4]:

a) **Dimensão pragmática**: se refere ao conhecimento ter uma finalidade prática e possibilitar a criação de "algo novo";
b) **Dimensão sintônica**: evidencia a relação dos conteúdos com a vida real dos estudantes;
c) **Dimensão sintática**: facilidade ao interagir com o ambiente;
d) **Dimensão semântica**: as ferramentas do ambiente apresentam significado para os estudantes;
e) **Dimensão social**: diz respeito à interação social e cultural dos estudantes.

O próximo capítulo apresenta a plataforma App Inventor, que é utilizada como um ambiente Construcionista para as atividades propostas neste trabalho. O emprego do App Inventor nas ativi-

dades propostas atende as cinco dimensões dos ambientes Construcionistas, tal como estabelecido por Papert: atende a dimensão pragmática, pois desenvolve nos estudantes a habilidade de criar aplicativos, que podem ser utilizados para resolver problemas reais; atende a dimensão sintônica, uma vez que *smartphones* e aplicativos estão cada vez mais presentes no cotidiano dos estudantes; atende a dimensão sintática, pois é uma plataforma relativamente simples e intuitiva; atende a dimensão semântica, ao permitir que o estudante relacione os recursos da plataforma e o conteúdo que está sendo abordado durante as atividades com outros aplicativos que ele usa no seu dia a dia; e atende a dimensão social, pois o estudante vai aprimorar o seu entendimento a respeito de aplicativos e *softwares* que tem grande relevância na sociedade atual.



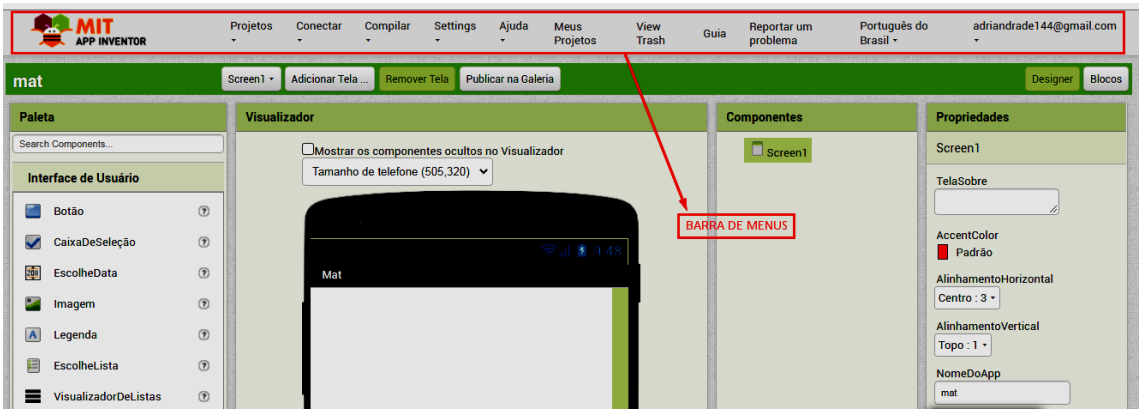
3. Apresentando o App Inventor

O App Inventor é uma plataforma de criação de aplicativos móveis originalmente desenvolvida pelo Google, e atualmente mantida pelo MIT –*Massachusetts Institute of Technology*. O App Inventor foi concebido com o objetivo de permitir que pessoas com pouca ou nenhuma experiência em programação de computadores possam criar aplicativos de forma rápida, através de uma interface em que o usuário programa arrastando e encaixando objetos gráficos.

3.1 Acessando a Plataforma

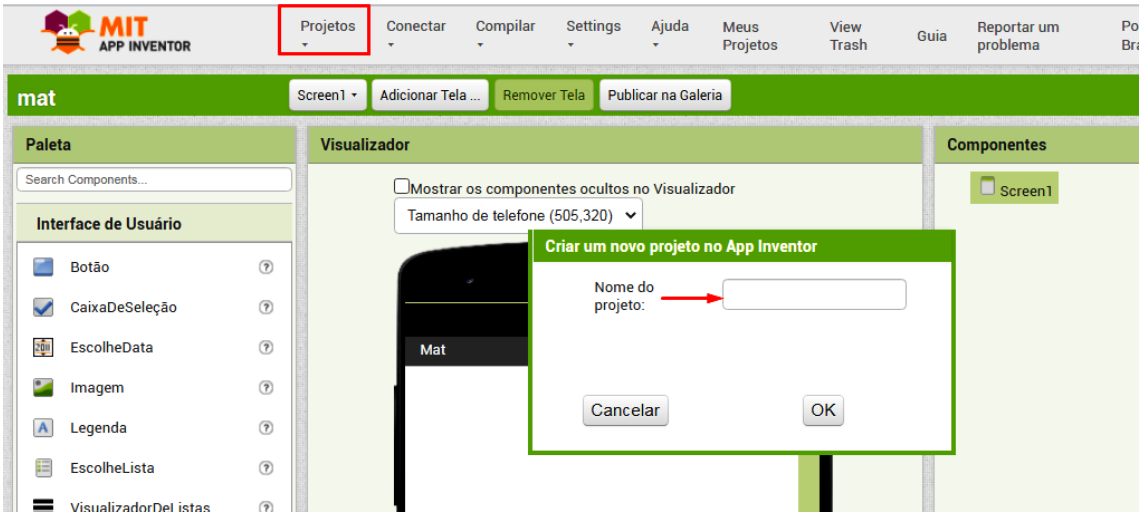
O acesso à plataforma App Inventor para a criação de aplicativos é realizada através de um navegador, no endereço eletrônico `http://appinventor.mit.edu`. Ao acessar este endereço clique em **Crie aplicativos**. Para efetuar o *login*, será solicitada uma conta de *e-mail* válida.

Ao acessar o App Inventor, podemos visualizar uma **Barra de menus** (observe a figura abaixo), que nos permite desenvolver um novo projeto, acessar projetos já criados, apagar projetos, executar os aplicativos que estão sendo criados no *smartphone*, emular o aplicativo no próprio computador, pedir ajuda e acessar a galeria com os projetos.



Para iniciar um novo projeto clique no menu **Projetos**, logo após **Iniciar novo projeto**. Será

solicitado o nome do projeto, conforme a figura abaixo. Digite o nome do projeto (sem espaços) e clique no botão **OK**.



O App Inventor oferece a opção de executar o aplicativo em desenvolvimento diretamente em *smartphones* com os sistemas operacionais Android ou iOS [1]. Para observar se os resultados estão ocorrendo conforme o planejado é necessário instalar previamente através da *Play Store* (para o sistema Android) ou *App Store* (para o sistema iOS) o aplicativo **MIT AI2 Companion** no seu *smartphone*. Após instalado, clique no ícone do aplicativo para ele iniciar

Os aplicativos podem ser executados no *smartphone* por meio de conexão por um cabo USB ou por conexão com a Internet com a leitura de um código QR. Para a segunda opção, no menu **Conectar** escolha a opção **Assistente AI**. Utlizando o aplicativo instalado previamente no *smartphone*, clique em **scan QR code** e escaneie o código QR criado no computador (conforme a figura abaixo), ou preencha o campo **Six Character Code** com o código de 6 caracteres correspondente.

Você vai visualizar seu aplicativo no *smartphone* e à medida que for acrescentando outros componentes no aplicativo, vai observar simultaneamente no seu *smartphone* a aparência e a funcionalidade dos mesmos no aplicativo em desenvolvimento.

[1] Até a presente data (julho de 2021), a utilização dos sensores pelo App Inventor apresenta problemas de compatibilidade no sistema iOS, o que pode prejudicar o bom funcionamento das atividades propostas nos capítulos 5 e 6.

O App Inventor é composto por duas janelas de edição. A primeira é chamada de *Designer* e por meio dela cria-se o *layout* do aplicativo. Essa tela é composta por **Paleta**, **Visualizador**, **Componentes**, **Propriedades** e **Mídias**:

Na **Paleta** encontram-se os componentes que serão usados para criar a aparência do aplicativo. As principais categorias em que estão divididos esses componentes são:

Interface do Usuário: componentes que proporcionam a interação com o aplicativo como, por exemplo, botões, legendas e caixas texto;

Organização: componentes que nos permitem organizar as posições de outros elementos no aplicativo para fiquem como uma boa apresentação. Por exemplo: organização horizontal e vertical;

Mídia: componentes que permitem o uso de vídeos e sons. Por exemplo: gravador, câmera;

Desenho e animação: componentes que permitem criar animações no aplicativo por meio de Pintura (a janela onde ocorre a animação), Bola (um círculo simples) e *Spritelmagem* (figura que pode ser carregada a partir de um arquivo de imagem);

Maps: componentes que permitem incluir mapas no aplicativo;

Sensores: componentes que identificam e respondem a algum estímulo, por exemplo, Sensor Acelerômetro, Temporizador e Sensor de Luz.

Social: componentes que possibilitam ao aplicativo uma interação social, por exemplo: Compartilhamento e Ligação.

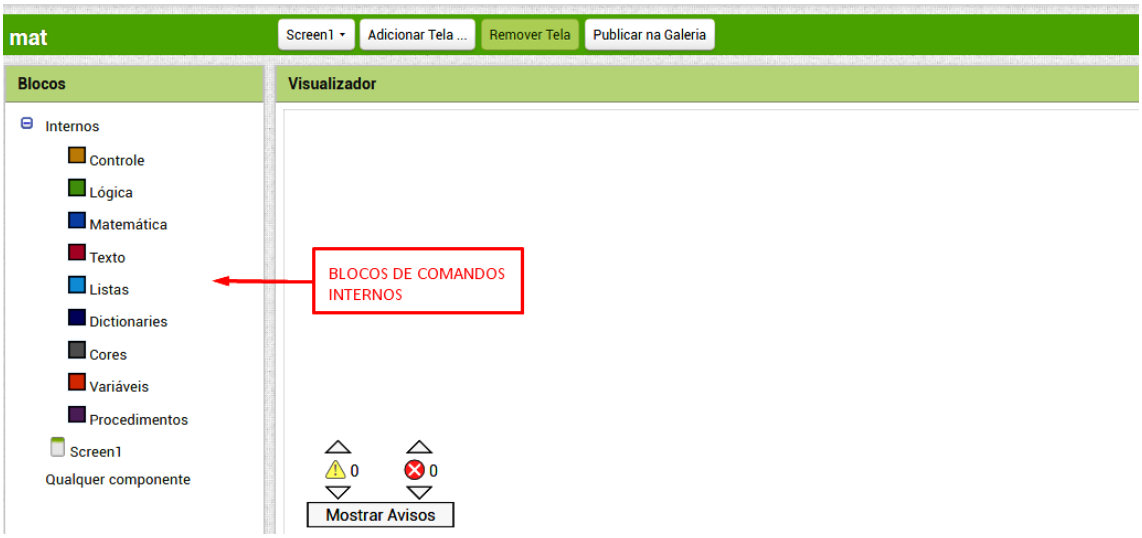
Armazenamento: componentes que armazenam os dados. Exemplo: Arquivo.

O **Visualizador** simula a tela do *smartphone*, é o local onde serão posicionados os componentes usados para criar o *layout* do aplicativo, como por exemplo: botões, legendas e imagens.

Em **Propriedades** é exibida uma lista ordenada dos componentes usados no desenvolvimento do aplicativo, sendo possível apagar ou renomear os componentes. Também é possível alterar características dos componentes usados na criação do aplicativo, como por exemplo, mudar a cor de fundo, a fonte, escrever algum texto, etc.

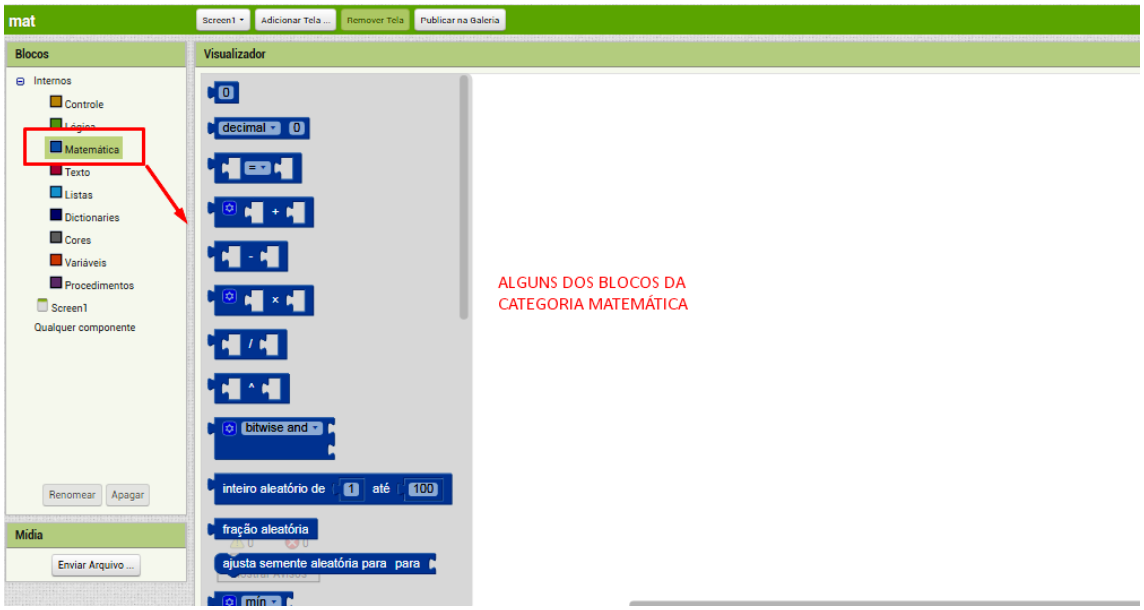
Na outra janela de edição do App Inventor será realizada a programação do aplicativo, usando os componentes do *layout*. Ela é chamada de **Blocos**. Para acessá-la é necessário clicar no botão **Blocos** que fica no canto superior esquerdo da tela:

Esta janela de edição apresenta blocos de comandos no lado esquerdo:

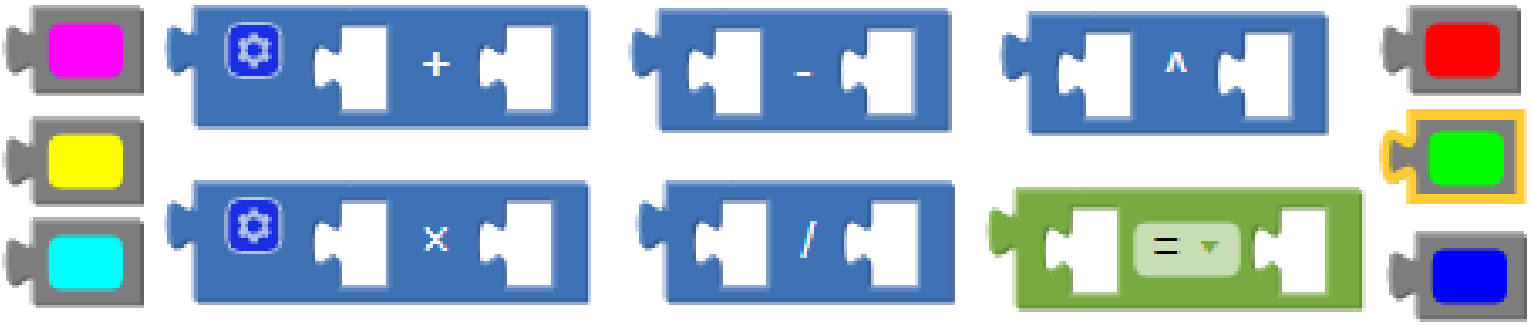


O menu dos blocos de comandos **Internos** está dividido nas categorias **Controle**, **Lógica**, **Matemática**, **Texto**, **Listas**, ***Dictionaries***, **Cores**, **Variáveis** e **Procedimentos**. Abaixo destes ficam os de blocos de comandos específicos para cada componente incluído no *Designer*.

Cada bloco representa um comando a ser executado pelo aplicativo, por exemplo, testar determinadas condições, realizar operações matemáticas, escrever um texto, criar uma lista, etc. Ao clicar em uma categoria qualquer de blocos, é aberta uma janela com as opções, assim o usuário pode escolher os blocos necessários para realizar sua programação. Por exemplo, algumas opções de blocos da categoria **Matemática** são mostradas abaixo:



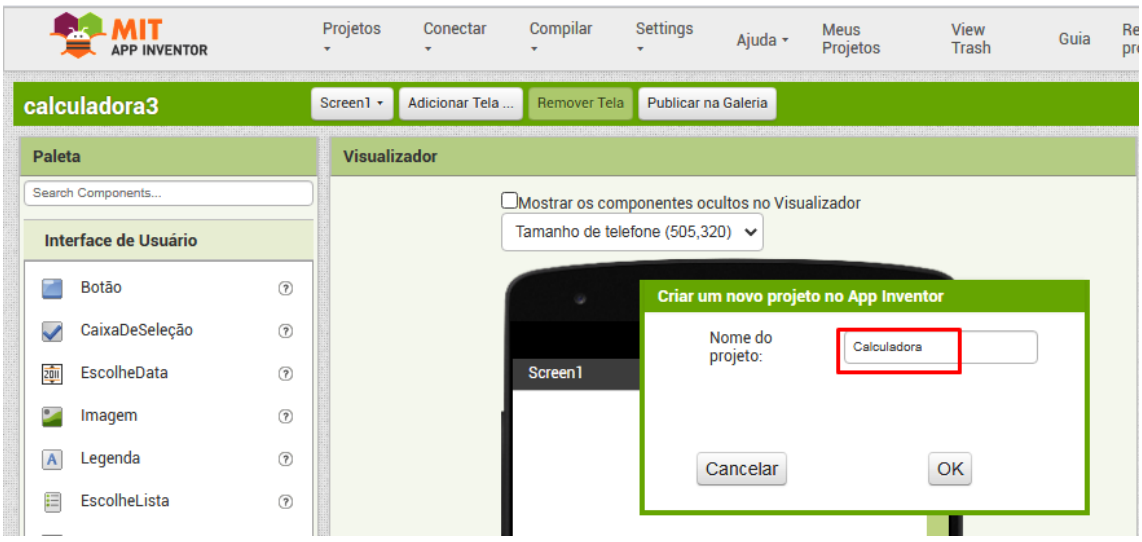
Para fazer a programação, o usuário arrasta os blocos escolhidos para o **Visualizador** e vai encaixando-os de forma a representar o encadeamento das ações desejadas.



4. Criando uma calculadora simples

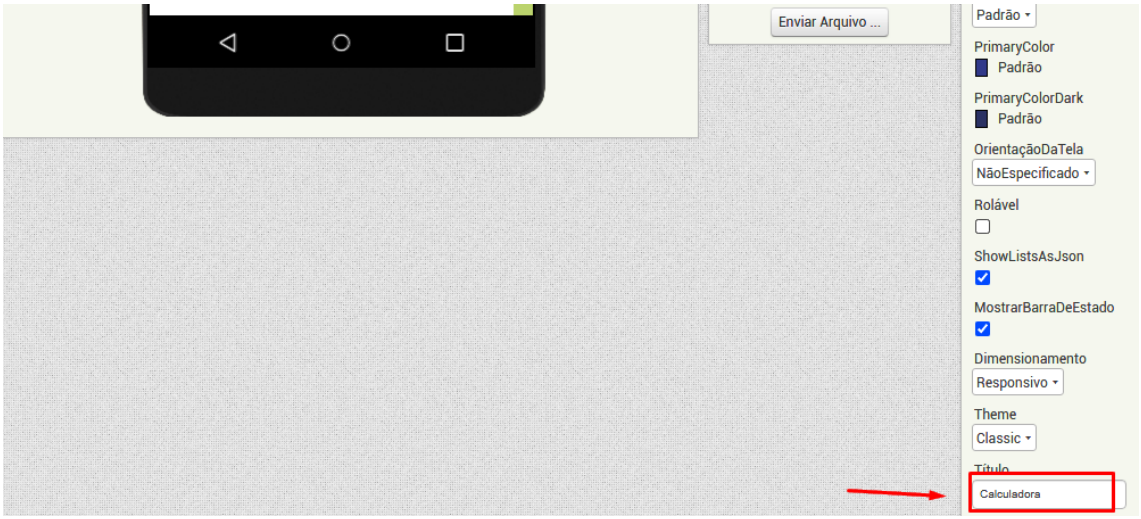
Neste capítulo será apresentado o desenvolvimento de um aplicativo de calculadora. Os objetivos desta atividade são apresentar a plataforma App Inventor, suas principais funcionalidades e interagir com esse ambiente.

Para começar, clique no menu **Projetos**, depois **Iniciar novo projeto**, escreva o nome do aplicativo e clique em **OK**:



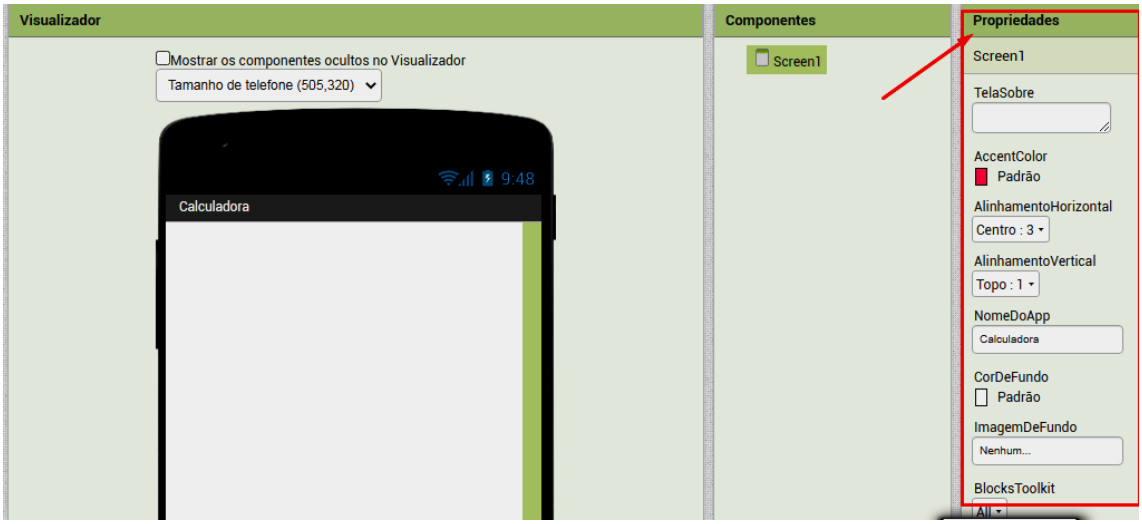
4.1 Criando a interface

O App Inventor permite criar aplicativos com diversas telas. Ao criar um novo projeto, uma tela vazia com o nome de **Screen1** é criada automaticamente. Por padrão, o nome da tela aparece no topo da tela do *smartphone*. Para mudar o nome da tela, clique no componente **Screen1**, e na lista de **Propriedades** altere o campo **Título** para **Calculadora**:

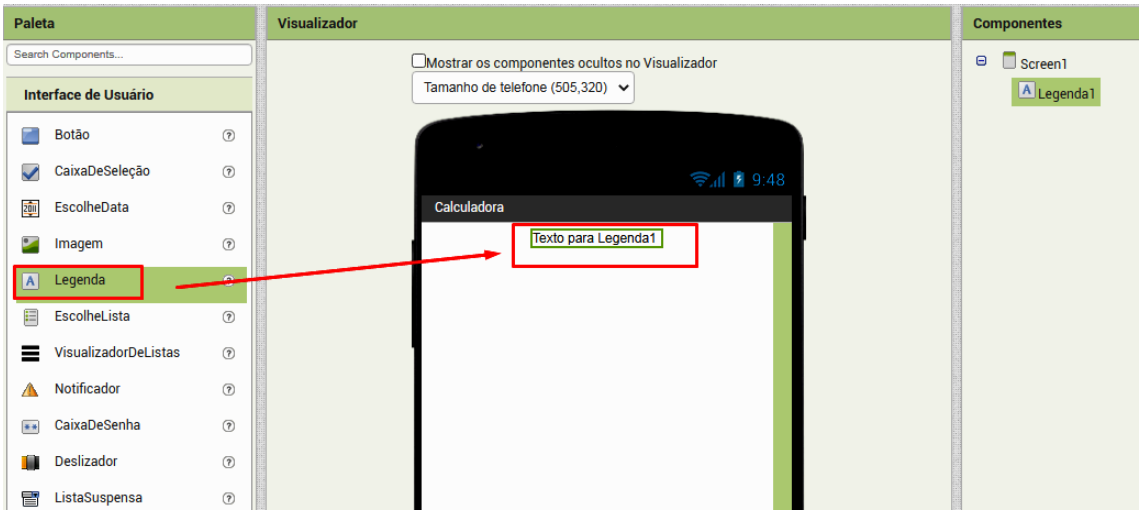


É desejável que durante a programação você conecte o *smartphone* ao **Assistente AI** para acompanhar o comportamento dos componentes do aplicativo.

Para uma melhor organização dos componentes usados no *layout* do aplicativo, nas **Propriedades** da tela, ajuste o **AlinhamentoHorizontal** no centro e o **AlinhamentoVertical** no topo. Você também pode mudar a cor do fundo da tela, para isso clique em **CorDeFundo** e escolha a cor de sua preferência:



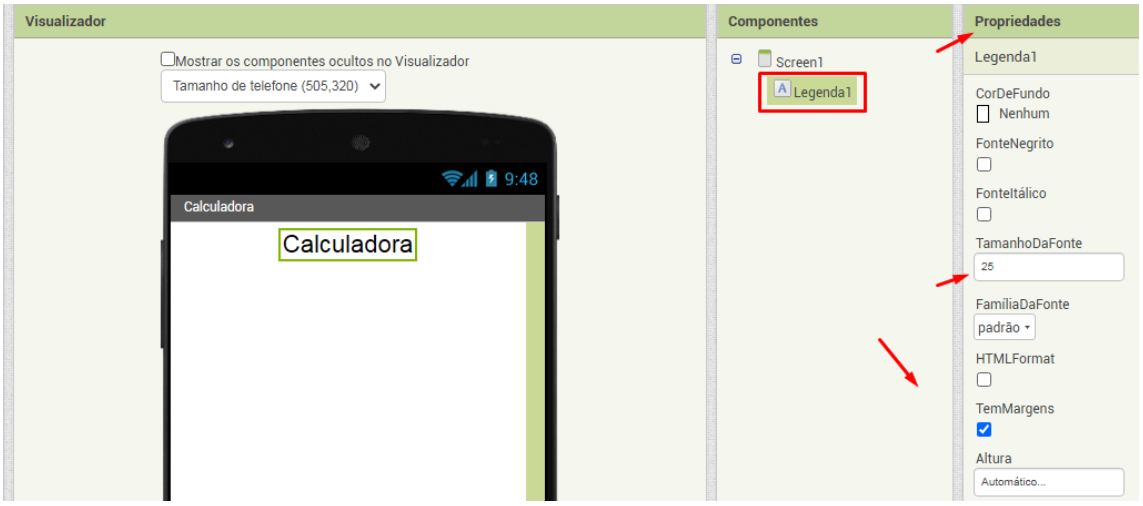
Para criar o *layout* do aplicativo, em **Paleta** - **Interface do usuário**, clique em **Legenda** e arraste para o **Visualizador**. A **Legenda** é um componente que permite apresentar qualquer tipo de texto ou mensagem no aplicativo. No aplicativo da Calculadora, esta legenda será utilizada para mostrar o nome do aplicativo.



Faça as alterações nas **Propriedades** da **Legenda**:

a) **TamanhoDaFonte**: 25
b) **Texto**: Calculadora
c) **AlinhamentoDoTexto**: Centro

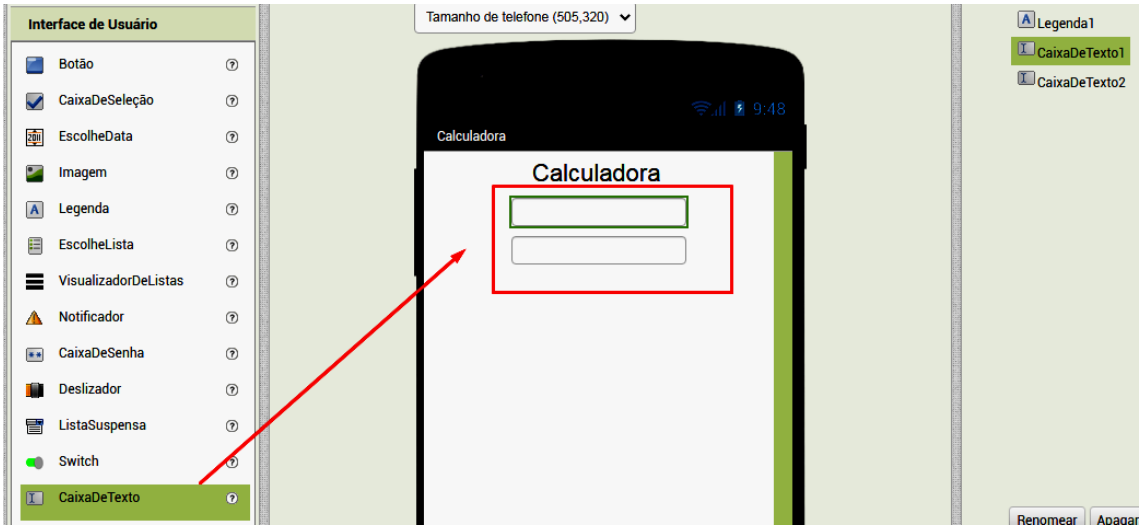
O aparência do aplicativo ficará como a figura abaixo:



Na **Paleta** - **Interface de Usuário**, clique e arraste duas caixas de texto (**CaixaDeTexto**). As caixas de texto são utilizadas para que o usuário digite alguma informação. No aplicativo da calculadora, o usuário vai inserir os operandos (números) da operação aritmética. Configure as seguintes propriedades da primeira caixa de texto:

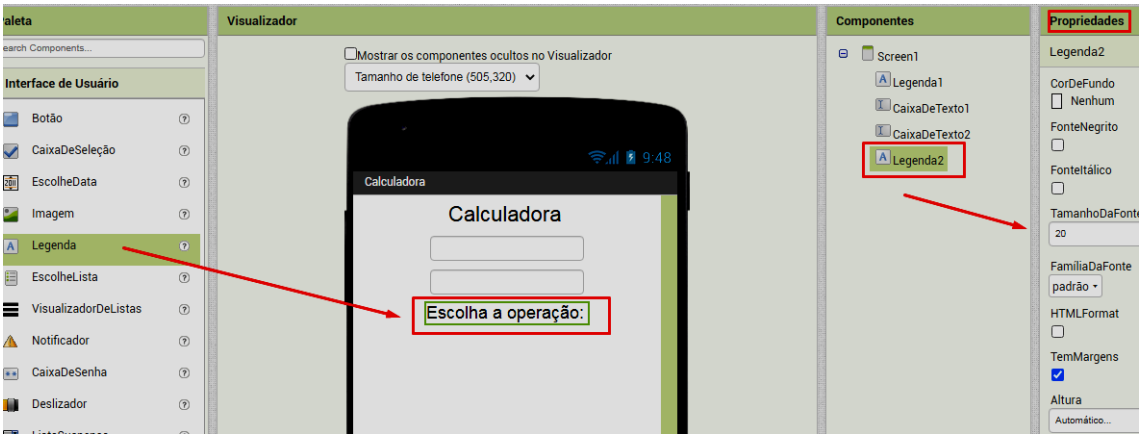
a) **TamanhoDaFonte**: 20
b) **Dica**: apague o texto existente
c) Marque a opção **SomenteNúmeros**
d) **Dica**: Caso use o sistema iOS é recomendável alterar as cores de fundo das caixas de textos. Para isso, clique na **CaixaDeTexto1** e, em **Propriedades** altere **CorDeFundo**.

Faça os mesmos procedimentos para configurar a segunda caixa de texto. Caso use o sistema iOS, sugere-se que sejam usadas cores diferentes no fundo das caixas de texto, assim destacamos ambos os campos de digitação de números, diferenciando um do outro. O *layout* do aplicativo ficará conforme a figura abaixo:



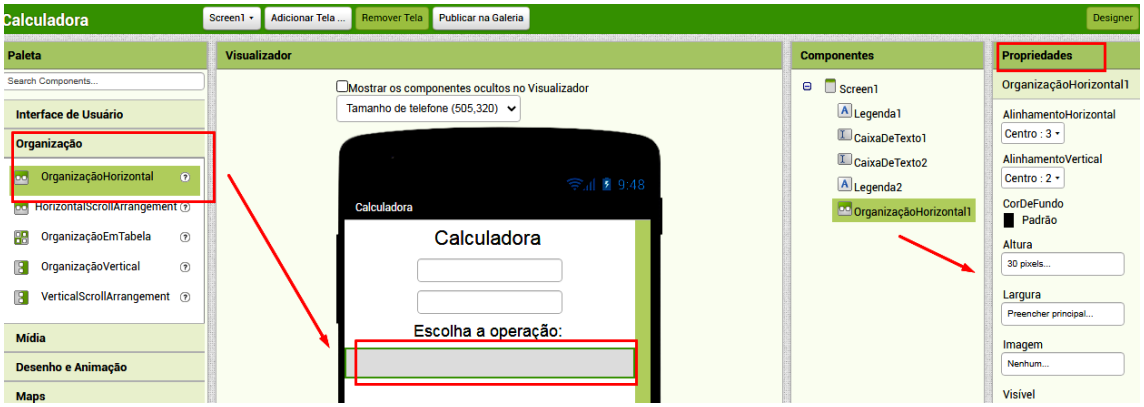
Acrescente uma legenda para informar ao usuário que ele, após digitar os valores, deverá escolher uma das operações aritméticas. Arraste uma **Legenda** para o **Visualizador** e faça as seguintes alterações em suas **Propriedades**:

a) Marque a opção **FonteNegrito**
b) **TamanhoDaFonte**: 20
c) **Texto**: Escolha a opção



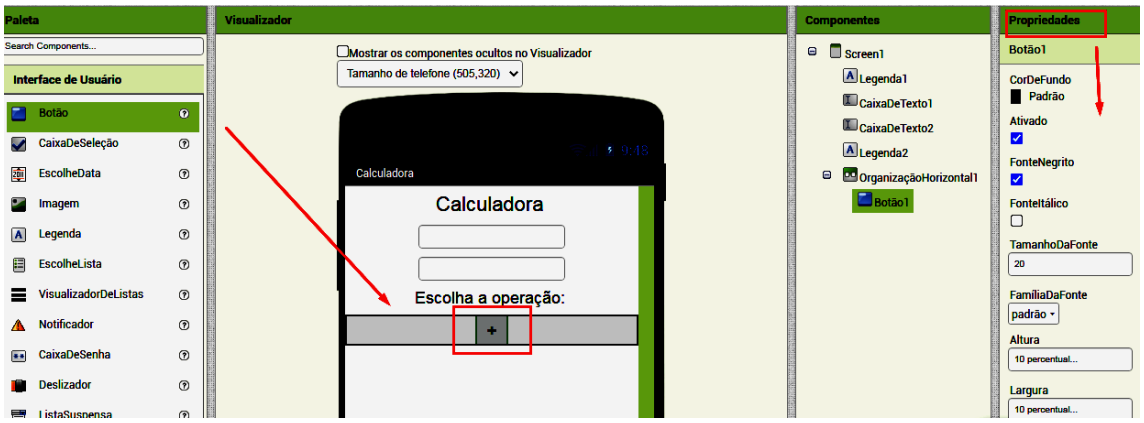
Para permitir que os botões que serão adicionados a seguir sejam colocados lado a lado, abra o menu **Organização** na **Paleta**, e arraste o componente **OrganizaçãoHorizontal** para o **Visualizador**. Configure as seguintes **Propriedades** deste componente:

a) **AlinhamentoHorizontal**: Centro
b) **AlinhamentoVertical**: Centro
c) **Altura**: 30 pontos
d) **Largura**: Preencher Principal

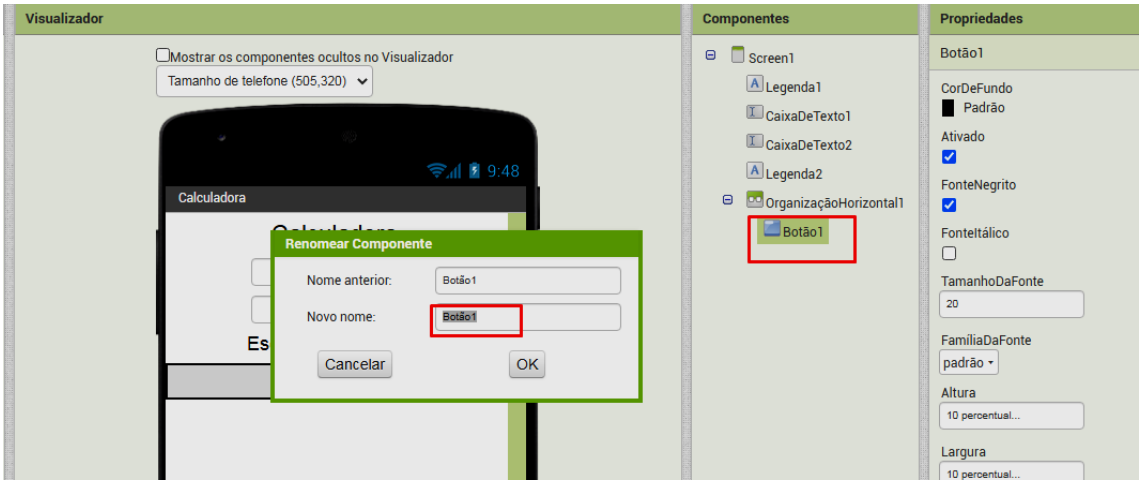


Os **botões** são componentes que podem ser clicados pelo usuário, e é possível atribuir comandos a serem executados a partir dos cliques, como veremos mais adiante. No menu de **Interface de Usuário** da **Paleta** arraste o componente **Botão** para o interior do retângulo da **OrganizaçãoHorizontal**, recém criada, e realize as seguintes mudanças nas **Propriedades**:

a) Marque a opção **FonteNegrito**
b) **TamanhoDaFonte**: 20
c) **Altura**: 10 percentagem
d) **Largura**: 10 percentagem
e) **AlinhamentoDoTexto**: centro
f) **Texto**: +

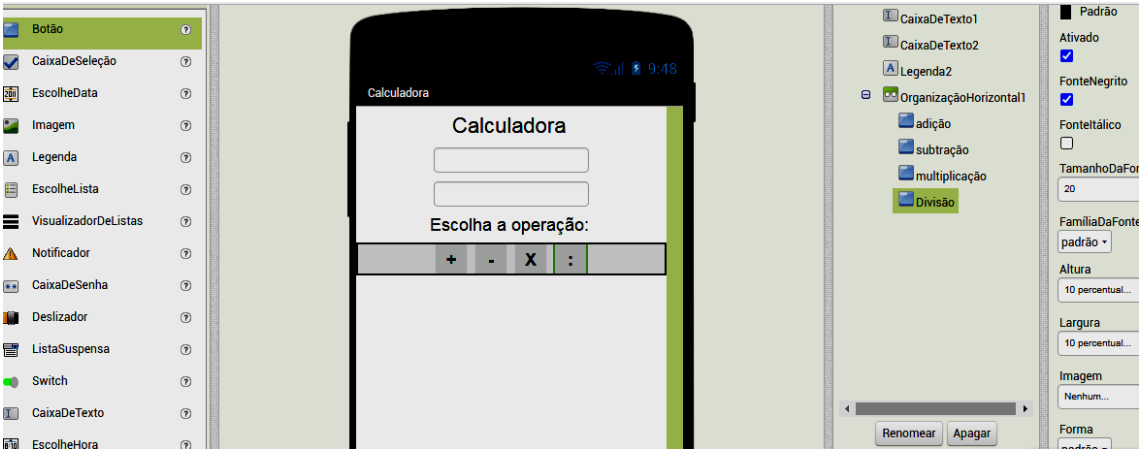


Para uma melhor organização dos componentes, vamos renomear este botão. Em **Componentes**, clique em **botão1** e a seguir em **Renomear**. Digite **adição** na janela que abrirá, e clique em **OK**:



Acrescente mais três botões, seguindo os mesmos procedimentos realizados com o botão **(+)**, para as demais operações aritméticas (subtração,multiplicação e divisão).

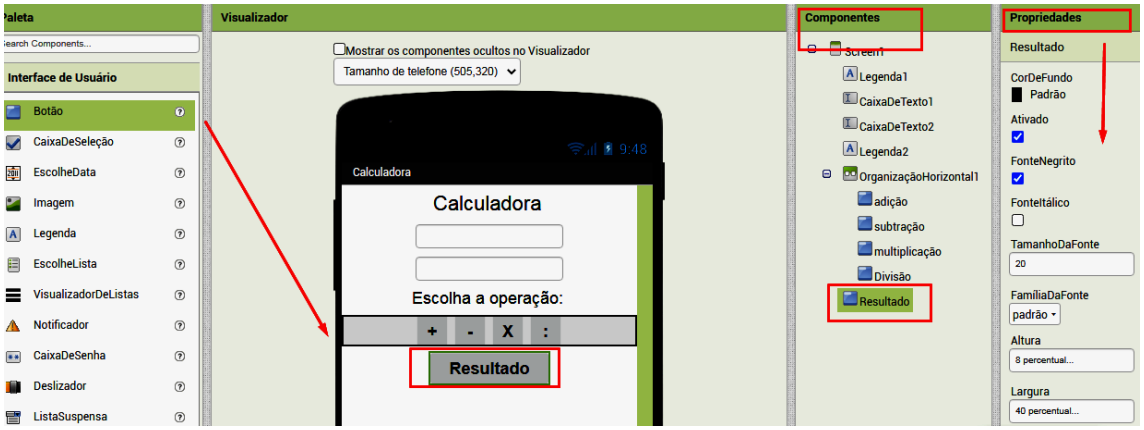
A tela do aplicativo deverá ficar como na figura abaixo:



Usaremos outra legenda para mostrar o resultado da operação realizada. Arraste outra **Legenda** para o **Visualizador** e altere as seguintes **Propriedades**:

a) Marque a opção **FonteNegrito**
b) **TamanhoDaFonte**: 20
c) **Altura**: 8 percentagem
d) **Largura**: 40 percentagem
e) **Texto**: resultado
f) **AlinhamentoDotexto**: centro

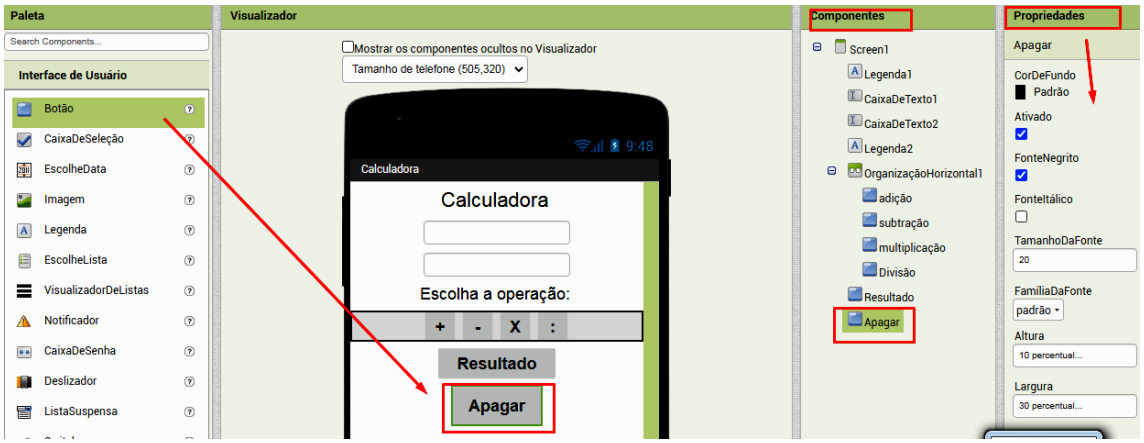
Renomeie esta **legenda** na lista de **Componentes**, dando a ela o nome de **Resultado**:



Para que após a realização de uma operação aritmética, não seja necessário apagar manualmente as caixas de texto, incluiremos um botão para executar esta função. Arraste um **Botão** para o **Visualizador**, e altera as **Propriedades**:

a) Marque a opção **FonteNegrito**
b) **TamanhoDaFonte**: 20
c) **Altura**: 10 percentagem
d) **Largura**: 30 percentagem
e) **Texto**: Apagar

Renomeie este botão com o nome **apagar**, da mesma forma que foi feito com a legenda de resultado.



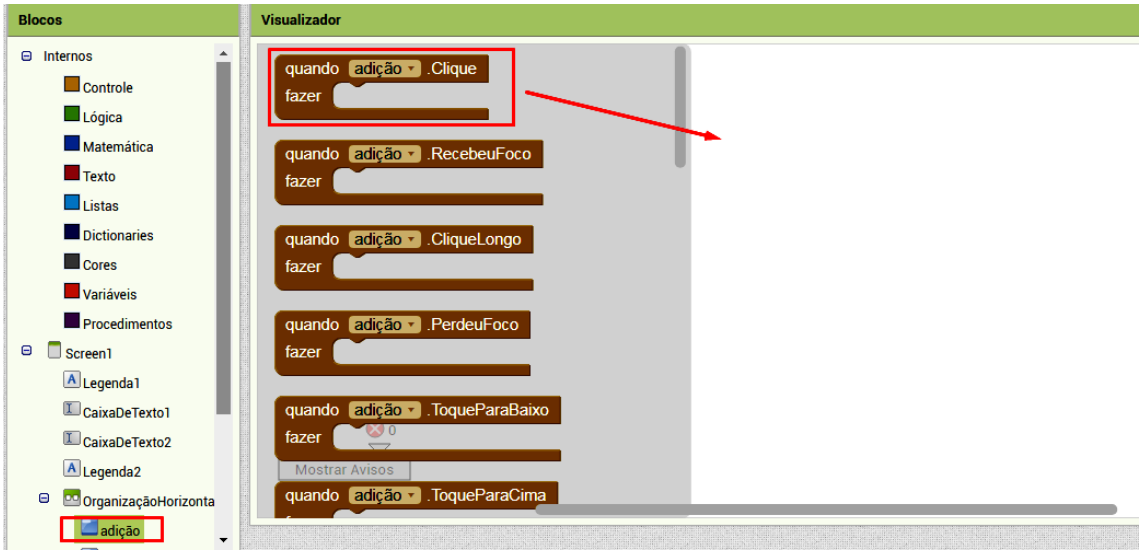
Assim, concluímos o *layout* básico do nosso aplicativo. Note que você pode personalizar as cores e tamanho de botões, fontes e legendas.

4.2 Programando o aplicativo

Agora, vamos iniciar a programação. Para acessar a janela de programação, clique no botão **Blocos** que está localizado no canto superior esquerdo da tela.

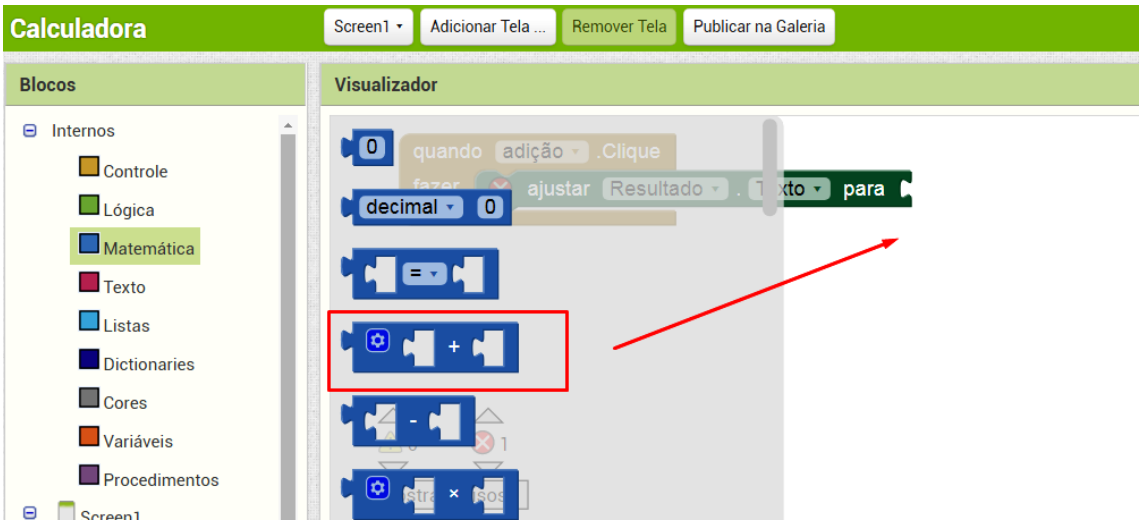
No menu de **Blocos**, à esquerda da tela, procure pelo componente do botão de **adição** e clique nele. Aparecerá na tela todas as possibilidades de blocos (comandos) que podem ser utilizados com este botão. Note que há três cores de blocos aqui: os blocos da cor laranja são usados para definir "gatilhos", ou seja, são blocos que "respondem" a alguma ação do usuário e executam comandos que podem ser encaixados dentro destes blocos; os blocos verde-claros representam valores de propriedades dos componentes, enquanto os blocos verde-escuros são comandos que alteram propriedades dos respectivos componentes.

Agora vamos fazer com que, ao clicar no botão **adição**, a calculadora realize a adição algébrica dos valores digitados nas caixas de texto. Para isso arraste o bloco **quando adição.Clique fazer** para o **Visualizador**:



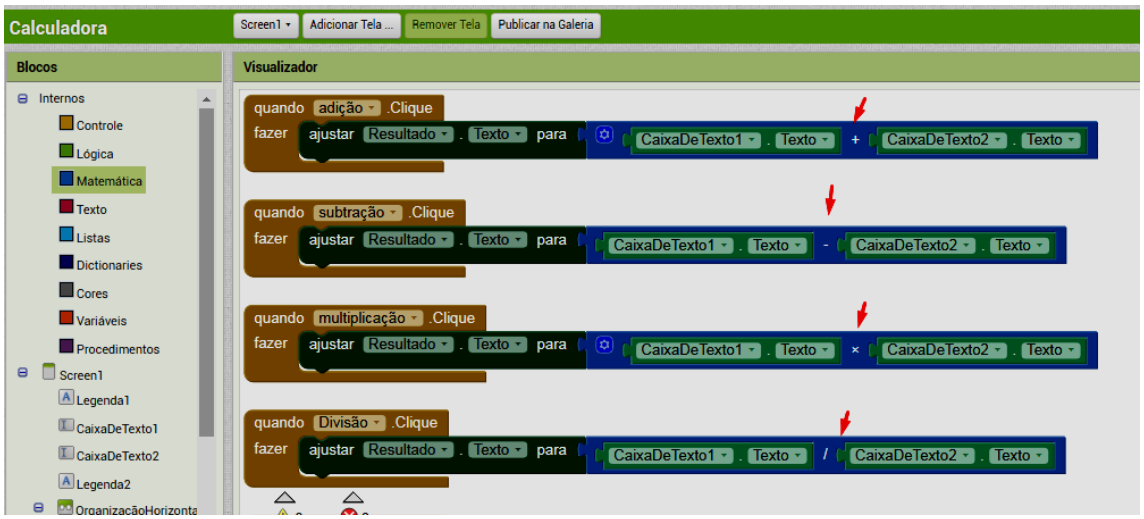
Para que as repostas das operações realizadas sejam exibidas na legenda de resultado clique no componente da legenda **resultado** no menu de **Blocos** e arraste o bloco **ajustar resultado.Texto para** para o **Visualizador**, encaixando-o dentro do bloco **quando clique.Fazer**:

No menu **Blocos**, clique em **Matemática**, e arraste o bloco que realiza a operação de adição:

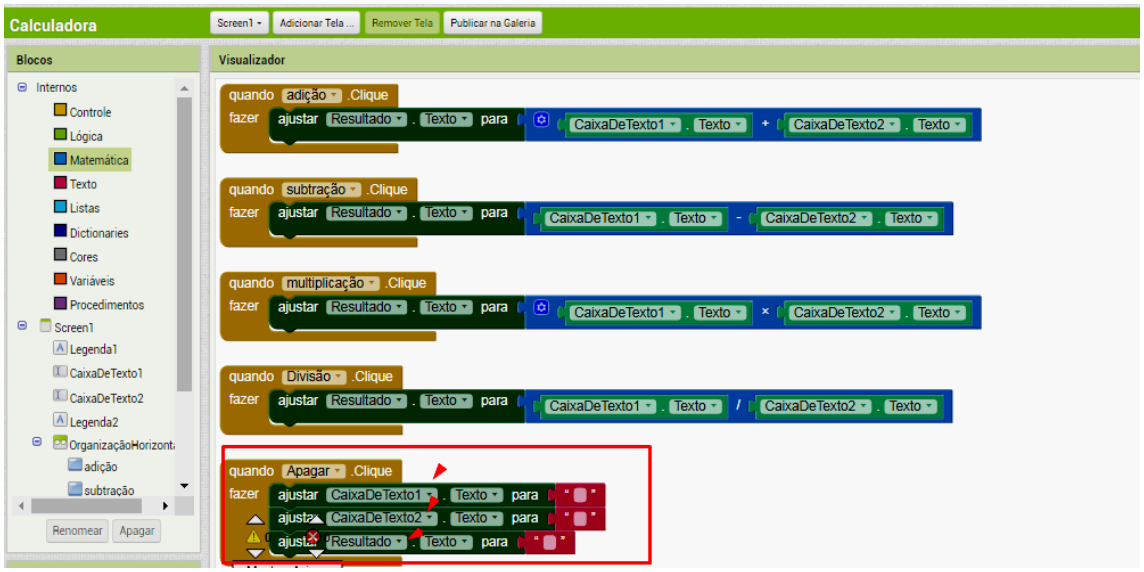


Selecione o componente **CaixaDeTexto1** no menu de **Blocos** e, em seguida arraste o bloco **CaixaDeTexto1.Texto** para o **Visualizador**, encaixando-o no primeiro espaço em branco do bloco da adição. Repita esse procedimento com a **CaixaDeTexto2**, colocando-a no interior do segundo espaço em branco. Estes blocos representam os respectivos valores das caixas de texto, ou seja, os valores que foram digitados.

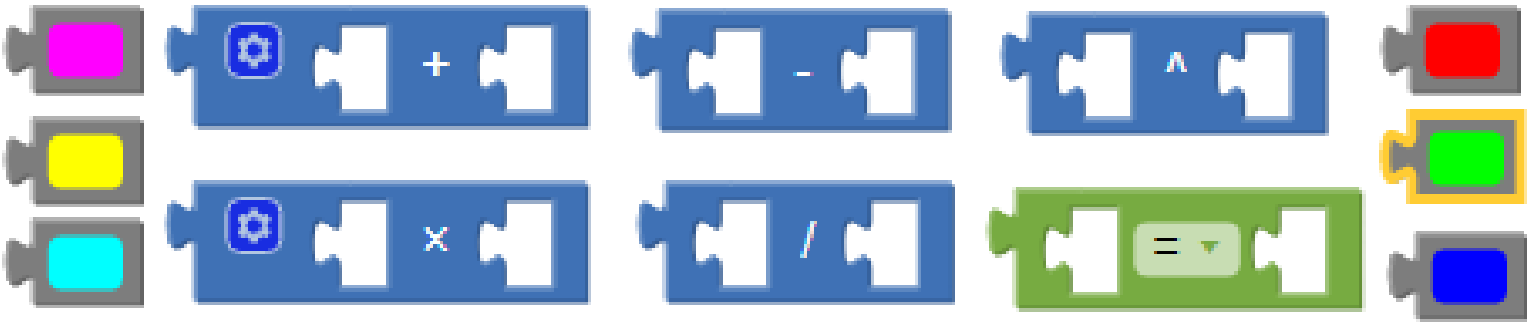
Agora, faça os mesmos procedimentos para os botões **subtração**, **multiplicação** e **divisão**. Para facilitar o processo, clique com botão direito do mouse nos blocos que deseja utilizar novamente e selecione duplicar:



Clique no componente **apagar** no menu de **Blocos** e arraste o bloco **quando Apagar.Clique fazer** para o **Visualizador**. Depois, clique no componente **CaixaDeTexto1** e arraste o bloco **ajustar CaixaDeTexto1.Texto** encaixando-o dentro do bloco **quando apagar.Clique fazer**. Finalmente, clique em **Texto** no menu de **Blocos** e arraste o primeiro bloco (**“ ”**) encaixando-o ao lado do bloco **ajustar CaixaDeTexto1.Texto para**. Esse procedimento deve ser feito também para a **CaixaDeTexto2** e a legenda **Resultado**, conforme a figura abaixo:



O seu aplicativo está concluído, não esqueça de verificar se todos os comandos estão funcionando conforme esperado. Essa verificação pode ser feita no seu *smartphone* por meio do aplicativo **Assistente AI**, conforme sugerido na seção 3.1. Note que, diferente de uma calculadora usual, onde se informa o primeiro operando, depois a operação, e por fim o segundo operando, nesta calculadora devem ser digitados os dois operandos primeiro e só depois clica-se na operação; esta ordem corresponde à **notação polonesa reversa**, ou **notação pós-fixada**.

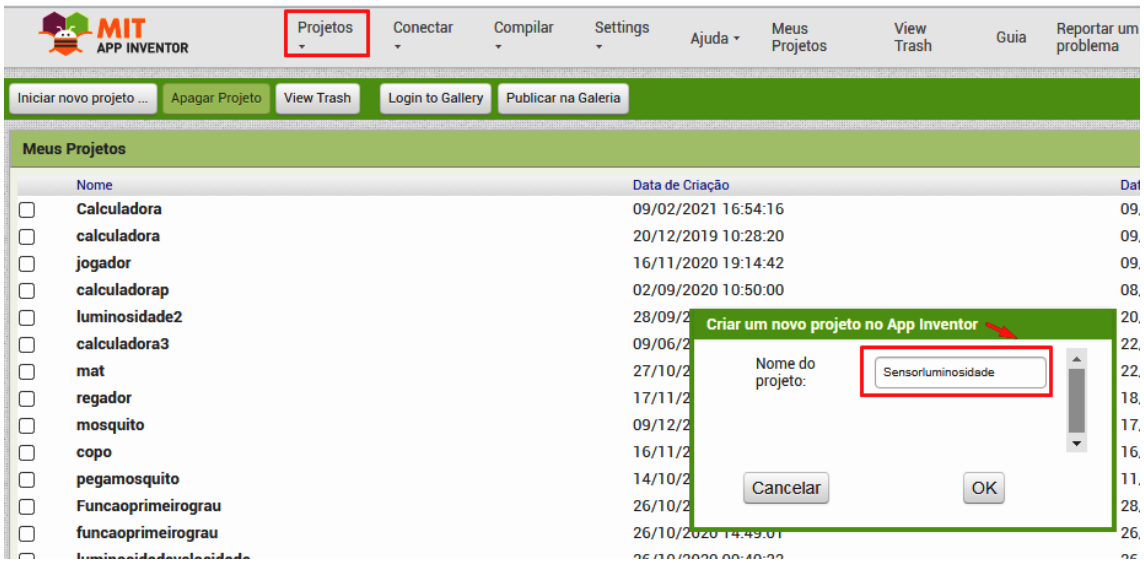


5. Criando a primeira animação

Neste capítulo será apresentada a criação de uma animação simples com os objetivo de possibilitar ao estudante a interação com os componentes de sensores, desenho e animação do App Inventor. O conhecimento prévio para realização da atividade é a lei de formação da Função de 1º grau $f(x) = ax + b$. Nesta atividade serão desenvolvidos os conteúdos: Função de 1º grau, domínio, contradomínio e imagem , função crescente e decrescente e sistema de coordenadas cartesianas.

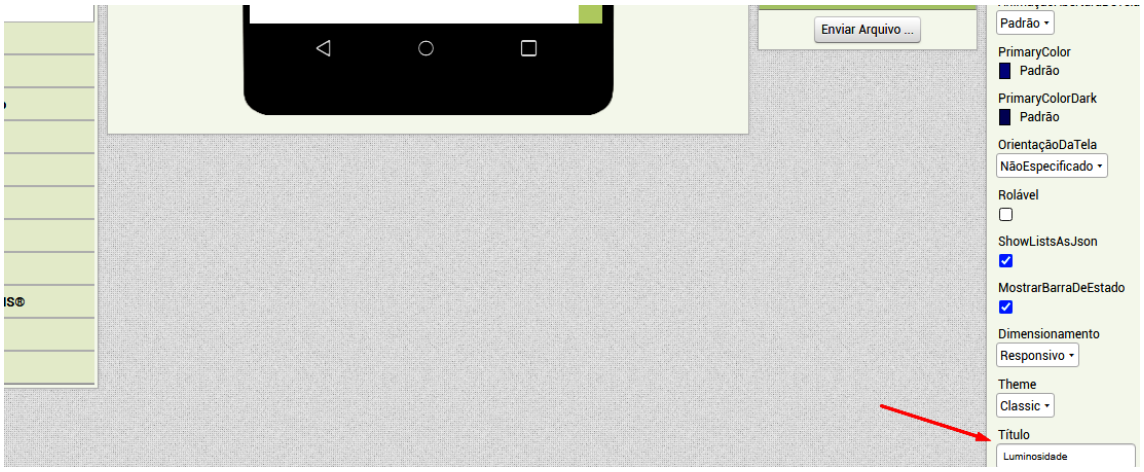
Para uma melhor organização dos componentes usados no *layout* da animação, nas **Propriedades** da tela, ajuste o **AlinhamentoHorizontal** no centro e o **AlinhamentoVertical** no topo.

Para iniciar a criação ao aplicativo, vá no menu **Projetos**, clique em **Iniciar novo projeto...** e escreva o nome do aplicativo, que chamaremos de **Sensorluminosidade** e clique em **OK**, conforme com a figura abaixo:

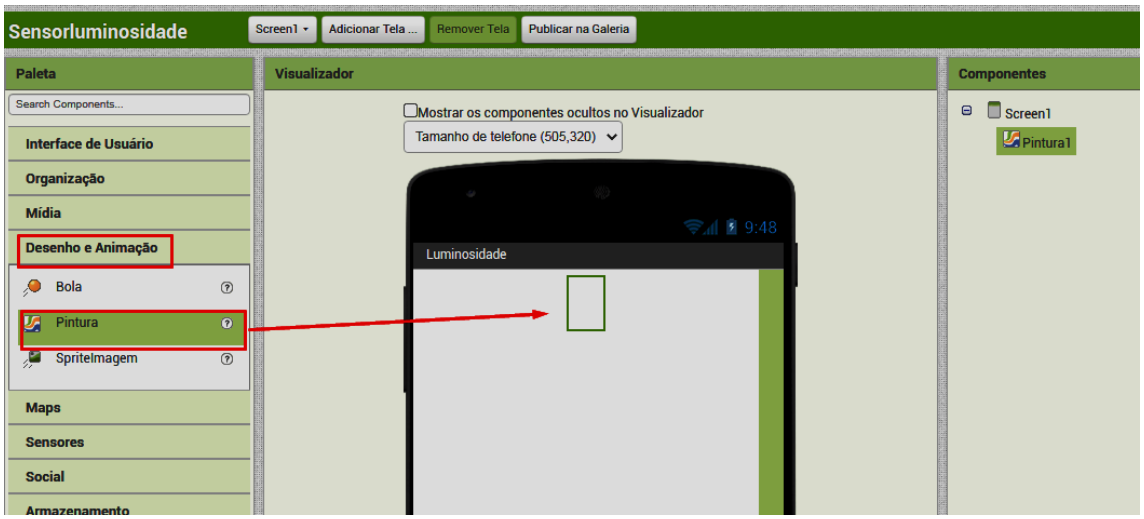


5.1 Criando a interface

Para alterar o nome da tela mostrada no aplicativo, clique no componente **Screen1**, e na lista de **Propriedades** altere o campo **Título** para **Luminosidade**:



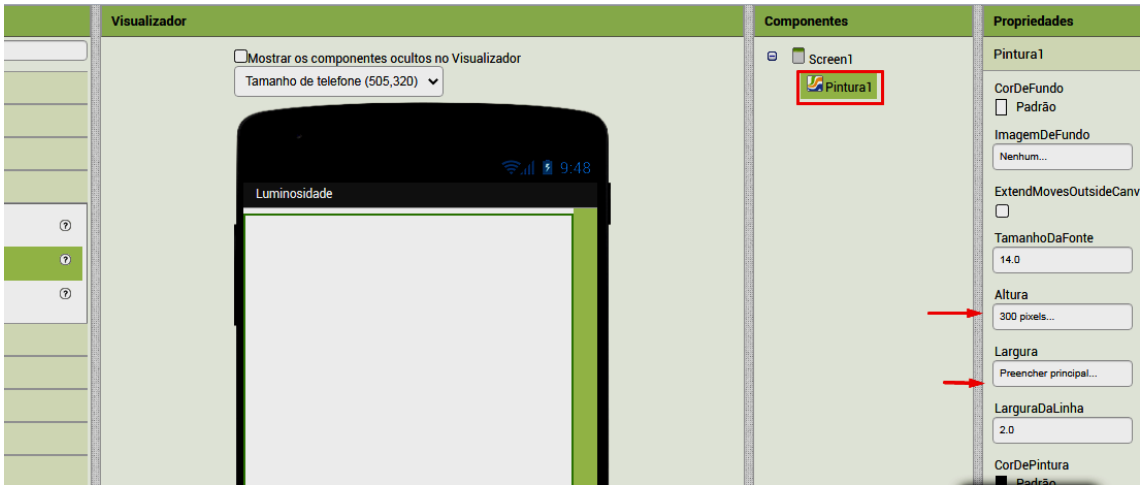
Para inserir um **Spritelmagem**, primeiramente clique na **Paleta Desenho e Animação** e arraste para o **Visualizador** o componente **Pintura** , conforme a figura:



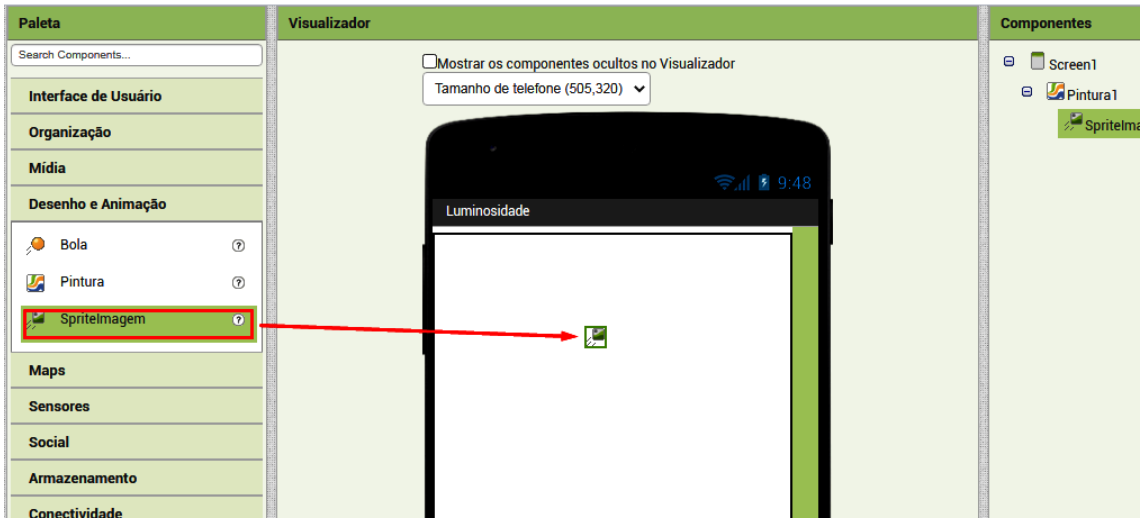
Configure as seguintes **Propriedades** da **Pintura**:

a) **Altura**: 300 pontos

b) **Largura**: Preencher principal

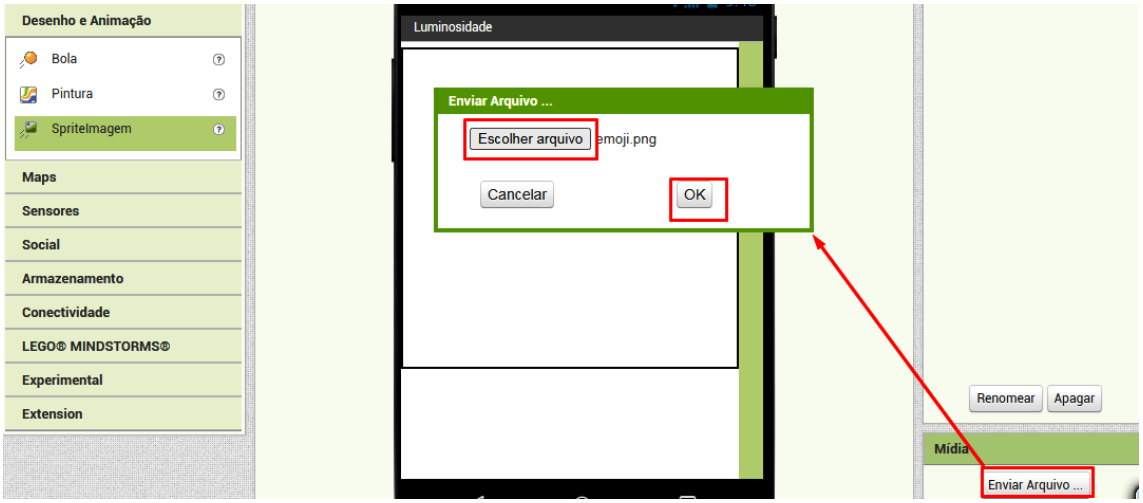


Em **Desenho e Animação** clique e arraste para a **Pintura** um **SprinteImagem**. Nesse componente será colocada a imagem de um *emoji* ou outra de sua preferência, conforme mostra a imagem abaixo:



Baixe em seu computador a imagem de um *emoji* ou outra imagem de sua preferência. Para inserir essa imagem na animação, clique em **Mídia**, na janela que se localiza abaixo da janela de **Componentes** :

a) Clique em **Enviar Arquivo ...**
b) Clique em **Escolher arquivo**
c) Selecione a pasta onde está essa imagem no computador
d) Clique em **OK**

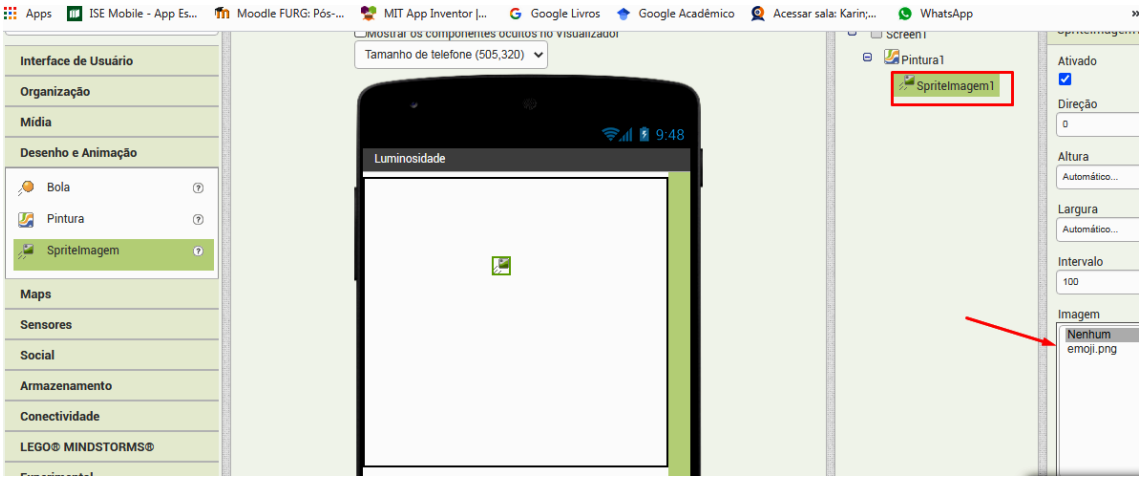


Para definir a imagem do **SpriteImagem** em **Propriedades**:

a) Escolha a opção **Imagem**

b) Selecione o arquivo onde está a **Imagem**

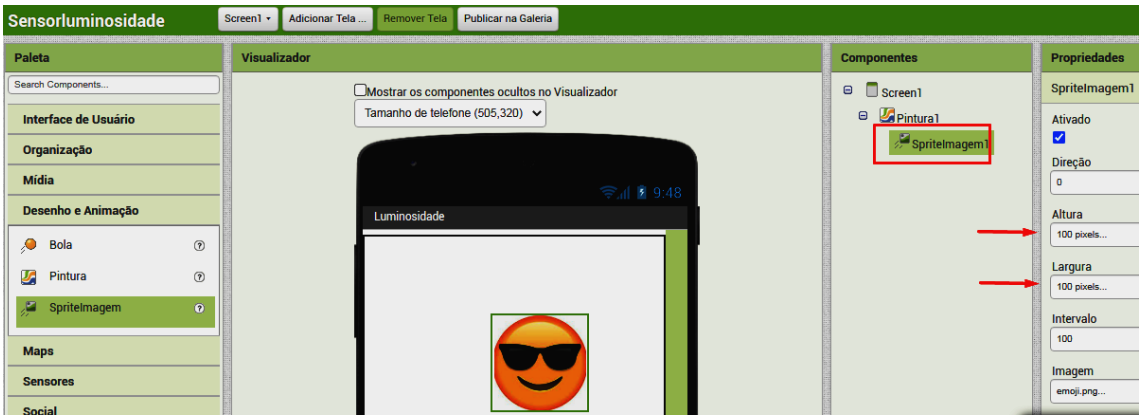
c) Clique em **OK**



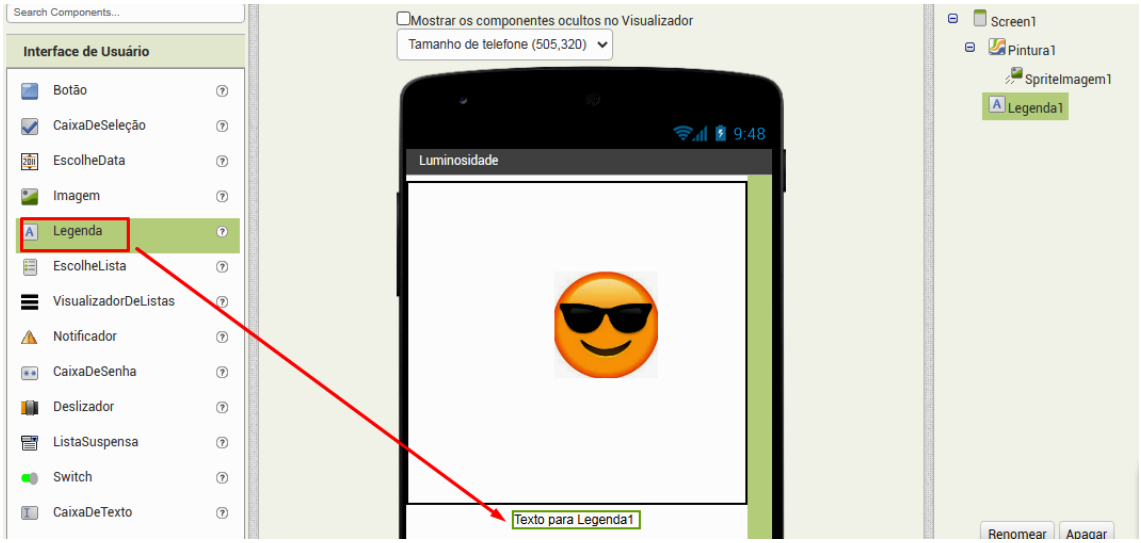
Configure as seguintes **Propriedades** da **Imagem**:

a) **Altura**: 100 pontos.

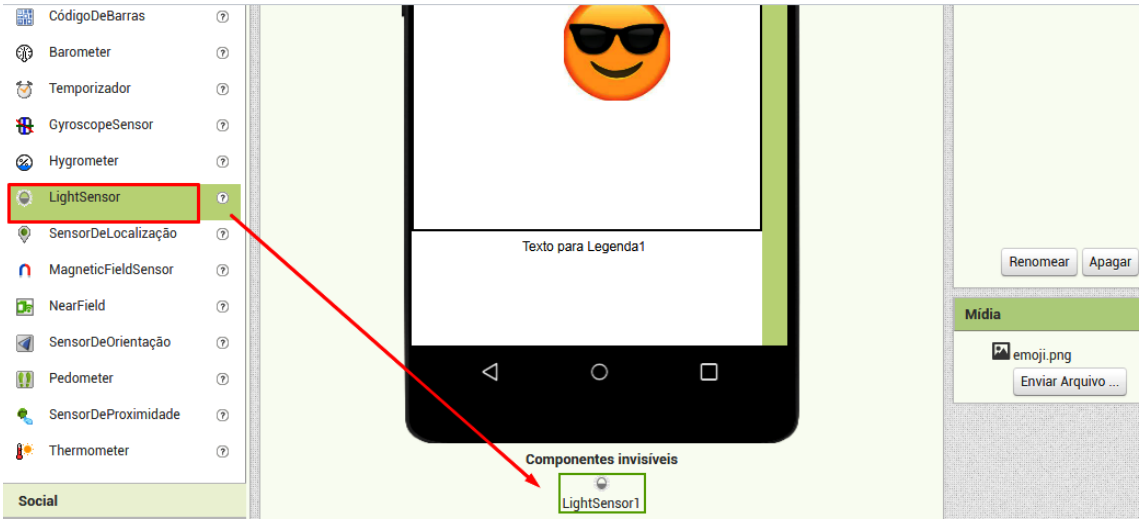
b) **Largura**:100 pontos.



Agora, na **Interface do Usuário**, selecione e arraste para o **Visualizador** uma **Legenda**, conforme a figura abaixo:



Na **Paleta** escolha a opção **Sensores**, clique e arraste o sensor **LightSensor** (sensor da luminosidade) para o **Visualizador**. Observe que ele é um componente invisível que ficará abaixo do **Visualizador**, conforme a figura:



5.2 Programando o aplicativo

A programação dessa animação representará uma função de primeiro grau **y=ax+b**, que relaciona os valores de **x** , obtidos a partir do sensor luminosidade (domínio), aos valores de **y**, as dimensões do *emoji* em pixels (contradomínio). Oriente os alunos a descreverem o tamanho **y** do emoji em relação a luminosidade como uma função de primeiro grau.Também estimule os estudantes a verificar o que acontece com diferentes valores dos coeficientes a e b.

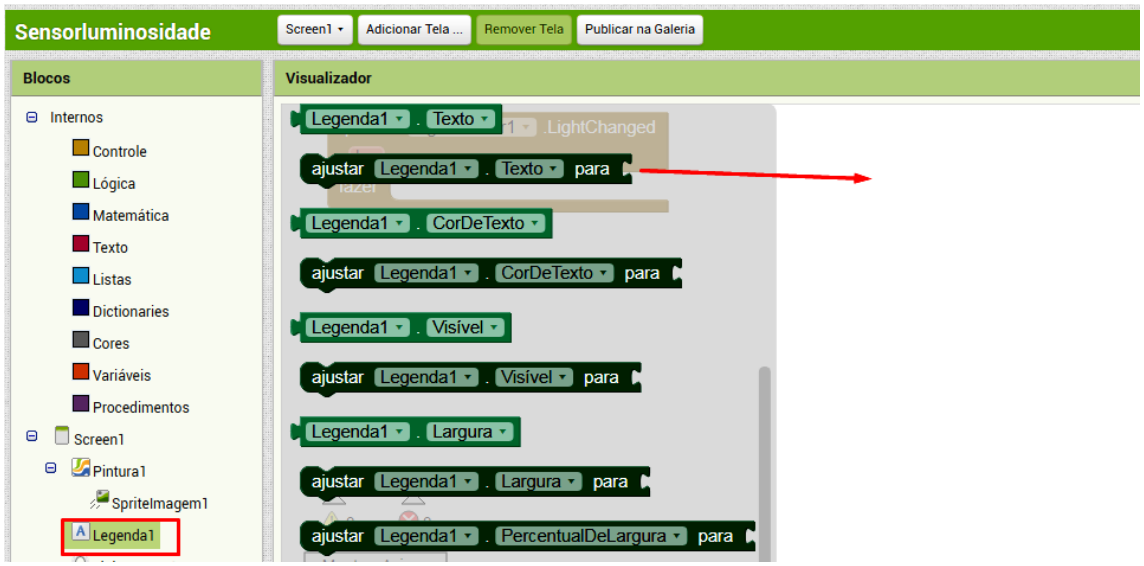
Para o acesso à janela de programação selecione a opção **Blocos** e, em seguida, clique no sensor **LightSensor** e selecione e arraste o bloco **quando LightSensor1.LightChanged fazer** para a tela de programação:

Agora para que a **Legenda** mude de acordo com a luminosidade recebida pelo sensor do *smartphone*:

a) Clique em **Legenda1**

b) Selecione o bloco **ajustar Legenda1.Texto para** e arraste para a tela de programação.

Observe a figura:



Continuando a programação:

a) Clique no bloco **Variáveis**

b) Escolha e arraste para a tela de programação o bloco **obter**, encaixando no bloco da **Legenda1**

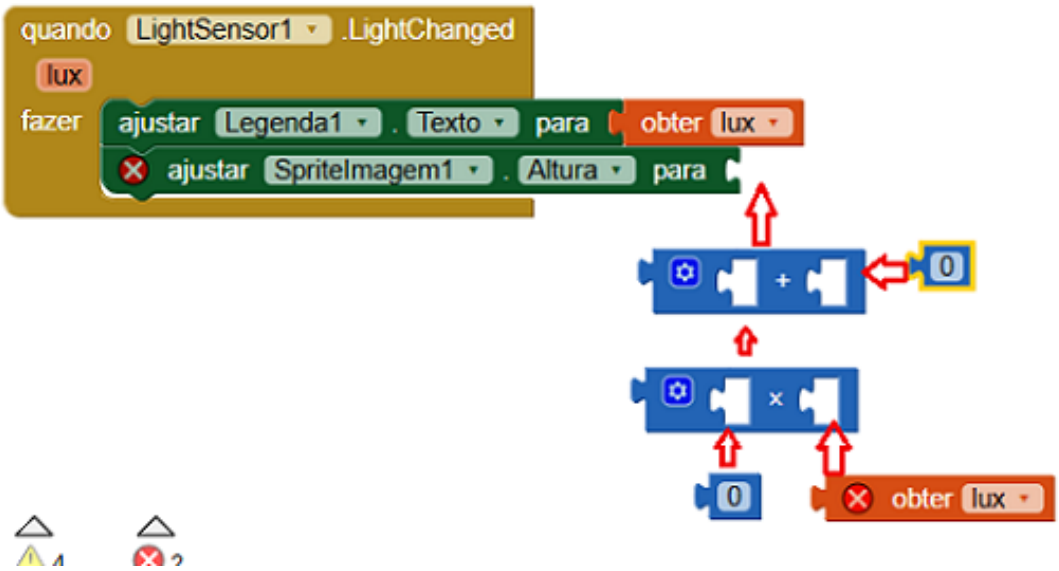
c) Clique no cantinho do bloco **obter** e em **lux**.

Observe a figura abaixo:

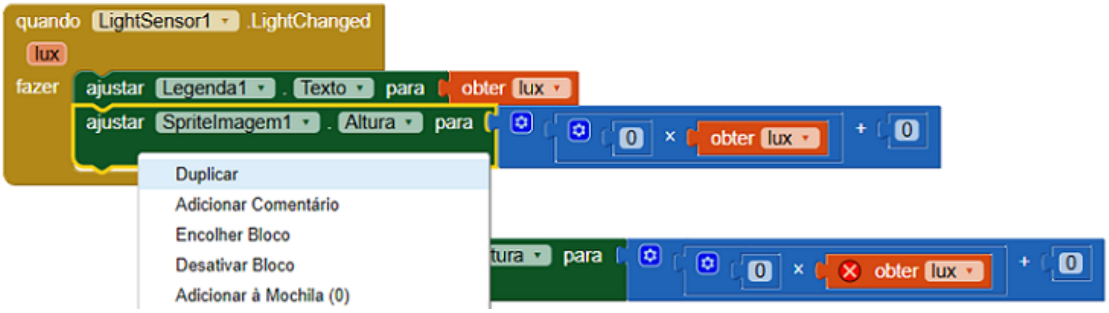
Para que as dimensões do **Spritelmagem1** sejam associadas à quantidade de luz obtida:

a) Clique em **Spritelmagem1**
b) Selecione e arraste para a tela de programação o bloco **ajustarSprintelmagem1.Altura para**
c) Abra o menu de blocos de **Matemática**
d) Escolha o bloco com sinal da **Soma**
e) Faça o mesmo procedimento anterior com os blocos: sinal da **multiplicação** e **número**

Observe a figura abaixo. Nossa variável **lux** é a quantidade de luminosidade, então a multiplicamos por um número, que representará o coeficiente *a* da função de 1º grau. O resultado dessa multiplicação será somado a outro número, que representará o coeficiente *b* da função de 1º grau.

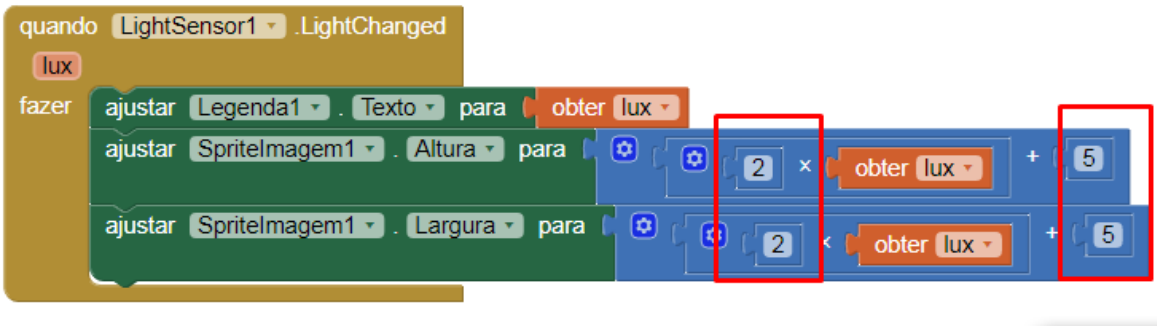


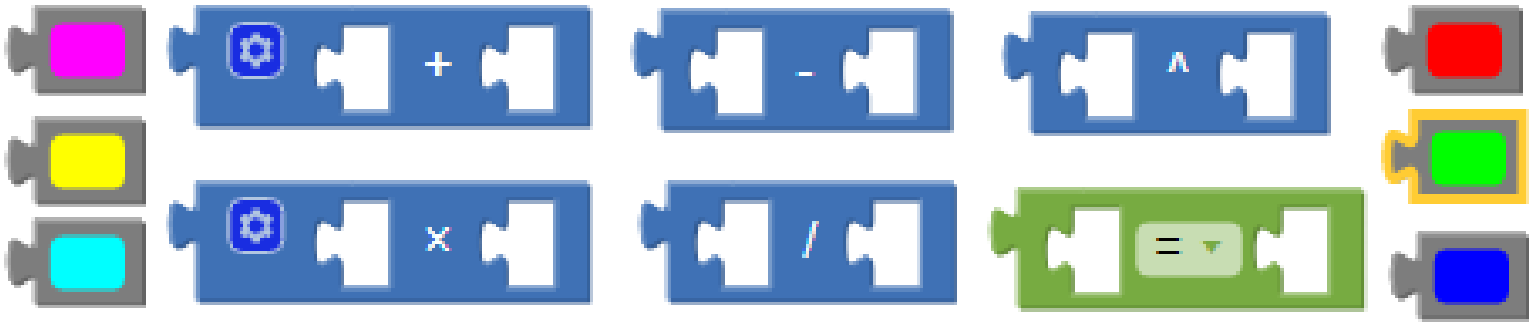
Agora para ajustar a **largura** também de forma automática, vamos duplicar o conjunto de blocos anteriores, clique com o lado direito do *mouse* e selecione a opção **Duplicar**.



Substitua a **Altura** por **Largura**, conforma a figura abaixo:

Para finalizar, vamos atribuir os valores dos coeficientes a e b. É importante que os estudantes experimentem valores diferentes e observem o que acontece com a figura (testando o aplicativo com QR):





6. Estimulando a criatividade

Neste capítulo será proposta uma atividade em que o estudante vai desenvolver uma animação usando o App Inventor e a lei de função do primeiro grau.

Os conteúdos desenvolvidos nessa atividade são : Função de 1º grau, domínio, contradomínio e imagem , função crescente e decrescente e sistema de coordenadas cartesiano. Os objetivos desta atividade são:

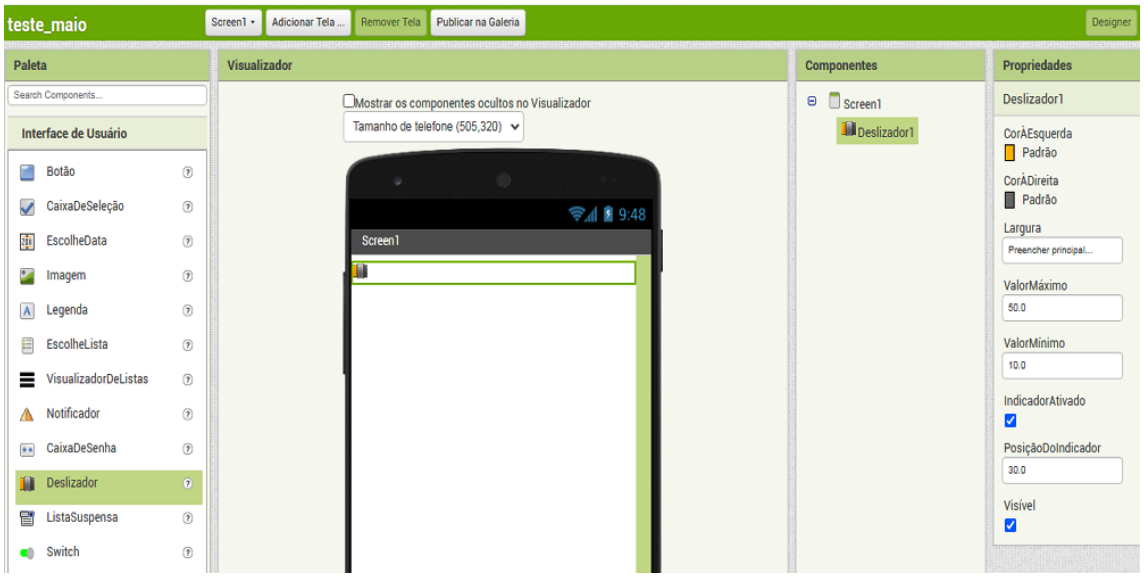
- Criar um aplicativo com animação envolvendo Função de 1º grau $f(x) = ax + b$;
- Incentivar o estudante ao uso de sua criatividade e autonomia;
- Estabelecer relações entre os componentes da animação e o domínio e contradomínio da função de primeiro grau;
- Analisar os coeficientes a e b da função de 1º grau;
- Reconhecer função do primeiro grau crescente e decrescente.

Para criar sua animação, o estudante vai usar uma pintura e escolher um objeto a ser animado, como uma **Bola** ou um **Spritelmagem** (em que ele poderá carregar uma imagem de sua preferência).

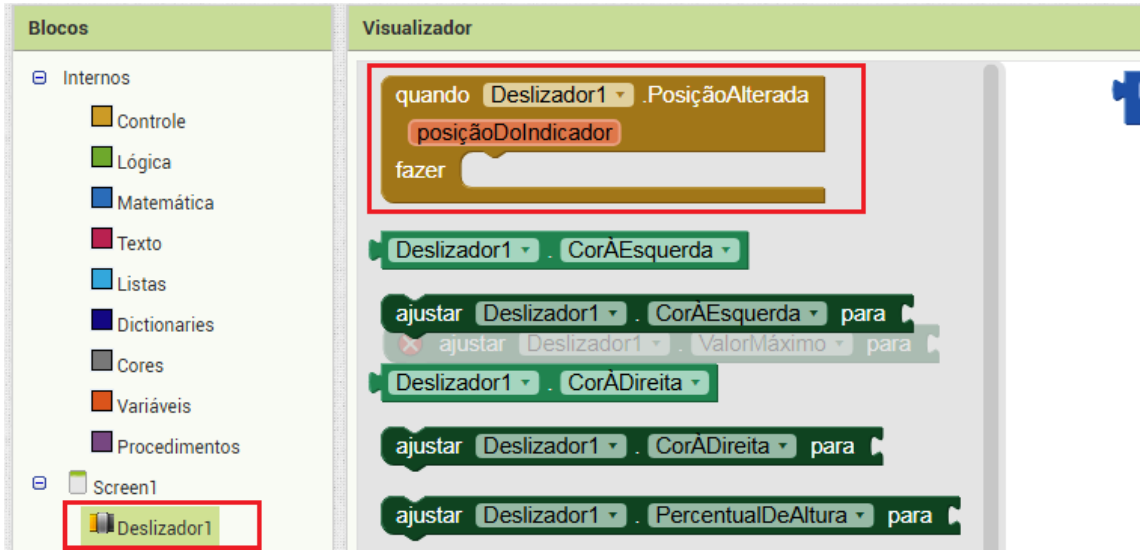
Para fornecer a variável independente da função de 1º grau, o estudante poderá utilizar um temporizador, uma barra deslizante ou algum sensor disponível em seu *smartphone*. Os valores obtidos a partir destes componentes representarão o domínio da função. Abaixo são apresentados alguns componentes disponíveis no App Inventor que podem ser utilizados para controlar animações:

O **Deslizador** é um componente do menu de **Interface de Usuário**, que consistem em uma barra deslizante. Nas propriedades do Deslizador, podemos atribuir um valor mínimo e um valor máximo, ou seja, podemos especificar o intervalo que constituirá o domínio da função. Durante a execução do aplicativo, ao deslizar com o dedo para a esquerda o valor da barra diminui e ao deslizar para a direita o valor aumenta.

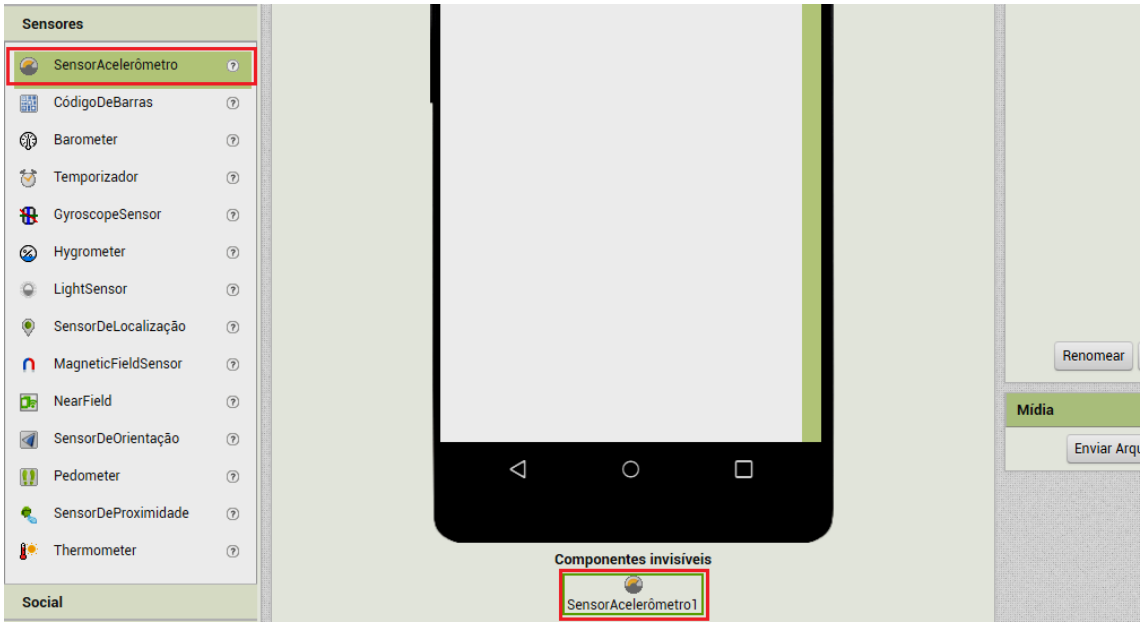
Na imagem observe a localização do deslizador na janela de edição do *Designer*:



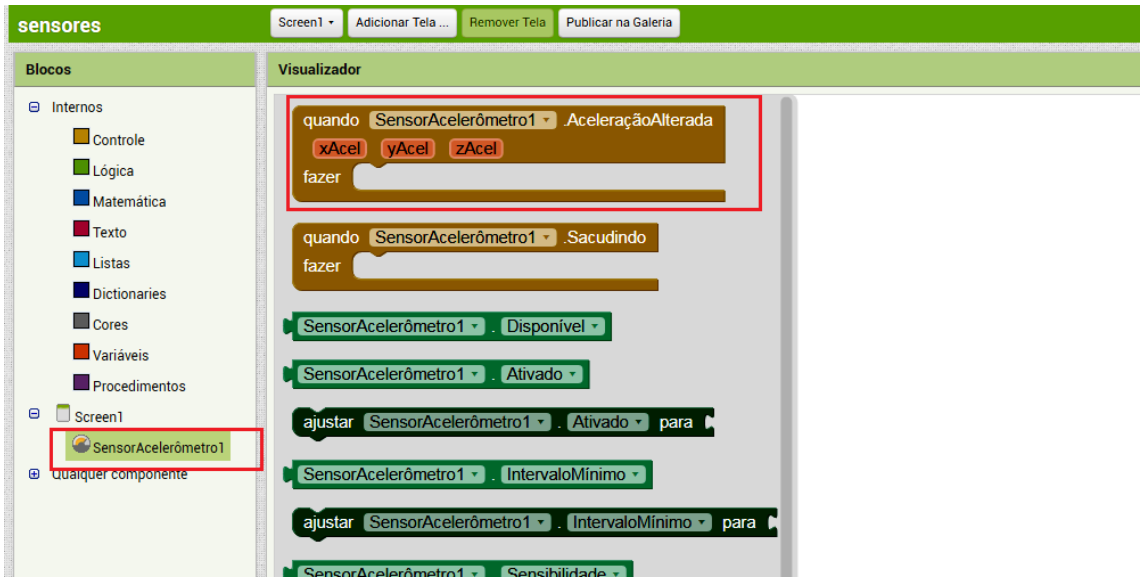
Na janela de edição da programação (**Blocos**), clicando no deslizador vai aparecer o menu de blocos com comandos para esse componente. O bloco **quando Deslizador.PosiçãoAlterada** é utilizado para especificar os comandos que deverão ser executados quando a posição deste deslizador for modificada. A variável **posiçãoDoIndicador**, por sua vez, contém o valor associado à posição do Deslizador.



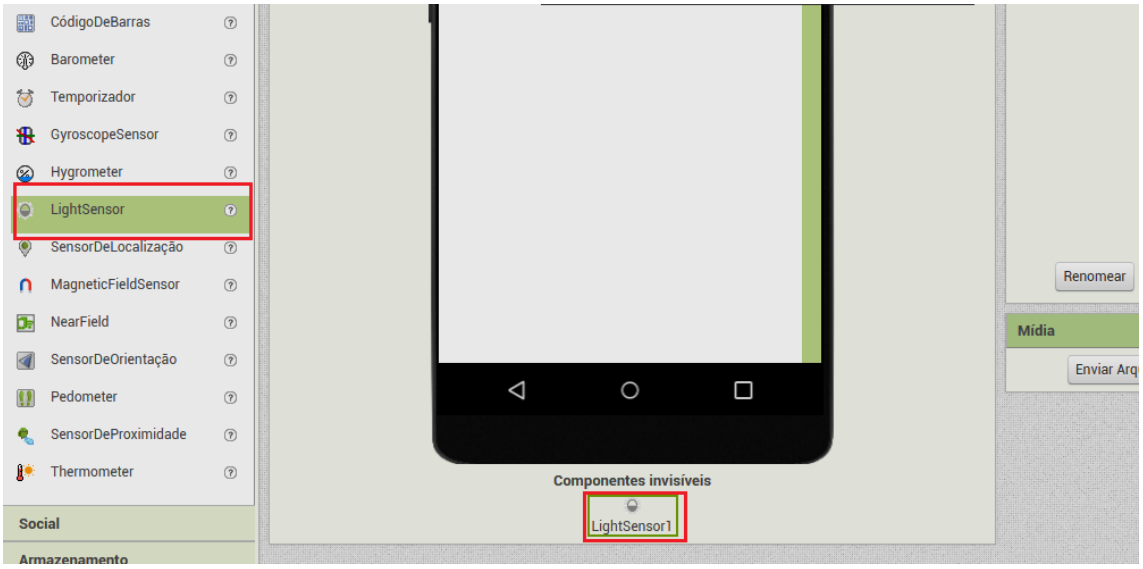
O **Sensor Acelerômetro** é um componente invisível na janela de edição *Designer* que detecta acelerações sofridas pelo *smartphone*. Este sensor pode ser usado para identificar movimentos aos quais o *smartphone* é submetido e, em repouso, responde à inclinação do *smartphone*, visto que este está sempre sobre o efeito da aceleração da gravidade.



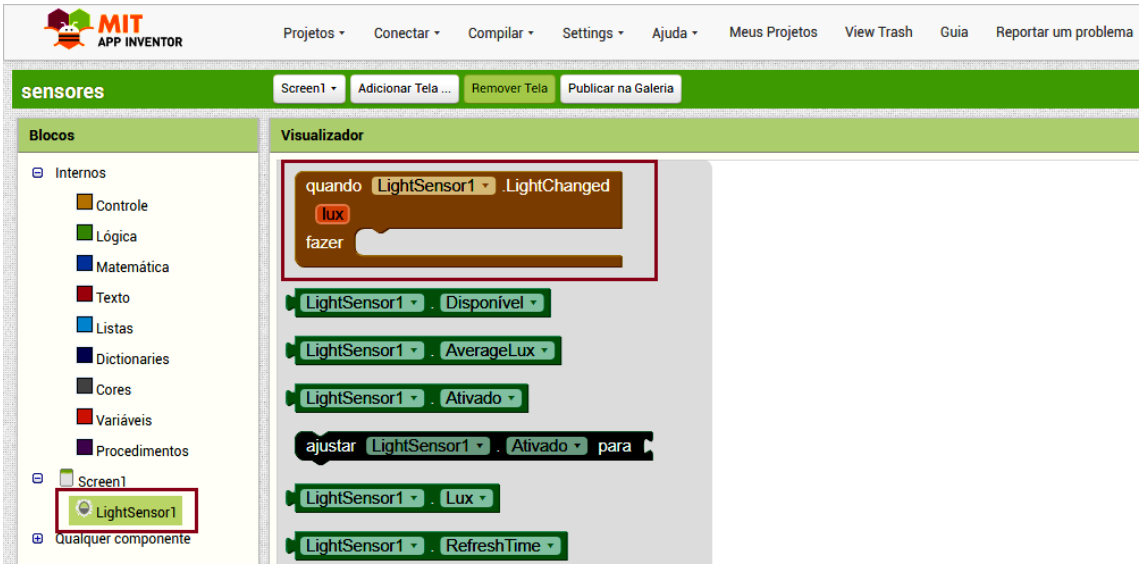
O bloco **quando SensorAcelerômetro.AceleraçãoAlterada** é acionado sempre que há uma mudança na aceleração obtida pelo sensor. As variáveis **xAcel**, **yAcel** e **zAcel** fornecem as acelerações, dadas em metros por segundo ao quadrado (m/s^2), em três direções diferentes. **xAcel** fornece a aceleração da direção 'esquerda-direita', **yAcel** na direção 'cima-baixo' e **zAcel** na direção 'frente-verso' do *smartphone*.



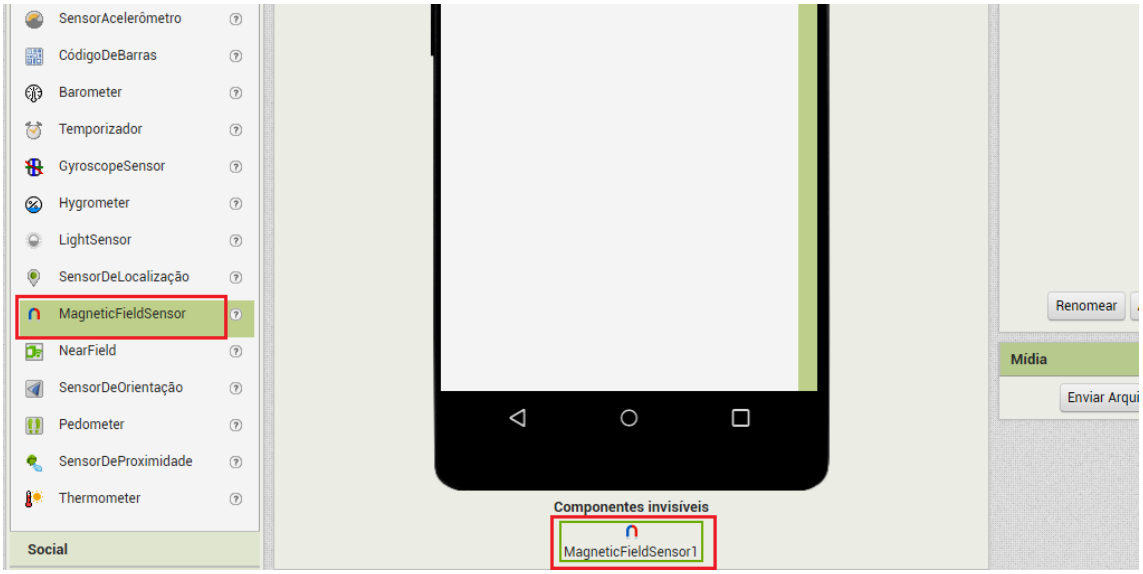
LightSensor é o sensor da luminosidade, um componente que mede a intensidade de luz. Os valores do sensor luminosidade variam de acordo com a quantidade de recebida pelo sensor do *smartphone* que está localizado geralmente na parte frontal do mesmo. Esse componente também é invisível na tela, observe na figura:



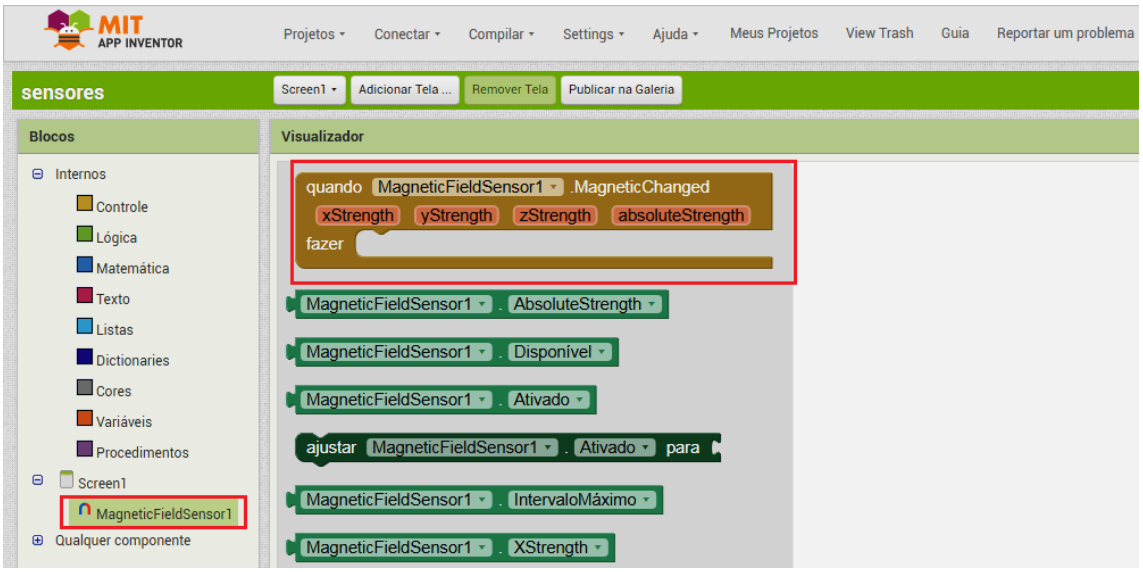
O bloco **quando LightSensor.LightChange** é acionado sempre que há uma mudança na quantidade de luz recebida pelo sensor. A variável **lux** fornece a intensidade da iluminação, dada em *lux* (unidade do Sistema Internacional). O valor dessa variável é sempre um número inteiro, que pode ir de zero (na escuridão total) até a ordem de centenas de milhares, sob luz solar direta.



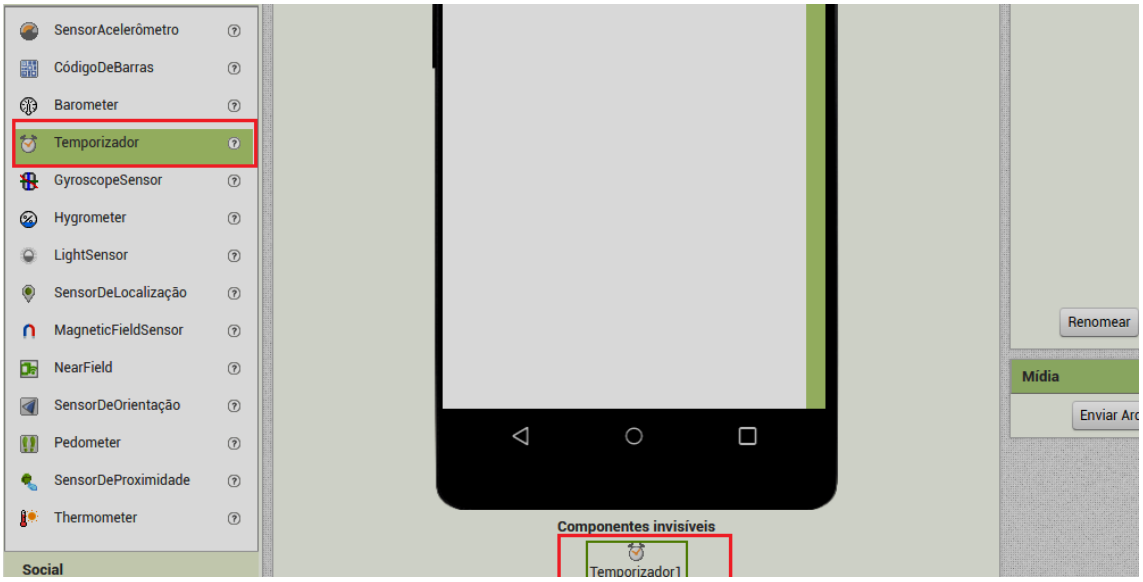
MagneticFieldSensor é o sensor de campo magnético. Ele mede a intensidade do campo magnético no *smartphone* em três direções diferentes. Assim, ele responde à aproximação de objetos magnéticos como imãs e equipamentos elétricos que geram campo magnético. Quando está suficientemente distante de objetos que geram campos magnéticos, ele pode ser usado para identificar a direção do campo magnético terrestre, tal como uma bússola.



O bloco quando **MagneticFieldSensor.MagneticChanged** é acionado sempre que há uma mudança no campo magnético. As variáveis **xStrength**, **yStrength** e **zStrength** fornecem a intensidade do campo magnético nas direções 'esquerda-direita', 'cima-baixo' e 'frente-verso', respectivamente, e estes valores são números reais que podem ser positivos ou negativos, dependendo da orientação do campo magnético. Já a variável **absoluteStrength** fornece a intensidade absoluta do campo magnético, e é sempre um valor positivo.



O **Temporizador** é uma espécie de cronômetro utilizado para a contagem de tempo em intervalos regulares no aplicativo. Ao adicionar este componente no aplicativo, podemos especificar em **Propriedades** o **Intervalo** entre os 'disparos' do temporizador, dado em milissegundos (*ms*). Assim, pode ser usado para descrever animações que evoluem com o tempo.



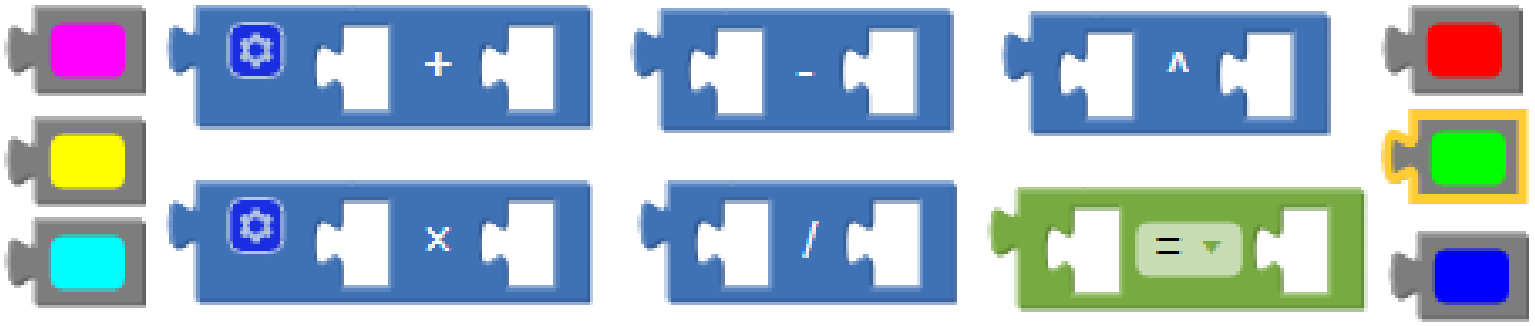
O bloco **quando Temporizador.Disparo** é acionado a cada disparo do temporizador, ou seja, é acionado automaticamente em intervalos de tempo regulares.

Podem ser utilizados ainda outros sensores, tais como barômetro, higrômetro, giroscópio, termômetro, sensor de proximidade e sensor de orientação, dependendo da disponibilidade destes no **smartphone**.

O contradomínio será representado por uma propriedade visual a ser controlada pelo aplicativo, como a posição (horizontal e/ou vertical) do objeto, sua velocidade, direção de movimento, tamanho, etc. Na janela de blocos, clicando no componente de animação escolhido, será apresentado o menu onde estarão os blocos com as propriedades visuais da animação.

Por exemplo, para alterar a velocidade de uma bolinha, na janela de edição de blocos, clique na **Bola** e depois selecione o bloco **ajustar Bola.Velocidade para**:

Uma vez definidos o domínio e contradomínio, o estudante pode estabelecer uma relação entre eles usando a lei da função de primeiro grau. Sugere-se que o estudante seja incentivado a experimentar diferentes valores dos coeficientes a e b e observe as mudanças no comportamento da animação, visando o aprimoramento do entendimento dos conceitos de função crescente e decrescente, e aperfeiçoando sua noção intuitiva sobre funções.



7. Considerações Finais

Neste capítulo serão apresentados alguns apontamentos e recomendações sobre a aplicação desse Guia de Atividades, que utilizou a plataforma App Inventor na criação de aplicativos para *smartphones* como uma facilitadora para o estudo da função de 1º grau .

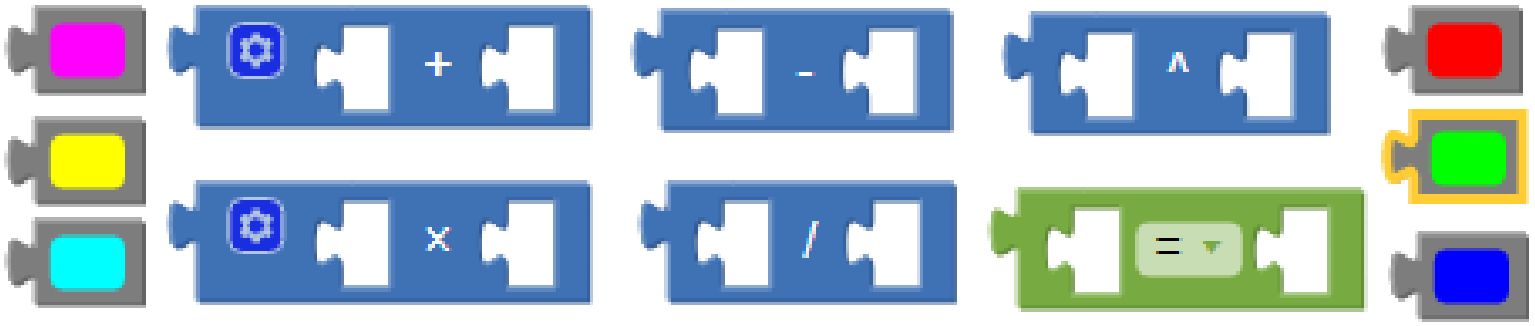
Nesse contexto, destaca-se a importância do papel do professor no decorrer das atividades tornando-se um mediador, desafiando o estudante para que construa seu conhecimento. Muitos estudantes tem o medo de errar, e o professor pode trabalhar o erro como algo natural, que contribui para a aprendizagem e incentiva os estudantes a se aprimorarem, buscando novas estratégias para chegar ao resultado desejado.

É comum durante a programação dos aplicativos que alguns estudantes, ao verificarem que tudo não saiu conforme o planejado, indaguem ao professor à procura de uma solução pronta. Nesse momento, recomenda-se que o professor não forneça as respostas, mas faça outras perguntas que auxiliem o estudante a elaborar suas estratégias para resolver o problema dando continuidade à criação do seu aplicativo. Assim, o professor estará estimulando o desenvolvimento do ciclo Construcionista: **descrição** → **execução** → **reflexão** → **depuração** → **descrição** (novamente) → ...

Na aplicação das atividades, é importante que o estudante seja motivado a realizar diferentes experimentações, observando o comportamento do aplicativo e elaborando suas próprias conclusões, que podem ser compartilhadas com os colegas e com o professor. Nesse sentido, o estudante vai sendo incentivado a desenvolver sua autonomia, criatividade, iniciativa e troca de conhecimentos com os colegas.

A plataforma App Inventor nos possibilita o desenvolvimento de diversos aplicativos, assim as atividades direcionadas ao estudo de Função de 1º grau também podem envolver outros conteúdos de Matemática como Função de 2º grau, Progressões Aritméticas e também conteúdos de Física, por exemplos, Movimento Uniforme (função da posição) e Movimento Uniformemente Variado (função da velocidade).

Espera-se que esse guia de atividades envolvendo a criação de aplicativos que estão tão presentes em nosso cotidiano, auxilie os professores na sua prática pedagógica, contribuindo para que o estudante construa o seu conhecimento participando ativamente do processo.



Referências Bibliográficas

[1] SEF Brasil. *Base Nacional Curricular Comum-BNCC*. 2017 (ver página 4).

[2] S. Papert. *Logo: computadores e educação*. Brasiliense, 1980. URL: `https://books.google.com.br/books?id=nGgOHQAACAAJ` (ver página 6).

[3] S. Papert. *A máquina das crianças: repensando a escola na era da informática*. ARTMED, 1994. ISBN: 9788573070071 (ver páginas 4, 6).

[4] Rafael Felipe Pszybylski. "O uso do software de programação App inventor 2 na formação inicial de professores de ciências". Tese de mestrado. Curitiba: Universidade Tecnológica Federal do Paraná, 2019, página 121 (ver página 7).

[5] Morgana Scheller, Lori Viali e Regis Alexandre Lahm. "A Aprendizagem no contexto das tecnologias: uma reflexão para os dias atuais". Em: *RENOTE-Revista Novas Tecnologias na Educação* 12.2 (2014) (ver páginas 6, 7).

[6] José Armando Valente et al. "O computador na sociedade do conhecimento". Em: *Campinas: Unicamp/NIED* 6 (1999) (ver páginas 6, 7).