

Janice Teresinha Reichert  
Bruna Miecowski  
Milton Kist

# Desenvolvimento de aplicativos matemáticos com o **APP INVENTOR**

Janice Teresinha Reichert  
Bruna Miecowski  
Milton Kist

Desenvolvimento  
de aplicativos  
matemáticos com o  
**APP INVENTOR**

**Editora chefe**

Profª Drª Antonella Carvalho de Oliveira

**Editora executiva**

Natalia Oliveira

**Assistente editorial**

Flávia Roberta Barão

**Bibliotecária**

Janaina Ramos

**Projeto gráfico**

Bruno Oliveira

Camila Alves de Cremo

Luiza Alves Batista

**Imagens da capa**

iStock

**Edição de arte**

Luiza Alves Batista

2023 by Atena Editora

Copyright © Atena Editora

Copyright do texto © 2023 Os autores

Copyright da edição © 2023 Atena

Editora

Direitos para esta edição cedidos à Atena Editora pelos autores.

Open access publication by Atena

Editora



Todo o conteúdo deste livro está licenciado sob uma Licença de Atribuição *Creative Commons*. Atribuição-Não-Comercial-NãoDerivativos 4.0 Internacional (CC BY-NC-ND 4.0).

O conteúdo do texto e seus dados em sua forma, correção e confiabilidade são de responsabilidade exclusiva dos autores, inclusive não representam necessariamente a posição oficial da Atena Editora. Permitido o *download* da obra e o compartilhamento desde que sejam atribuídos créditos aos autores, mas sem a possibilidade de alterá-la de nenhuma forma ou utilizá-la para fins comerciais.

Todos os manuscritos foram previamente submetidos à avaliação cega pelos pares, membros do Conselho Editorial desta Editora, tendo sido aprovados para a publicação com base em critérios de neutralidade e imparcialidade acadêmica.

A Atena Editora é comprometida em garantir a integridade editorial em todas as etapas do processo de publicação, evitando plágio, dados ou resultados fraudulentos e impedindo que interesses financeiros comprometam os padrões éticos da publicação. Situações suspeitas de má conduta científica serão investigadas sob o mais alto padrão de rigor acadêmico e ético.

**Conselho Editorial**

**Ciências Humanas e Sociais Aplicadas**

Prof. Dr. Adilson Tadeu Basquerote Silva – Universidade para o Desenvolvimento do Alto Vale do Itajaí

Prof. Dr. Alexandre de Freitas Carneiro – Universidade Federal de Rondônia

Prof. Dr. Alexandre Jose Schumacher – Instituto Federal de Educação, Ciência e Tecnologia do Paraná

Prof. Dr. Américo Junior Nunes da Silva – Universidade do Estado da Bahia

Profª Drª Ana Maria Aguiar Frias – Universidade de Évora

Profª Drª Andréa Cristina Marques de Araújo – Universidade Fernando Pessoa

Prof. Dr. Antonio Carlos da Silva – Universidade de Coimbra

Prof. Dr. Antonio Carlos Frasson – Universidade Tecnológica Federal do Paraná  
 Prof. Dr. Antonio Gasparetto Júnior – Instituto Federal do Sudeste de Minas Gerais  
 Prof. Dr. Antonio Isidro-Filho – Universidade de Brasília  
 Prof. Dr. Arnaldo Oliveira Souza Júnior – Universidade Federal do Piauí  
 Prof. Dr. Carlos Antonio de Souza Moraes – Universidade Federal Fluminense  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Caroline Mari de Oliveira Galina – Universidade do Estado de Mato Grosso  
 Prof. Dr. Crisóstomo Lima do Nascimento – Universidade Federal Fluminense  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Cristina Gaio – Universidade de Lisboa  
 Prof. Dr. Daniel Richard Sant’Ana – Universidade de Brasília  
 Prof. Dr. Deyvison de Lima Oliveira – Universidade Federal de Rondônia  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Dilma Antunes Silva – Universidade Federal de São Paulo  
 Prof. Dr. Edvaldo Antunes de Farias – Universidade Estácio de Sá  
 Prof. Dr. Elson Ferreira Costa – Universidade do Estado do Pará  
 Prof. Dr. Eloi Martins Senhora – Universidade Federal de Roraima  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Geuciane Felipe Guerim Fernandes – Universidade Estadual de Londrina  
 Prof. Dr. Gustavo Henrique Cepolini Ferreira – Universidade Estadual de Montes Claros  
 Prof. Dr. Humberto Costa – Universidade Federal do Paraná  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Ivone Goulart Lopes – Istituto Internazionele delle Figlie de Maria Ausiliatrice  
 Prof. Dr. Jadilson Marinho da Silva – Secretaria de Educação de Pernambuco  
 Prof. Dr. Jadson Correia de Oliveira – Universidade Católica do Salvador  
 Prof. Dr. Jodeyson Islony de Lima Sobrinho – Universidade Estadual do Oeste do Paraná  
 Prof. Dr. José Luis Montesillo-Cedillo – Universidad Autónoma del Estado de México  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Juliana Abonizio – Universidade Federal de Mato Grosso  
 Prof. Dr. Julio Candido de Meirelles Junior – Universidade Federal Fluminense  
 Prof. Dr. Kárpio Márcio de Siqueira – Universidade do Estado da Bahia  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Kátia Farias Antero – Faculdade Maurício de Nassau  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Keyla Christina Almeida Portela – Instituto Federal do Paraná  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Lina Maria Gonçalves – Universidade Federal do Tocantins  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Lucicleia Barreto Queiroz – Universidade Federal do Acre  
 Prof. Dr. Luis Ricardo Fernandes da Costa – Universidade Estadual de Montes Claros  
 Prof. Dr. Lucio Marques Vieira Souza – Universidade do Estado de Minas Gerais  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Natiéli Piovesan – Instituto Federal do Rio Grande do Norte  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Marianne Sousa Barbosa – Universidade Federal de Campina Grande  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Marcela Mary José da Silva – Universidade Federal do Recôncavo da Bahia  
 Prof. Dr. Marcelo Pereira da Silva – Pontifícia Universidade Católica de Campinas  
 Prof. Dr. Marcelo Pereira da Silva – Pontifícia Universidade Católica de Campina  
 sProf<sup>ª</sup> Dr<sup>ª</sup> Maria Luzia da Silva Santana – Universidade Federal de Mato Grosso do Sul  
 Prof. Dr. Miguel Rodrigues Netto – Universidade do Estado de Mato Grosso  
 Prof. Dr. Pedro Henrique Máximo Pereira – Universidade Estadual de Goiás  
 Prof. Dr. Pablo Ricardo de Lima Falcão – Universidade de Pernambuco  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Paola Andressa Scortegagna – Universidade Estadual de Ponta Grossa  
 aProf<sup>ª</sup> Dr<sup>ª</sup> Rita de Cássia da Silva Oliveira – Universidade Estadual de Ponta Grossa  
 Prof. Dr. Rui Maia Diamantino – Universidade Salvador  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Sandra Regina Gardacho Pietrobon – Universidade Estadual do Centro-Oeste  
 Prof. Dr. Saulo Cerqueira de Aguiar Soares – Universidade Federal do Piauí  
 Prof. Dr. Urandi João Rodrigues Junior – Universidade Federal do Oeste do Pará  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Vanessa Bordin Viera – Universidade Federal de Campina Grande  
 Prof<sup>ª</sup> Dr<sup>ª</sup> Vanessa Ribeiro Simon Cavalcanti – Universidade Federal da Bahia / Universidade de Coimbra  
 Prof. Dr. William Cleber Domingues Silva – Universidade Federal Rural do Rio de Janeiro  
 Prof. Dr. Willian Douglas Guilherme – Universidade Federal do Tocantins

## Desenvolvimento de aplicativos matemáticos no APP Inventor

**Diagramação:** Letícia Alves Vitral  
**Correção:** Flávia Roberta Barão  
**Indexação:** Amanda Kelly da Costa Veiga  
**Revisão:** Os autores  
**Autores:** Janice Teresinha Reichert  
 Bruna Miecoanski  
 Milton Kist

### Dados Internacionais de Catalogação na Publicação (CIP)

R351 Reichert, Janice Teresinha  
 Desenvolvimento de aplicativos matemáticos no APP  
 Inventor / Janice Teresinha Reichert, Bruna  
 Miecoanski, Milton Kist. – Ponta Grossa - PR: Atena,  
 2023.

Formato: PDF

Requisitos de sistema: Adobe Acrobat Reader

Modo de acesso: World Wide Web

Inclui bibliografia

ISBN 978-65-258-0916-8

DOI: <https://doi.org/10.22533/at.ed.168233101>

1. Software aplicativo. 2. Matemática. I. Reichert,  
 Janice Teresinha. II. Miecoanski, Bruna. III. Kist, Milton. IV.  
 Título.

CDD 005.3

Elaborado por Bibliotecária Janaina Ramos – CRB-8/9166

**Atena Editora**

Ponta Grossa – Paraná – Brasil

Telefone: +55 (42) 3323-5493

[www.atenaeditora.com.br](http://www.atenaeditora.com.br)

contato@atenaeditora.com.br

## DECLARAÇÃO DOS AUTORES

Os autores desta obra: 1. Atestam não possuir qualquer interesse comercial que constitua um conflito de interesses em relação ao conteúdo publicado; 2. Declaram que participaram ativamente da construção dos respectivos manuscritos, preferencialmente na: a) Concepção do estudo, e/ou aquisição de dados, e/ou análise e interpretação de dados; b) Elaboração do artigo ou revisão com vistas a tornar o material intelectualmente relevante; c) Aprovação final do manuscrito para submissão.; 3. Certificam que o texto publicado está completamente isento de dados e/ou resultados fraudulentos; 4. Confirmam a citação e a referência correta de todos os dados e de interpretações de dados de outras pesquisas; 5. Reconhecem terem informado todas as fontes de financiamento recebidas para a consecução da pesquisa; 6. Autorizam a edição da obra, que incluem os registros de ficha catalográfica, ISBN, DOI e demais indexadores, projeto visual e criação de capa, diagramação de miolo, assim como lançamento e divulgação da mesma conforme critérios da Atena Editora.

## DECLARAÇÃO DA EDITORA

A Atena Editora declara, para os devidos fins de direito, que: 1. A presente publicação constitui apenas transferência temporária dos direitos autorais, direito sobre a publicação, inclusive não constitui responsabilidade solidária na criação dos manuscritos publicados, nos termos previstos na Lei sobre direitos autorais (Lei 9610/98), no art. 184 do Código Penal e no art. 927 do Código Civil; 2. Autoriza e incentiva os autores a assinarem contratos com repositórios institucionais, com fins exclusivos de divulgação da obra, desde que com o devido reconhecimento de autoria e edição e sem qualquer finalidade comercial; 3. Todos os e-book são *open access*, *desta forma* não os comercializa em seu site, sites parceiros, plataformas de *e-commerce*, ou qualquer outro meio virtual ou físico, portanto, está isenta de repasses de direitos autorais aos autores; 4. Todos os membros do conselho editorial são doutores e vinculados a instituições de ensino superior públicas, conforme recomendação da CAPES para obtenção do Qualis livro; 5. Não cede, comercializa ou autoriza a utilização dos nomes e e-mails dos autores, bem como nenhum outro dado dos mesmos, para qualquer finalidade que não o escopo da divulgação desta obra.

Esta obra traz uma explanação sobre a plataforma App Inventor e a sua utilização na construção de aplicativos para o ensino e aprendizagem de conteúdos da Matemática na Educação Básica.

O livro destaca uma possibilidade de inserção de conceitos da Computação na Educação Básica, através do uso da linguagem de programação através de blocos de encaixe, aliada a conhecimentos específicos. Constitui-se, desta forma, como um material de apoio aos professores de Matemática na construção de aplicativos para *smartphones* e *tablets*, tendo como base os conteúdos da Matemática.

Esperamos contribuir para a inclusão de novas ferramentas no processo de ensino e aprendizagem da Matemática, mais especificamente relacionadas a Computação na Educação Básica e seus eixos: mundo digital, cultura digital e pensamento computacional.

Desejamos uma excelente leitura!

Janice Teresinha Reichert

Bruna Miecowski

Milton Kist

A obra foi elaborada com recursos do projeto de pesquisa: Desenvolvimento e Análise de Aplicativos para Dispositivos Móveis como Ferramenta de Aprendizagem na Educação Básica aprovado no edital 270/UFGS/2020. Assim, nossos agradecimentos à Universidade Federal da Fronteira Sul.

**SUMÁRIO**

<b>APRESENTAÇÃO DO MIT APP INVENTOR</b> .....	<b>1</b>
<b>CONHECENDO O APP INVENTOR 2</b> .....	<b>2</b>
ACESSO AO SOFTWARE .....	2
ABA DESIGNER (OU EDITOR DE ECRÃS) .....	3
ABA BLOCOS .....	4
TESTE DO APLICATIVO (TESTE AO VIVO) .....	6
GALERIA DO APP INVENTOR .....	8
<b>APLICATIVO I: BEM VINDO AO APP INVENTOR 2</b> .....	<b>10</b>
<b>APLICATIVO II: CLASSIFICAÇÃO DE TRIÂNGULOS</b> .....	<b>17</b>
CONCLUINDO O APLICATIVO .....	21
<b>APLICATIVO III: MEDIDAS DE TENDÊNCIA CENTRAL</b> .....	<b>25</b>
<b>APLICATIVO IV: DESCOBRINDO O LADO DESCONHECIDO DE UM TRIÂNGULO RETÂNGULO</b> .....	<b>35</b>
<b>APLICATIVO V: ENCONTRANDO AS RAÍZES DA EQUAÇÃO DE 2° GRAU</b> .....	<b>41</b>
<b>APLICATIVO VI: OPERAÇÕES ENTRE MATRIZES E O DETERMINANTE DE UMA MATRIZ</b> .....	<b>50</b>
<b>APLICATIVO VII: ANÁLISE COMBINATÓRIA</b> .....	<b>63</b>
<b>APLICATIVO VIII: QUIZ SOBRE FUNÇÕES DO 1° E DO 2° GRAU</b> .....	<b>67</b>
<b>CONSIDERAÇÕES FINAIS</b> .....	<b>72</b>
<b>REFERÊNCIAS</b> .....	<b>73</b>
<b>SOBRE OS AUTORES</b> .....	<b>74</b>

# APRESENTAÇÃO DO MIT APP INVENTOR

Em 2010 o Google lançou o software Google App Inventor, como resultado de uma pesquisa de vários anos do Massachusetts Institute of Technology (MIT), coordenada pelo professor de Ciência da Computação e Engenharia, Hal Abelson. Pouco tempo depois o software se intitulou MIT App Inventor, onde é mantido até os dias atuais pela equipe do MIT, composta por alunos, funcionários e liderada pelo professor Hal.

O MIT App Inventor é um ambiente de programação que permite indivíduos com pouco ou nenhum conhecimento sobre programação criarem seus próprios aplicativos Android, para smartphones e tablets, em um período de tempo curto e de maneira prática. Sua linguagem de programação se dá através da montagem de blocos, assemelhando-se a montagem de um quebra-cabeças.

A versão original do MIT App Inventor conhecida como App Inventor Classic (AI1), foi desligada em 2015, entretanto em 2014 já havia uma nova versão do MIT App Inventor, com o objetivo de proporcionar uma melhor experiência aos usuários, chamada de MIT App Inventor 2 (AI2), ou somente **App Inventor 2**.

A versão atualizada do App Inventor possui blocos visuais e recursos de linguagem adicionais no próprio software, sem precisar de um processo separado. O AI2 é composto por duas abas, uma para a montagem visual e inserção dos componentes necessários para o funcionamento do aplicativo e outra aba para a montagem dos blocos, ou seja, para a programação do aplicativo, além disso, a plataforma possui um recurso que permite testar o aplicativo que está sendo desenvolvido.

Nesta obra, o App Inventor é explorado e são apresentados os passos para o desenvolvimento de aplicativos matemáticos com ênfase em objetos do conhecimento específico.

# CONHECENDO O APP INVENTOR 2

Antes de trabalhar com qualquer ferramenta é imprescindível conhecer os seus objetos, a sua organização e o seu funcionamento, assim apresentamos nesta seção as duas abas e os componentes do App Inventor, bem como os passos para o primeiro acesso ao software e a funcionalidade de teste oferecida por ele.

## ACESSO AO SOFTWARE

Para acessar a plataforma do App Inventor 2 e conhecer um pouco da sua funcionalidade siga os itens a seguir na ordem apresentada:

- I. Acesse o site: <http://ai2.appinventor.mit.edu/> .
- II. Faça *login* com uma conta do Google (conta do gmail). Caso não possua, crie uma conta em: <https://accounts.google.com/signup/v2/webcreateaccount?continue=https%3A%2F%2Faccounts.google.com%2FManageAccount%3Fnc%3D1&hl=pt-br&flowName=GlifWebSignIn&flowEntry=SignUp> .
- III. Feito o login, abrirá uma aba, nela clique em CONTINUE.
- IV. Em seguida abrirá uma nova aba, clique em CLOSE.
- V. O idioma estará em inglês, altere-o para “português do Brasil”, conforme a figura 1.

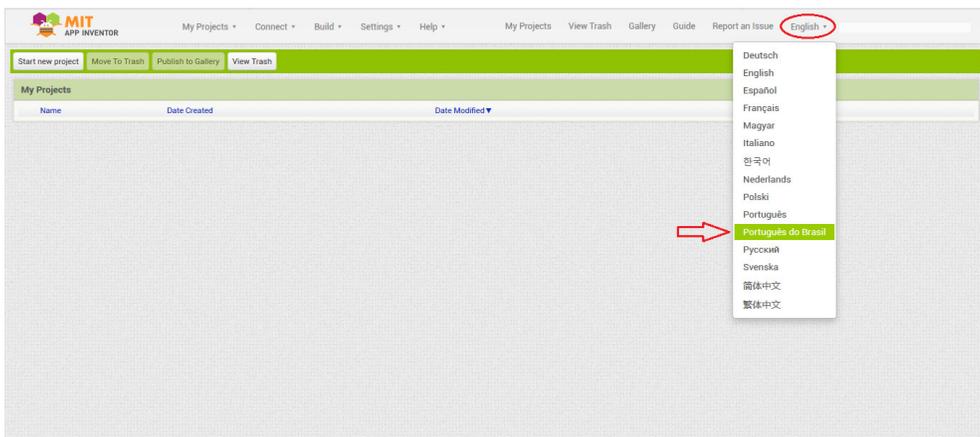


Figura 1 - Alterando o idioma.  
Fonte: Adaptado de App Inventor (2020).

- VI. As duas abas anteriores abrirão novamente. Repita o mesmo processo: clique em CONTINUE e em seguida CLOSE.
- VII. No canto superior esquerdo clique em “Iniciar novo projeto” e dê um nome ao projeto. Observe a figura 2.

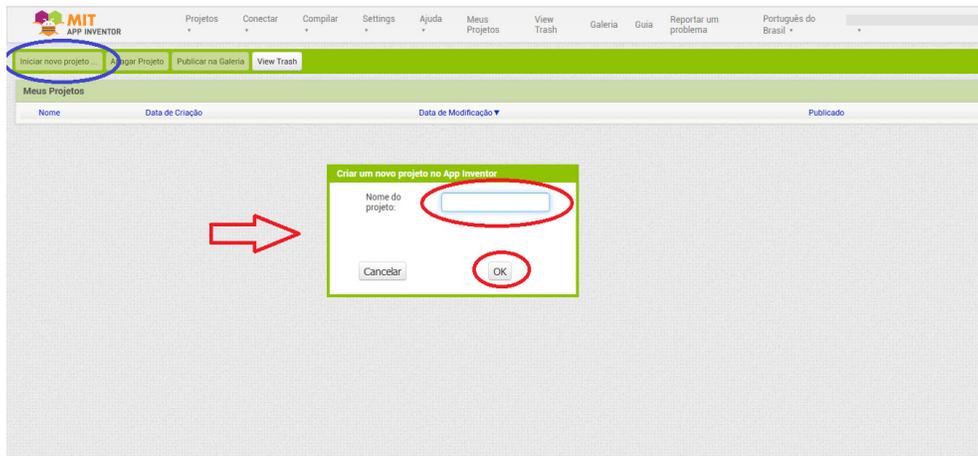


Figura 2 - Novo Projeto.  
Fonte: Adaptado de App Inventor (2020).

Dessa forma você já sabe como acessar e iniciar um projeto no App Inventor, assim o próximo passo é explorar as duas abas.

## ABA DESIGNER (OU EDITOR DE ECRÃS)

A aba de designer ou editor de ecrãs (depende do navegador utilizado), cuja interface está representada na figura 3, é utilizada para criar a aparência do aplicativo, inserindo todos os componentes necessários para o funcionamento desejado do mesmo.

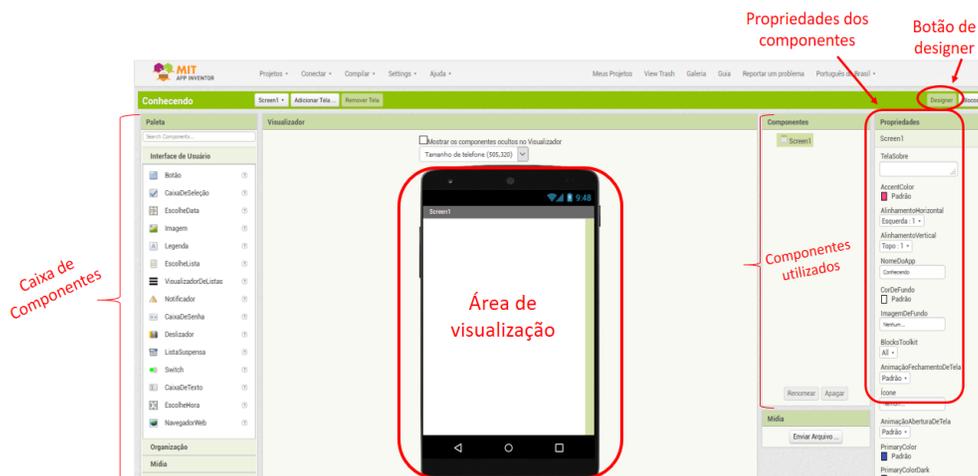


Figura 3 - Interface da Aba de Designer.  
Fonte: Adaptado de App Inventor (2020).

A aba Designer é composta por:

- **Paleta:** É uma caixa de componentes, os quais estão divididos em seções (ga-

vetas). Ao clicar em uma seção aparecerá os seus componentes, por exemplo, na figura 3 está selecionada a seção “interface do usuário”, que tem como alguns de seus componentes o *Botão*, a *CaixaDeSeleção* e o *NavegadorWeb*. Para conhecer mais sobre a paleta abra cada uma das seções e verifique quais os componentes que as compõem, de modo que ao clicar no ponto de interrogação (?) ao lado de cada componente você obterá mais informações sobre o mesmo.

- **Componentes:** No campo “componentes”, assim como representado na figura acima, aparecerão todos os componentes utilizados no desenvolvimento do aplicativo.
- **Propriedades:** Nesse campo você tem acesso as propriedades de cada componente, basta clicar sobre o componente no campo “componentes” (onde aparecem os componentes utilizados). As propriedades são mutáveis, você pode alterar forma, tamanho, cor e imagem, por exemplo, entre várias outras propriedades.
- **Visualizador:** Para montar o aplicativo os componentes das seções são arrastados para o campo visualizador, cujo formato retrata um celular/*tablet*. Nesse campo os componentes podem ser organizados de modo que o aplicativo tenha a aparência desejada.

Em resumo, na aba Designer você tem acesso a todos os componentes para formar o seu aplicativo, de modo que esses devem ser arrastados e organizados na área de visualização e as propriedades dos mesmos podem ser alteradas conforme o desejo do usuário. Assim, conhecendo o essencial desta aba vamos conhecer a Aba Blocos.

## ABA BLOCOS

Na aba Blocos, cuja interface está representada na figura 4, é realizada a programação do comportamento do aplicativo através da montagem de blocos, semelhante a montagem de um quebra cabeça.

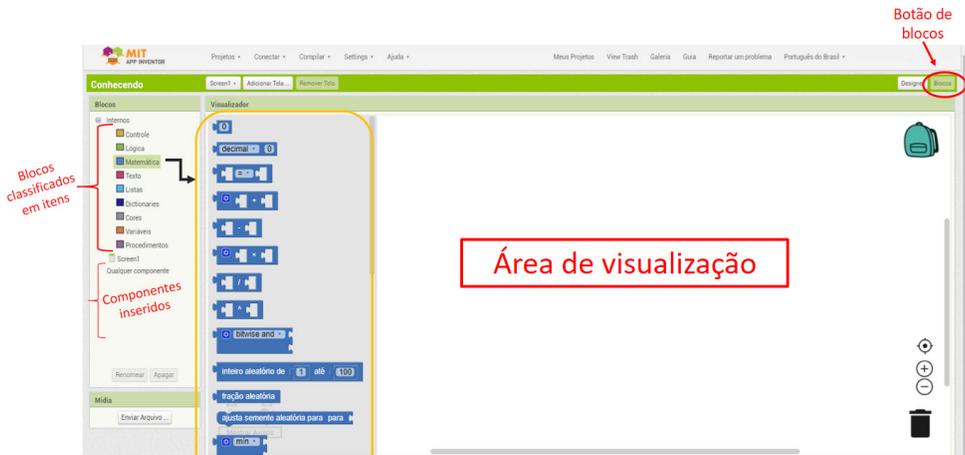


Figura 4 - Interface da Aba Blocos.  
 Fonte: Adaptado de App Inventor (2020).

A aba Blocos é composta por:

- **Botão de Blocos:** Se você está na aba designer e deseja ir para a aba blocos basta clicar no botão “bloco”.
- **Blocs:** Os blocos são classificados em itens de acordo com o comportamento exercido por cada, onde ao clicar em um item, todos os blocos que o compõem são exibidos. Como exemplo, na figura 4 está clicado o item “matemática” e ao lado estão exibidos todos os blocos cujo comportamento se refere a esse item.
- **Blocs específicos:** Abaixo do item “screen1” (ou o nome da sua tela), aparecem todos os componentes inseridos de modo que ao clicar sobre um componente são exibidos os blocos específicos do mesmo, como exemplo na figura 5 o componente “botão” está inserido.

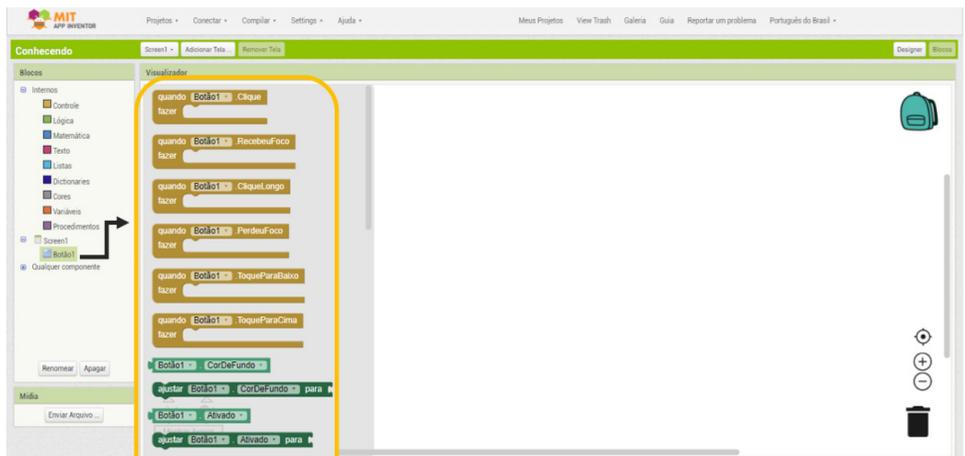


Figura 5 - Os blocos específicos do componente Botão.  
 Fonte: Elaborado pelos autores.

- **Visualizador:** Para programar o aplicativo os blocos são arrastados para o visualizador e ali ocorre a montagem dos mesmos.

## TESTE DO APLICATIVO (TESTE AO VIVO)

Uma das grandes vantagens do App Inventor é a possibilidade de testar o aplicativo enquanto está sendo desenvolvido, com o intuito de verificar se os mesmos estão funcionando como o esperado e possibilitando ajustes a fim de melhorá-los. Há três formas de testes, chamados de testes ao vivo, sendo que a forma mais prática e **recomendada** pelo *site* do MIT App Inventor é testar com o **dispositivo android** (*smartphone* ou *tablet*) e o PC (computador ou notebook) conectados a uma rede de internet, para realizar esse teste siga os passos abaixo:

- I. Acesse a loja de jogos do seu dispositivo *Android* (Google Play Store), procure pelo aplicativo **MIT AI2 Companion** e clique em instalar. Ou clique no *link* a seguir que você será direcionado diretamente para efetuar o download: <https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3>.
- II. Há duas opções a serem seguidas de acordo com a quantidade de dispositivos que você deseja conectar para testar o aplicativo:

II.I. **Para um dispositivo:** Nesse caso, é possível testar o aplicativo em somente um dispositivo simultaneamente, para isso acesse o *site* do App Inventor 2 no PC e inicie ou selecione um projeto. Em seguida no menu superior clique em “Conectar - Assistente AI”, como mostrado na figura 6.



Figura 6 - Testando o aplicativo em um dispositivo.  
Fonte: Adaptado de App Inventor (2020).

II.II. **Para dois ou mais aplicativos:** Aqui é possível que dois ou mais dispositivos se conectem ao aplicativo simultaneamente, a fim de testá-lo/utilizá-lo, de modo que a conexão pode ser realizada por no máximo duas horas. Para realizar esse teste você deve clicar em “Compilar - App (fornecer o QR code para o apk)”, no menu superior da tela, conforme a figura 7.

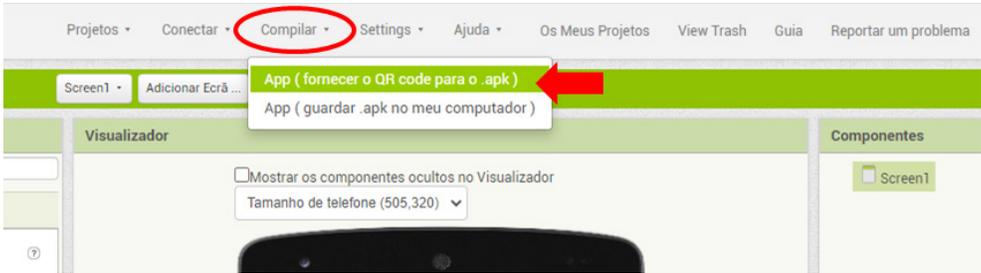


Figura 7 - Testando o aplicativo em vários dispositivos.  
Fonte: Adaptado de App Inventor (2020).

- III. Após realizado o passo II.I ou II.II você deve acessar o aplicativo MIT A1Companion no seu dispositivo e clicar na opção “sScan QR Code” e em seguida escanear o código QR da tela do PC ou digitar o código, observe a figura 8.



Figura 8 - Conectando o dispositivo ao aplicativo.  
Fonte: Elaborado pelos autores.

Agora você pode acessar no seu dispositivo o aplicativo, que será atualizado conforme as modificações feitas na aba de designer e de blocos, entretanto como mencionado anteriormente há três possibilidades para realizar o teste. A segunda possibilidade é através do emulador instalado no PC, onde para obter mais informações sobre o processo acesse o *link* <http://appinventor.mit.edu/explore/ai2/setup-emulator.html>. A terceira possibilidade, cujo uso é recomendado em últimos casos, é através do dispositivo conectado ao PC com um cabo USB, de modo que as informações sobre essa possibilidade se encontram no *link* <http://appinventor.mit.edu/explore/ai2/setup-device-usb.html>.

Agora você já sabe como testar os seus aplicativos, salientamos que foi exposto

somente o passo a passo da primeira opção por se tratar da opção recomendada pela equipe do MIT, que mantém o App Inventor, e por ser uma opção simples e prática, onde no processo não há perda ou alteração de informações.

## GALERIA DO APP INVENTOR

O App Inventor possui uma galeria própria que contém aplicativos dos mais variados temas, desenvolvidos por seus usuários, de modo que o acesso a essa galeria deve ser feito através da sua conta no App Inventor. Estando em sua conta, clique na opção “os meus projetos” e em seguida na opção “login to gallery” na parte superior, assim você será direcionado para a galeria, representada na figura 9.



Figura 9 - Galeria do App Inventor.  
Fonte: Elaborado pelos autores.

Na galeria você tem acesso aos aplicativos disponibilizados, podendo buscar através de assunto, tem acesso a informações da sua conta como aos aplicativos que você publicou na galeria. Em relação às informações da conta, você pode preencher nome, e-mail, utilizar imagem de perfil e escolher a privacidade dos seus aplicativos, de modo que se você selecionar a opção “outras pessoas podem ver seu perfil?”, representada na figura 10, qualquer usuário do App Inventor poderá acessar o seu aplicativo.

Você pode atualizar as informações da sua conta aqui.

Seu endereço de email:

Seu primeiro nome:

Seu último Nome:

Outras pessoas podem ver seu perfil?

Arquivo de imagem:  Nenhum arquivo selecionado

Figura 10 - Privacidade da galeria no App Inventor.  
Fonte: Adaptado de App Inventor (2020).

Caso você não selecionar a opção o aplicativo ficará visível na sua galeria própria, onde pode ser acessado somente por você e pela equipe do MIT, entretanto ao publicar um aplicativo na galeria é gerado um *link* de acesso a ele, o qual você pode disponibilizar para pessoas do seu interesse a fim delas terem acesso a esse aplicativo.

Após a explanação dessa seção, conhecendo um pouco da composição do App Inventor e das suas funcionalidades, você está pronto para começar a desenvolver aplicativos com o mesmo, mas antes de criar os seus próprios aplicativos é importante reproduzir alguns exemplos. Nesse sentido, acompanhe o passo a passo do desenvolvimento dos aplicativos trazidos na sequência e reproduza o desenvolvimento dos mesmos e também altere as propriedades dos componentes e incremente os aplicativos, pois isso lhe auxiliará e facilitará a construção dos seus próprios aplicativos.

## APLICATIVO I: BEM VINDO AO APP INVENTOR 2

Siga os passos citados no capítulo anterior para acessar a plataforma do AI2 e vamos construir o primeiro aplicativo, simples, não matemático, mas que permitirá que você conheça como ocorre a inserção de alguns componentes, linguagens de programação e a respectiva funcionalidade destes. Assim, acompanhe e realize os seguintes passos na ordem descrita.

- **Nomeie o seu projeto/aplicativo**

Chamamos o nosso projeto de *BemVindoAoAPPInventor2*, perceba que o nome não pode conter espaços, pontuação ou acentuação. O aplicativo será um aplicativo “que fale”, quando o botão é tocado ou o celular movimentado.

- **Na aba editor de ecrãs ou designer**

Observe a figura 11 e insira dois botões, arrastando dois componentes “botão” para dentro da caixa de visualização. Em seguida clique sobre o botão 1 e em propriedades altere o texto para “olá”, depois altere o texto do botão 2 para “qual o próximo passo?”.

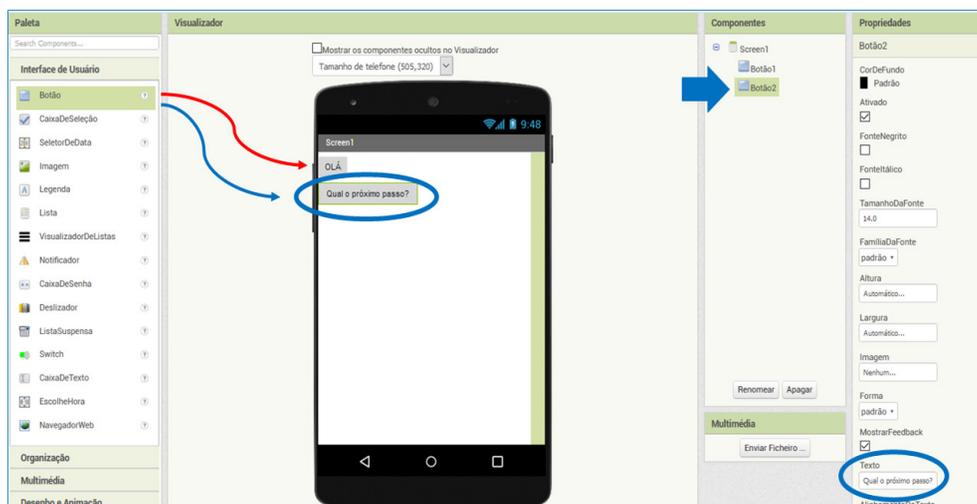


Figura 11 - Inserindo botão no aplicativo I.  
Fonte: Elaborado pelos autores.



Figura 12 - Inserindo acelerômetro.  
Fonte: Elaborado pelos autores.

Queremos que cada botão e o sensor acionem uma fala, portanto na seção multimídia insira três componentes “texto para falar”, lembrando que este componente também é invisível, como apresentado na figura 13.



Figura 13 - Inserindo “texto para falar”.  
Fonte: Elaborado pelos autores.

Na aba de designer já foram inseridos todos os componentes necessários para o aplicativo e agora vamos programá-los na aba blocos.

- **Na aba blocos**

Vamos programar a funcionalidade de todos os componentes, sendo que queremos que quando o “botão 1” for clicado o aplicativo fale. Para tanto, faça como na figura 14.

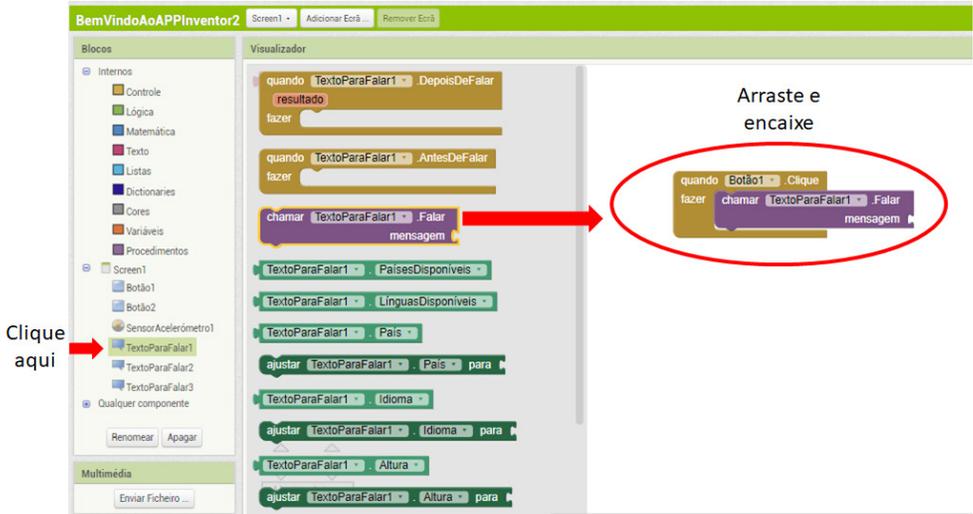


Figura 14 - Blocos do aplicativo I (a).  
Fonte: Elaborado pelos autores.

Queremos que a fala do aplicativo seja: “Olá! Bem vindo ao app inventor 2!”, seguindo a figura 15 insira uma caixa de texto e digite a frase.

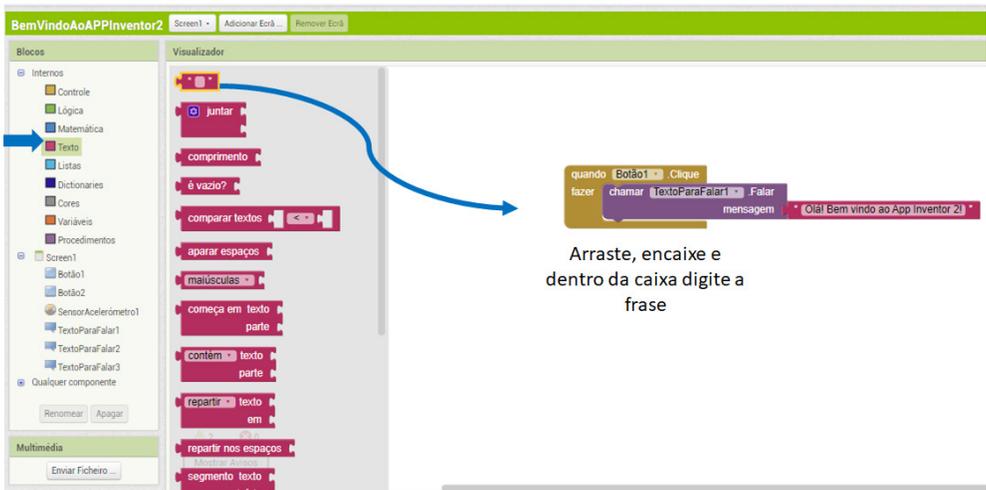


Figura 15 - Blocos do aplicativo I (b).  
Fonte: Elaborado pelos autores.

Queremos também que quando o “botão 2” for clicado o aplicativo fale outra frase, para tanto faça de acordo com a figura 16.

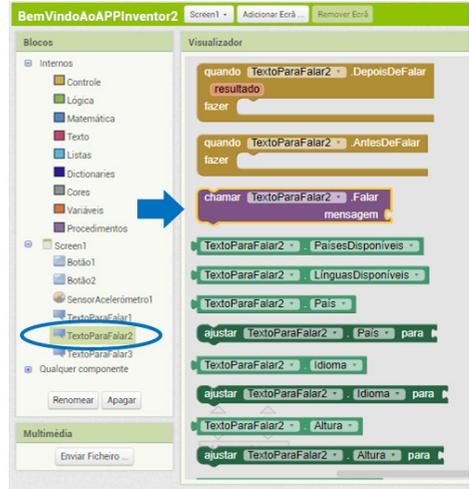
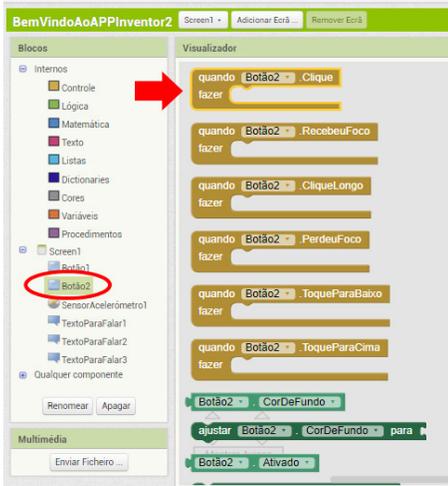


Figura 16 - Blocos do aplicativo (c).  
Fonte: Elaborado pelos autores.

Quando o “botão 2” for clicado o aplicativo irá dizer: “Utilize a sua criatividade e incremente este aplicativo, por exemplo, insira imagem, altere a cor e forma dos textos” e como é uma frase um tanto extensa, para proporcionar uma melhor pronúncia pelo aplicativo é preciso dividi-la em três frases. Assim, seguindo a figura 17 insira três caixas de texto, uma para cada frase e um bloco para juntar as três frases.

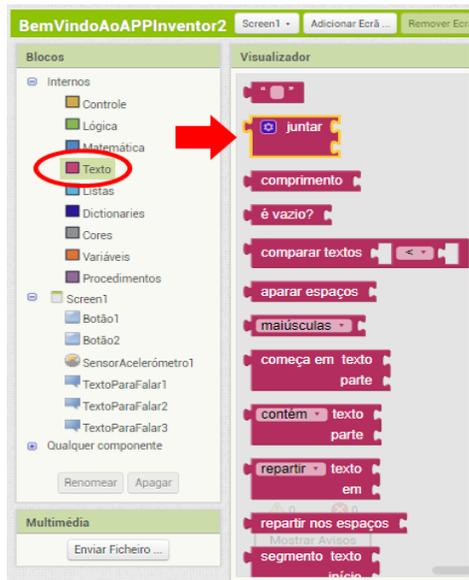
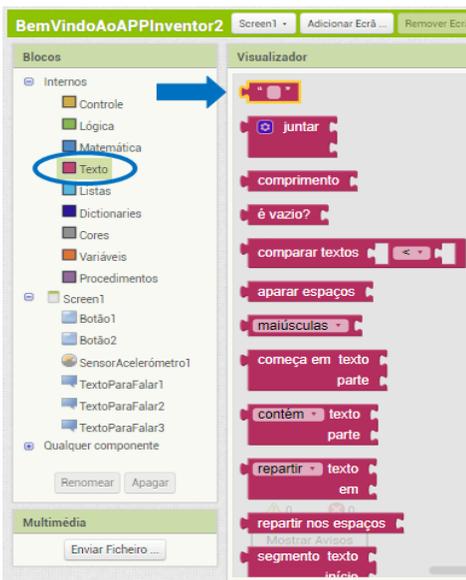


Figura 17 - Blocos do aplicativo I (d).  
Fonte: Elaborado pelos autores.

O bloco de juntar textos está somente com duas janelas, ou seja, com ele podemos

juntar somente dois textos, entretanto podemos abrir uma nova janela para juntar as duas frases com a terceira, como mostra a figura 18 e depois basta encaixar os blocos e digitar as frases.

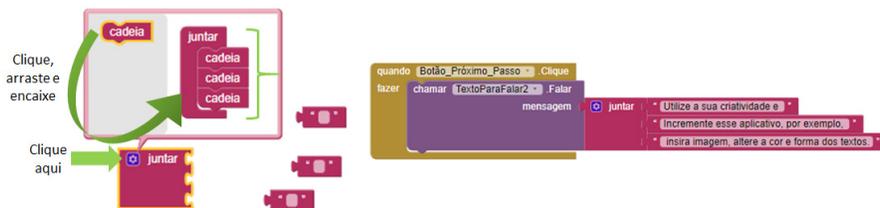


Figura 18 - Blocos do aplicativo I (e).  
Fonte: Elaborado pelos autores.

Queremos que quando o dispositivo for movimentado/sacudido o aplicativo diga “o que você está achando deste software?”, para que isso ocorra, faça conforme a figura 19.

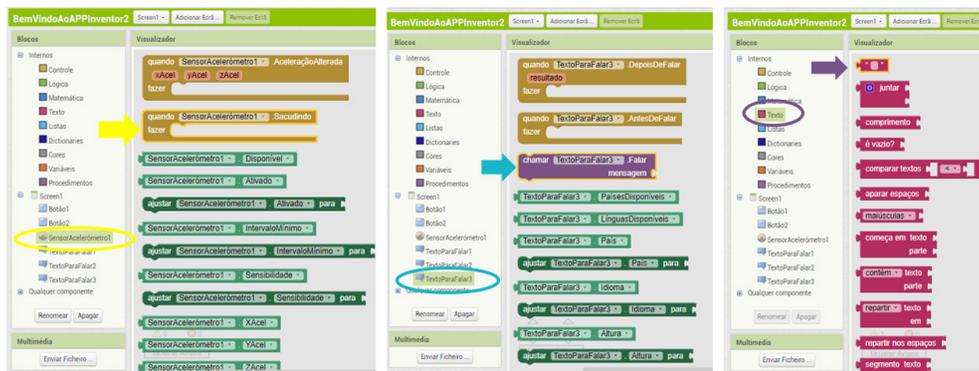


Figura 19 - Blocos do aplicativo I (f).  
Fonte: Elaborado pelos autores.

Em seguida encaixe os blocos e digite a frase, como na figura 20.

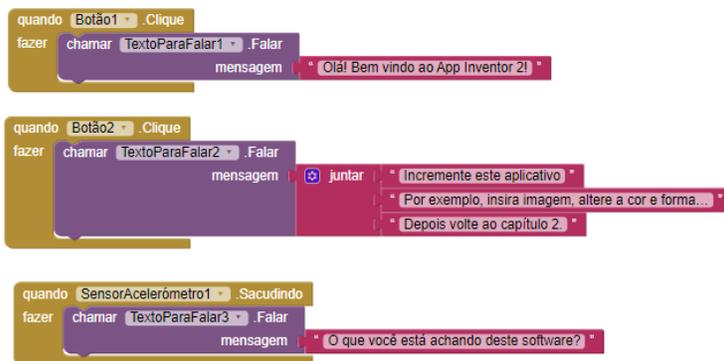


Figura 20 - Blocos do aplicativo I (g).  
Fonte: Elaborado pelos autores.

O aplicativo está programado e agora precisamos testá-lo para ver se o mesmo está funcionando de acordo.

- **Testando o aplicativo**

Com o aplicativo feito, você deve testá-lo para ver se está funcionando como o esperado, assim escolha uma das três formas e seguindo os passos teste o seu aplicativo. Ressaltamos que é importante testar o aplicativo durante todo o seu desenvolvimento e não somente no final, pois se houver algum erro na funcionalidade do mesmo será mais fácil de localizar o equívoco cometido.

- **Incrementando o aplicativo**

Você pode e deve realizar algumas modificações no seu aplicativo, com a finalidade de deixá-lo mais elaborado e de conhecer a sistematização de alguns componentes. Trazemos aqui algumas modificações como mostra a figura 21, chamamos a tela de “seja bem vindo”, ajustamos a organização horizontal dos componentes para “centro”, alteramos algumas propriedades dos botões e os renomeamos, lembrando que nomear os componentes especificamente auxilia na organização, principalmente quando há uma quantidade significativa de componentes utilizados.

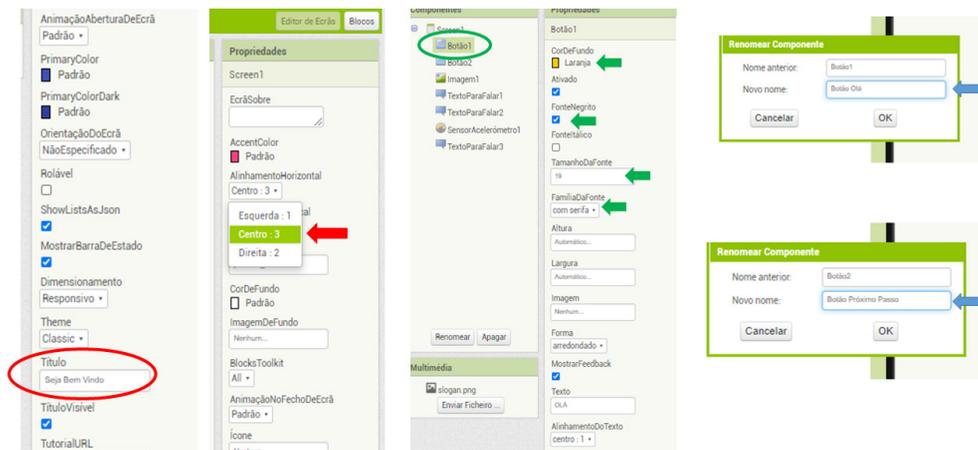


Figura 21 - Incrementando o aplicativo I (a).

Fonte: Elaborado pelos autores.

Além disso, inserimos uma imagem que é mostrada somente quando o botão 1 é clicado. A inserção da imagem é mostrada pela figura 22.

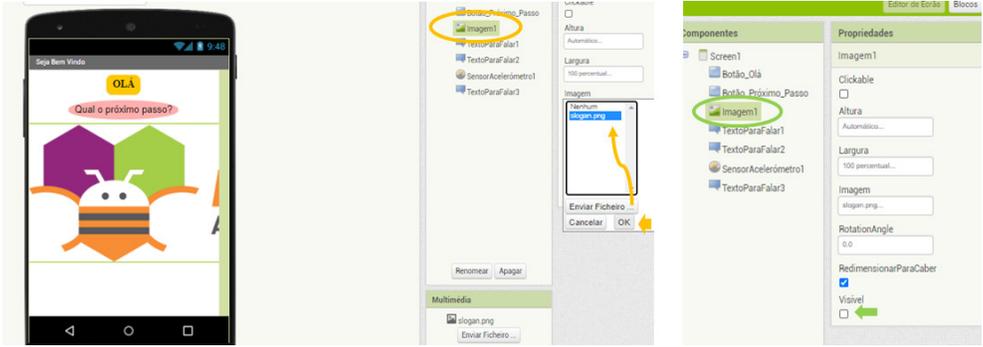


Figura 22 - Incrementando o aplicativo I (b).  
Fonte: Elaborado pelos autores.

Para você compreender a programação referente a imagem, observe a figura 23, que contém também a imagem do aplicativo pronto no dispositivo.



Figura 23 - Incrementando o aplicativo (c).  
Fonte: Elaborado pelos autores.

Os incrementos e modificações foram trazidas como exemplos, mas você pode realizar outras modificações, sendo que o seu primeiro aplicativo está pronto e você pode seguir adiante e desenvolver o próximo.

## APLICATIVO II: CLASSIFICAÇÃO DE TRIÂNGULOS

O aplicativo que será desenvolvido nessa seção é um aplicativo matemático assim como os demais que serão apresentados nesta apostila. Esse está totalmente relacionado ao conteúdo classificação de triângulos e para desenvolvê-lo siga os passos descritos a seguir, lembrando que os triângulos podem ser classificados tanto quanto aos lados como quanto aos ângulos.

### • Na aba Designer

Vamos criar primeiramente um espaço para a classificação quanto aos lados e em seguida o espaço para a classificação quanto aos ângulos. Para o **espaço destinado a classificação quanto aos lados**, representado na figura 24, arraste para a área de visualização três caixas de texto, coloque como dica “lado 1”, “lado 2” e “lado 3”, respectivamente, em propriedades selecione a opção “somente números” e você pode renomear as caixas de texto. Na sequência, acima das caixas de texto insira uma legenda e mude o texto para “Informe a medida dos lados do triângulo”, em seguida abaixo das caixas de texto acrescente um botão e mude o seu texto para “Clique aqui para classificar o triângulo quanto aos lados” e depois adicione uma legenda e exclua o seu texto.

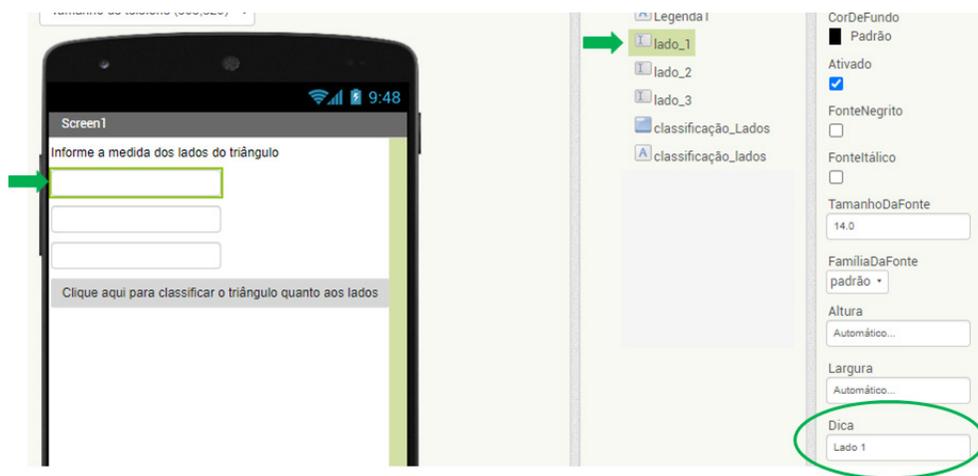


Figura 24 - Espaço para a classificação quanto aos lados.  
Fonte: Elaborado pelos autores.

Agora para o **espaço de classificação quanto aos ângulos** acrescente os mesmos componentes e na mesma ordem conforme os já existentes na área de visualização, de modo que a área de visualização fique conforme a figura 25.

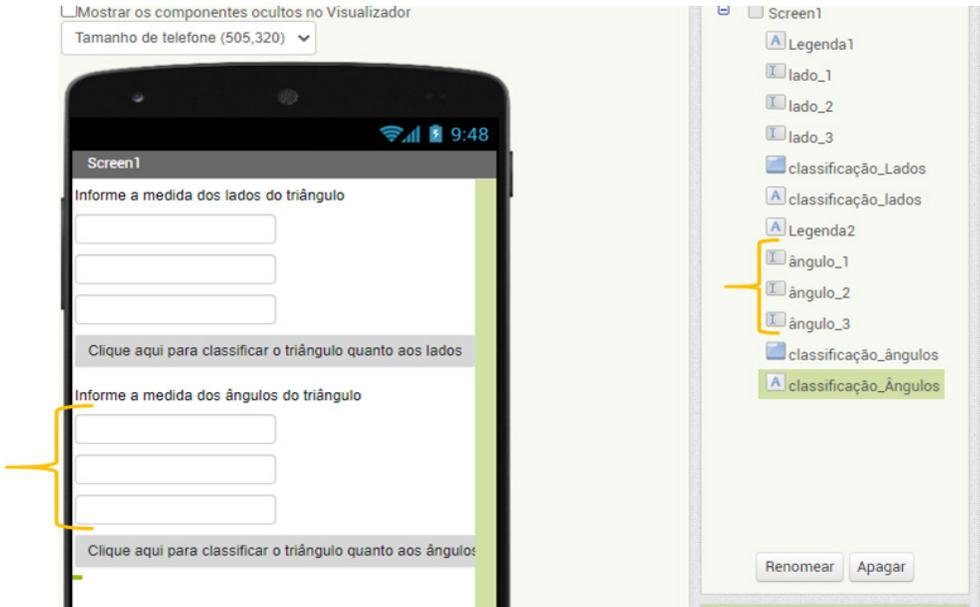


Figura 25 - Componentes na área de visualização do aplicativo II.  
Fonte: Elaborado pelos autores.

Já temos todos os componentes necessários para programar o funcionamento do nosso aplicativo, vamos então para a aba blocos.

#### • Na aba Blocos

Primeiramente programaremos a classificação quanto aos lados, para isso lembre-se dos conceitos: O triângulo é equilátero quando todos os lados possuem a mesma medida, o triângulo é isósceles quando dois lados possuem a mesma medida e o triângulo é escaleno quando os três lados possuem medidas diferentes. A programação dos blocos consistirá em reproduzir essas definições, conforme descrito na sequência.

Primeiramente vamos programar os blocos para a classificação do triângulo equilátero, fazendo como mostra a figura 26, sabendo que a frase contida no bloco texto rosa é “Quanto aos lados o triângulo é classificado como equilátero” e lembre-se que a cor do bloco se refere ao item de blocos que ele está classificado.



Figura 26 - Blocos do aplicativo II (a).  
Fonte: Elaborado pelos autores.

Agora em relação ao triângulo isósceles, perceba que a sua definição não afirma

se o terceiro lado possui medida diferente ou igual a medida dos dois lados com medidas iguais, porém na programação do aplicativo iremos considerar que o terceiro lado possui medida diferente, pois se não afirmarmos que o terceiro lado é diferente, o aplicativo “se confunde” no momento de classificar o triângulo em equilátero ou em isósceles. Então, monte os blocos para a classificação do triângulo isósceles seguindo a figura 27 e observe que a frase escrita no bloco de texto rosa é “ Quanto aos lados o triângulo é classificado como isósceles”.

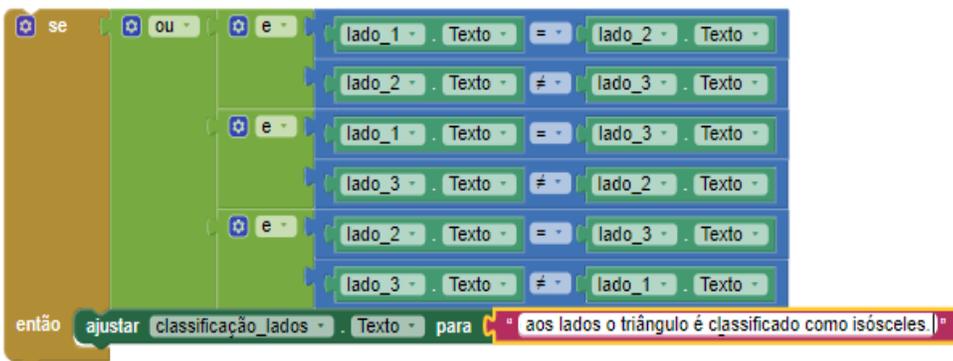


Figura 27 - Blocos do aplicativo II (b).  
Fonte: Elaborado pelos autores.

Agora na classificação do triângulo como escaleno basta somente reproduzir a sua definição com os blocos, assim como mostra a figura 28.

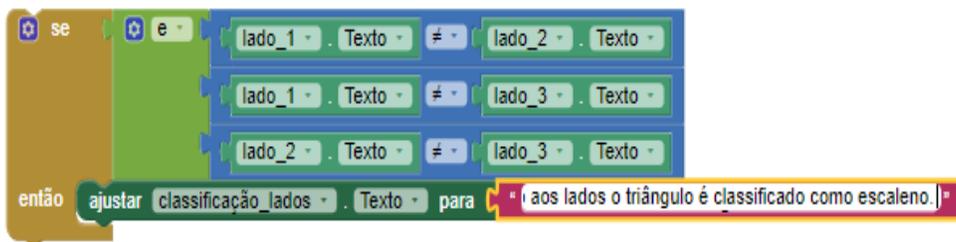


Figura 28 - Blocos do aplicativo II (c).  
Fonte: Elaborado pelos autores.

Para concluir a programação da classificação quanto aos lados encaixe todos os blocos conforme a figura 29.

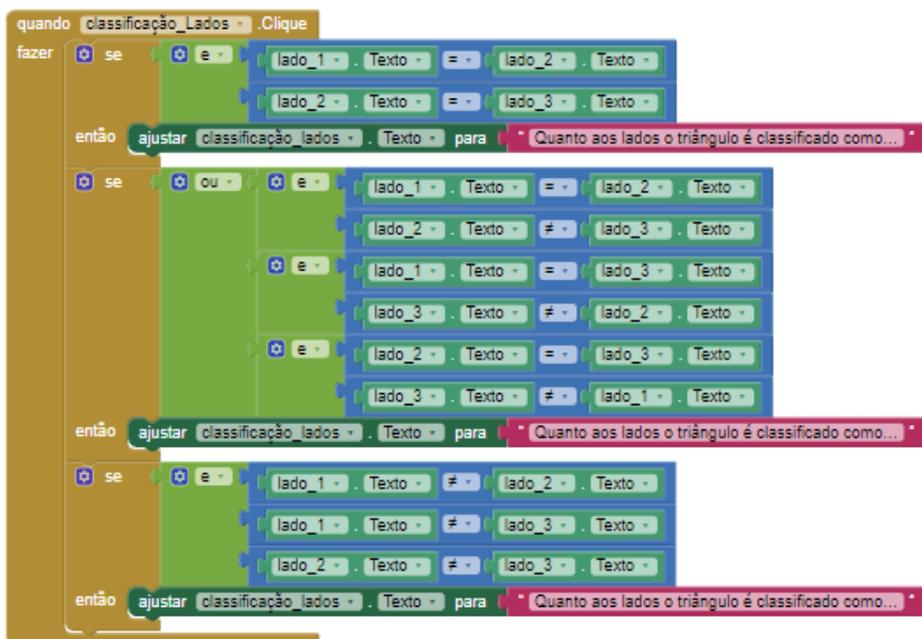


Figura 29 - Blocos do aplicativo II (d).  
 Fonte: Elaborado pelos autores.

Agora vamos programar os componentes para a classificação quanto aos ângulos, lembrando que o triângulo retângulo tem um ângulo de  $90^\circ$ , o triângulo obtusângulo tem um ângulo maior de  $90^\circ$  e o triângulo acutângulo possui todos os ângulos menores que  $90^\circ$ . Assim, seguindo os conceitos da classificação quanto aos ângulos, vamos programar os blocos, de forma semelhante ao que foi feito na classificação quanto aos lados.

Monte os blocos de acordo com a figura 30, sabendo que a frase no primeiro bloco de texto rosa é “ Quanto aos ângulos o triângulo é classificado como retângulo”, no segundo é “Quanto aos ângulos o triângulo é classificado como obtusângulo” e no terceiro “Quanto aos ângulos o triângulo é classificado como acutângulo”.



Figura 30 - Blocos do aplicativo II (e).  
 Fonte: Elaborado pelos autores.

## CONCLUINDO O APLICATIVO

Ao utilizar o aplicativo pode ocorrer do usuário não possuir a medida de todos os ângulos ou de todos os lados. Se as medidas informadas ao aplicativo estão compreendidas na definição da classificação, como por exemplo, se a única medida de ângulo for  $102^\circ$ , logo o aplicativo irá informar que se trata de um triângulo obtusângulo, pois o triângulo possui um ângulo maior que  $90^\circ$  (definição de triângulo obtusângulo). Porém, se as medidas informadas não estão compreendidas na definição, como por exemplo, informando somente 17 e 25 como medidas dos lados, o aplicativo não apresenta nenhuma classificação quanto aos lados, pois dependendo da outra medida o triângulo pode ser isósceles ou escaleno. Assim, deixe claro ao usuário, de acordo com a figura 31, que se ele não possuir todas as medidas pode deixar o campo destinado às demais em branco, porém caso o aplicativo não informe a classificação ele deve buscar informar as demais medidas.

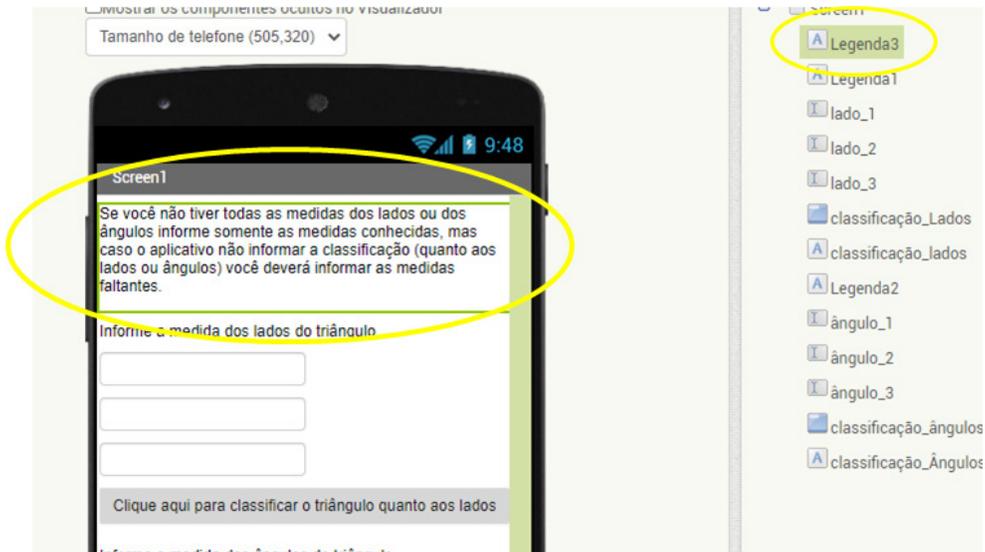


Figura 31 - Concluindo o aplicativo II.  
Fonte: Elaborado pelos autores.

Lembre-se que a soma dos ângulos internos de um triângulo é  $180^\circ$  e que de acordo com a desigualdade triangular a soma da medida de dois lados é sempre maior que a medida do terceiro lado. Porém, pode ocorrer do usuário cometer algum equívoco e informar ângulos cuja soma dos três ângulos não seja  $180^\circ$  ou medida de um lado que seja maior do que a soma da medida dos outros dois lados, onde neste caso não se trata de medidas referentes a triângulos. Entretanto o aplicativo do jeito que está não sabe destas condições e acabará informando a classificação dos supostos triângulos de acordo com os comandos/definições que ele possui, por isso programe o aplicativo para ele informar o usuário se ele está cometendo algum dos equívocos, para isso siga a figura 32.



Figura 32 - Blocos do aplicativo II (f).  
 Fonte: Elaborado pelos autores.

Agora você deve testar o aplicativo para saber se ele está funcionando conforme o esperado, além disso você pode modificar o designer do aplicativo tornando-o mais interessante, para isso altere algumas propriedades dos componentes já existentes ou insira outros componentes que achar conveniente. A figura 33 representa o designer do aplicativo pronto, cujas imagens, cada uma representando uma classificação, são exibidas de acordo com a classificação do triângulo.

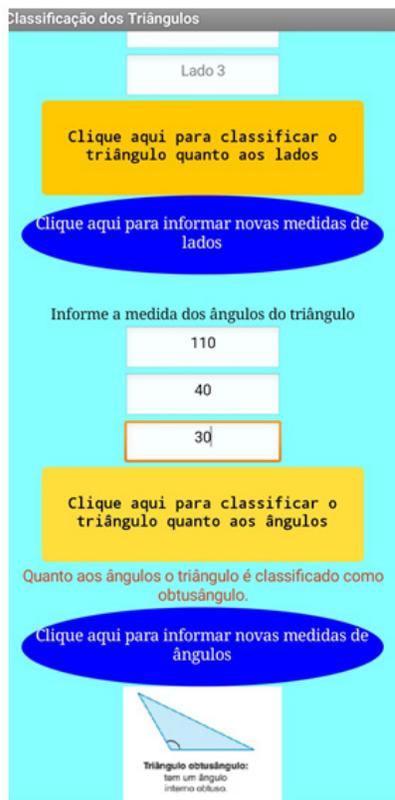
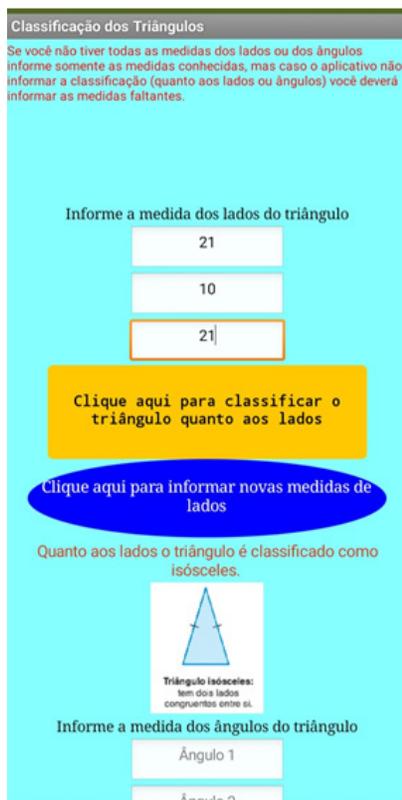


Figura 33 - Interface do aplicativo II concluído.

Fonte: Elaborado pelos autores.

O aplicativo sobre classificação de triângulos está concluído, podemos assim desenvolver o próximo. Ao utilizar o aplicativo pode ocorrer do usuário não possuir a medida de todos os ângulos ou de todos os lados. Se as medidas informadas ao aplicativo estão compreendidas na definição da classificação, como por exemplo, se a única medida de ângulo for  $102^\circ$ , logo o aplicativo irá informar que se trata de um triângulo obtusângulo, pois o triângulo possui um ângulo maior que  $90^\circ$  (definição de triângulo obtusângulo). Porém, se as medidas informadas não estão compreendidas na definição, como por exemplo, informando somente 17 e 25 como medidas dos lados, o aplicativo não apresenta nenhuma classificação quanto aos lados, pois dependendo da outra medida o triângulo pode ser isósceles ou escaleno. Assim, deixe claro ao usuário, de acordo com a figura 31, que se ele não possuir todas as medidas pode deixar o campo destinado às demais em branco, porém caso o aplicativo não informe a classificação ele deve buscar informar as demais medidas.

## APLICATIVO III: MEDIDAS DE TENDÊNCIA CENTRAL

Vamos agora construir um aplicativo sobre as tendências centrais, com o objetivo de calcular a média aritmética, a média aritmética ponderada, a mediana e a moda. No caso do nosso aplicativo ele será elaborado para cálculos com no máximo cinco elementos, caso você queira pode fazer para um número maior ou menor de elementos.

Nesse aplicativo precisaremos trabalhar com mais de uma tela para facilitar o desempenho e o desenvolvimento do aplicativo, assim após acessar a sua conta no App Inventor siga os passos descritos a seguir.

### • Tela 1 - Screen 1

Para a elaboração do nosso aplicativo precisaremos trabalhar com mais de uma tela, desse modo desenvolvemos primeiro a tela 1. Na aba Designer arraste para a área de visualização quatro botões e mude os seus textos para “Média aritmética”, “Média aritmética ponderada”, “Mediana” e “Moda”, respectivamente. Cada um desses botões abrirá uma nova tela, por isso adicione quatro telas e para adicionar vá em “Adicionar Ecrã” na faixa verde superior na aba de Designer, como mostra a figura 34.

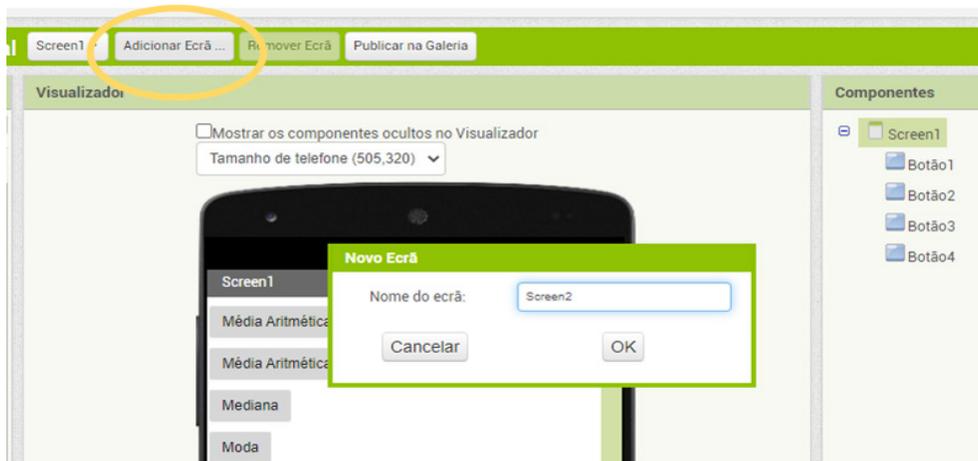


Figura 34 - Adicionando telas ao aplicativo.

Fonte: Elaborado pelos autores.

Agora na aba blocos programe os botões para que quando um botão for clicado ele abra uma outra tela, assim conforme a figura 35, onde os botões estão nomeados conforme a sua função.

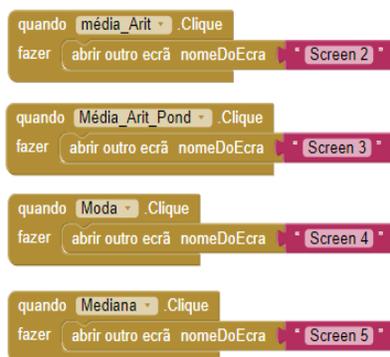


Figura 35 - Blocos do aplicativo III (a).  
Fonte: Elaborado pelos autores.

A tela 1 está pronta, assim vamos para a próxima tela.

### • Tela 2 - Screen 2

Essa tela será responsável pela média aritmética, assim iremos desenvolvê-la para tal. Logo na aba designer insira uma legenda e mude o seu texto para “Qual o número de elementos?”, uma caixa de texto e mude a dica para “Número de elementos”, uma legenda e mude o seu texto para “Quais são os elementos?”, insira também cinco caixas de texto e mude as suas dicas para “elemento 1”, “elemento 2”, ..., “elemento 5”, um botão e altere o seu texto para “Calcular a média aritmética” e por último uma legenda e exclua o seu texto.

Você pode renomear os componentes para facilitar a programação e lembre-se que a média aritmética é dada por  $\frac{x_1+x_2+\dots+x_n}{n}$  e no nosso caso a média aritmética será realizada sobre no máximo cinco elementos, ou seja, a média será calculada sobre um, dois, três, quatro ou cinco elementos.

As cinco caixas de texto inseridas são destinadas para o usuário informar os elementos, porém como nem sempre serão cinco elementos, peça para o usuário atribuir 0 (zero) aos elementos inexistentes. Por exemplo, o usuário deseja calcular a média sobre os elementos 2, 4 e 6, está é dada por  $\frac{2+4+6}{3} = 4$ , o que equivale a  $\frac{2+4+6+0+0}{3} = 4$ .

Observe a figura 35 que representa a interface da tela 2, o que lhe auxiliará na inserção dos componentes.

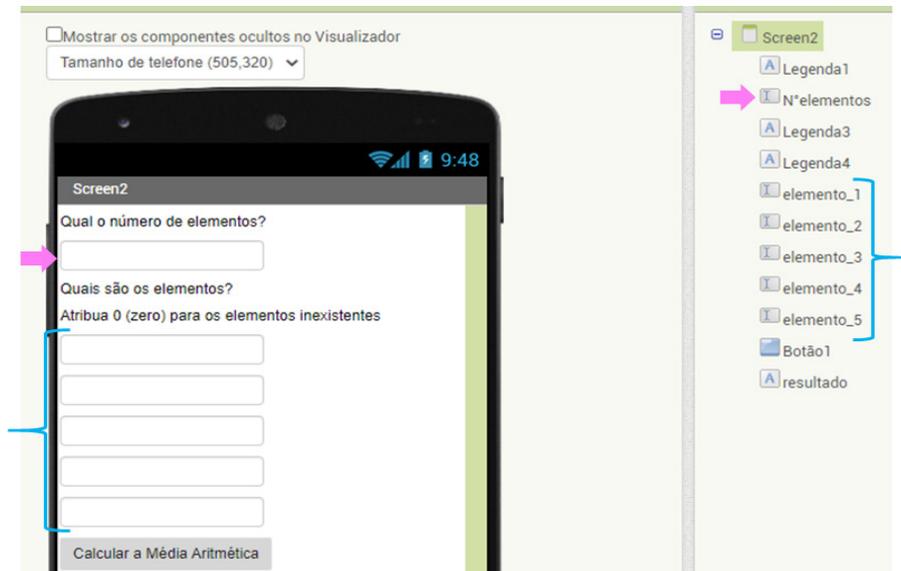


Figura 35 - Interface da Tela 2 do aplicativo III.  
Fonte: Elaborado pelos autores.

Agora na aba blocos reproduza com os blocos a equação da média, de acordo com a figura 36.

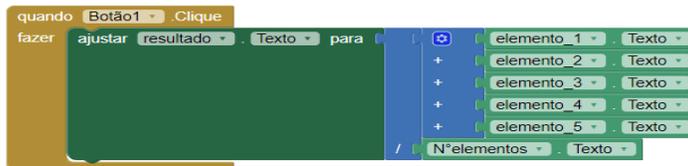


Figura 36 - Blocos do aplicativo III (b).  
Fonte: Elaborado pelos autores.

Nossa tela 2 está pronta, vamos agora desenvolver a tela 3.

### • Tela 3 - Screen 3

A funcionalidade desta tela será calcular a média aritmética ponderada, lembrando que a média aritmética ponderada é dada por  $\frac{x_1 p_1 + x_2 p_2 + \dots + x_n p_n}{p_1 + p_2 + \dots + p_n}$ , onde  $x_i$  é o elemento  $p_i$  e é o seu peso e para facilitar a inserção dos elementos visualize a figura 37.

Na aba designer insira uma legenda e mude o texto para “Na tabela abaixo informe os elementos e os seus respectivos pesos”, abaixo da legenda insira uma organização em tabela com duas colunas e seis linhas, onde na primeira linha e primeira coluna insira uma legenda e mude o texto para “Elemento” e na primeira linha e segunda coluna insira uma legenda e mude o texto para “Peso”.

Na primeira coluna insira cinco caixas de texto e mude a largura delas, pode ser 30%, altere as dicas para “ $x_1$ ”, ..., “ $x_5$ ”, na respectiva ordem das caixas, além disso pode também renomeá-las segundo as suas dicas. Na segunda coluna insira cinco caixas de texto e mude a largura delas, pode ser 20%, altere as dicas para “ $p_1$ ”, ..., “ $p_5$ ”, na respectiva ordem das caixas de texto e caso queira as renomeie.

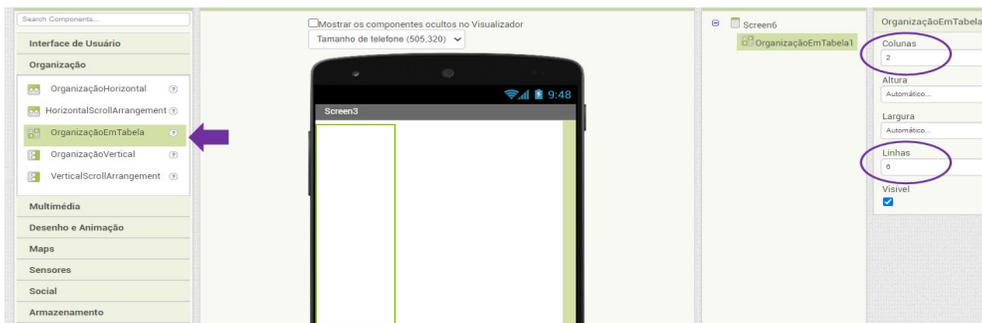


Figura 37 - Interface da tela 3 do aplicativo III (a).  
Fonte: Elaborado pelos autores.

Ainda na aba de designer insira um botão e mude o texto para “Calcular a Média Aritmética Ponderada”, também uma legenda e exclua o seu texto. Observe o botão na figura 38, bem como as propriedades da tabela.

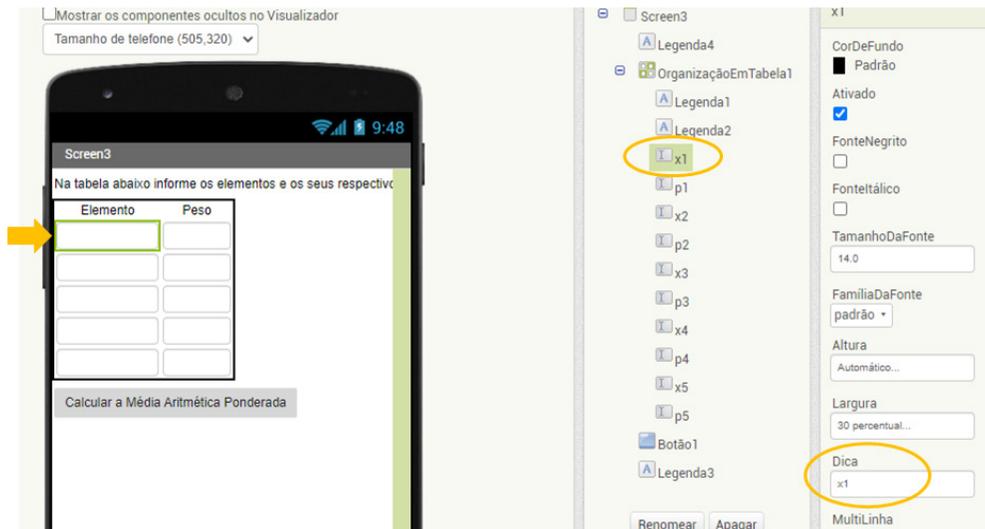


Figura 38 - Interface da tela 3 do aplicativo III (b).  
Fonte: Elaborado pelos autores.

Do mesmo modo como ocorreu na tela 2, aqui o usuário pode querer calcular a média aritmética ponderada de um, dois, três, quatro ou cinco elementos, de modo que quando for menos que cinco elementos o usuário deve atribuir zero ao(s) elemento (s)

inexistente(s) e ao seu peso correspondente.

Já temos os componentes necessários para a tela 3, vamos agora fazer a programação. Na aba blocos reproduza a equação para calcular a média aritmética ponderada, como mostra a figura 39.

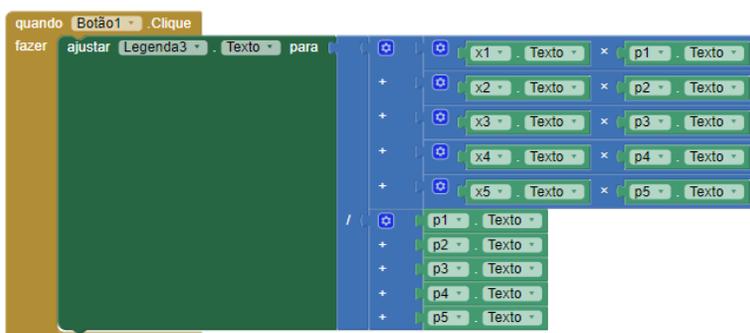


Figura 39 - Blocos do aplicativo III (c).

Fonte: Elaborado pelos autores.

A tela 3 está pronta, seguimos agora para a tela 4.

#### • Tela 4 - Screen 4

A aba designer da tela 4 está representada na figura 40, sendo que a tela 4 será a responsável pela mediana, lembrando que para determinar a mediana de um conjunto estes valores devem ser postos em ordem crescente ou decrescente, de modo que se a quantidade de elementos for ímpar a mediana corresponde ao valor central e quando a quantidade for par a mediana é obtida realizando a média aritmética dos dois elementos centrais.

Primeiramente, na aba designer insira uma legenda e mude o texto para “Qual o número de elementos?”, uma caixa de texto, uma legenda e mude o texto para “Informe os elementos em ordem crescente ou decrescente”, insira cinco caixas de texto e mude as dicas para “elemento 1”, ..., “elemento 5” na respectiva ordem, um botão e mude o texto para “Determinar a Mediana” e por último uma legenda e exclua o seu texto.

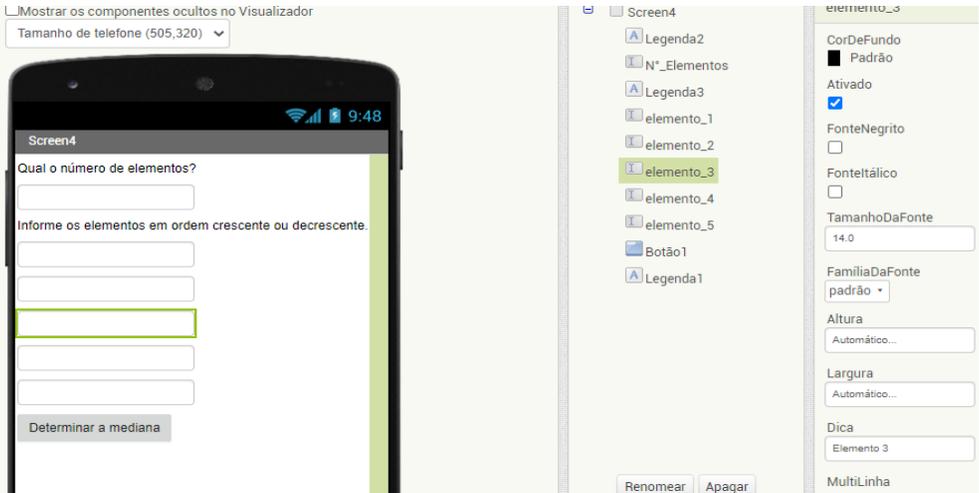


Figura 40 - Interface da tela 4 do aplicativo III.  
Fonte: Elaborado pelos autores.

Na aba blocos programe os blocos segundo o conceito de mediana, de acordo com a figura 41.

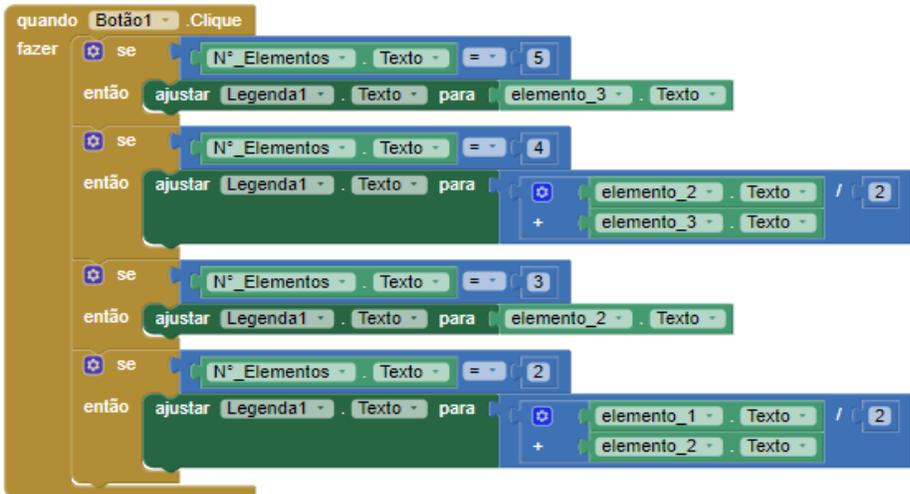


Figura 41 - Blocos do aplicativo III (d).  
Fonte: Elaborado pelos autores.

Nossa tela 4 está pronta!

#### • Tela 5 - Screen 5

Essa tela será responsável por determinar a moda, que é o elemento que mais se repete em um conjunto. Assim, na aba designer insira uma legenda e mude o texto para “Informe os elementos e o número de vezes que ele aparece”, uma legenda e mude o texto

para “Para os elementos inexistentes atribua 0 (zero) ao elemento e a sua frequência”.

Abaixo insira uma organização em tabela com duas colunas e seis linhas, de modo que na primeira linha e primeira coluna insira uma legenda e mude o texto para “Elementos”, na primeira linha e segunda coluna insira uma legenda e mude o texto para “Frequência”, na primeira coluna insira cinco caixas de texto e altere a dica para “ $x_1$ ”, ..., “ $x_5$ ”, respectivamente e na segunda coluna insira cinco caixas de texto e altere a dica para “ $p_1$ ”, ..., “ $p_5$ ”, respectivamente. Para finalizar insira um botão e mude o texto para “Determinar a moda” e também uma legenda e exclua o seu texto, agora observe a interface da tela 5 representada na figura 42.

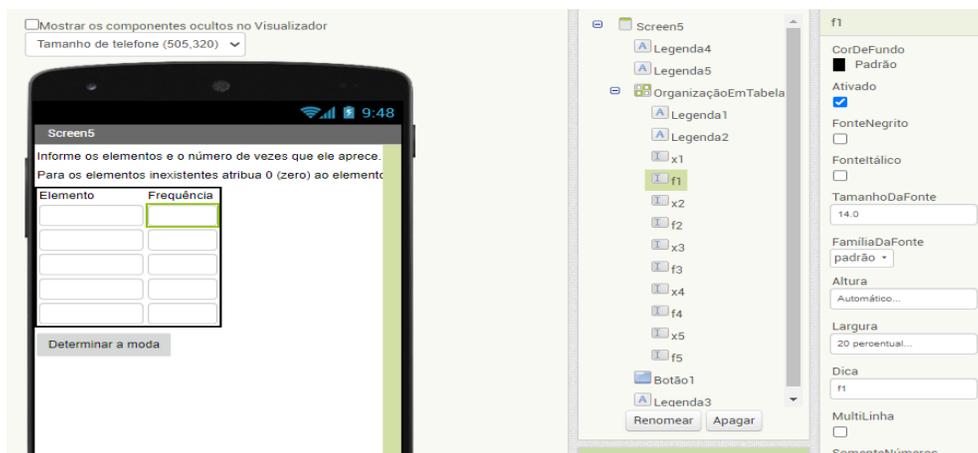


Figura 42 - Interface da tela 5 do aplicativo III.

Fonte: Elaborado pelos autores.

Para concluir a tela 5 encaixe os blocos segundo a figura 43.

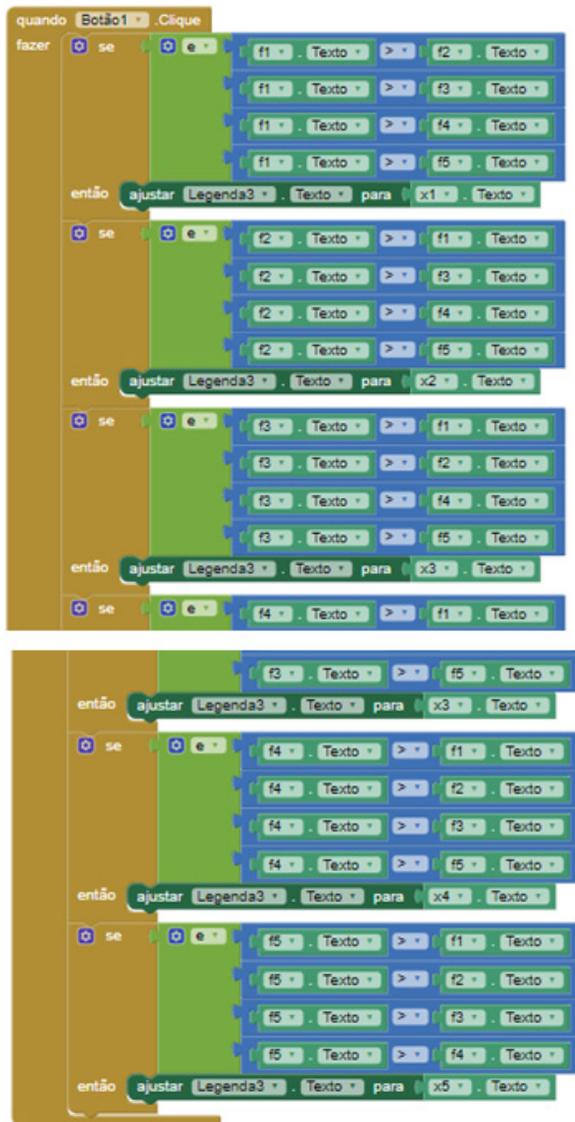


Figura 43 - Blocos do aplicativo III (e).  
 Fonte: Elaborado pelos autores.

### • Finalizando o aplicativo

Nosso aplicativo está basicamente pronto, só precisamos acrescentar alguns botões nas telas para complementar. Nas telas 2, 3, 4 e 5 insira dois botões e programe o primeiro para o usuário informar novos valores e o outro botão para retornar a tela 1. Observe estes botões na tela 2, da figura 44.

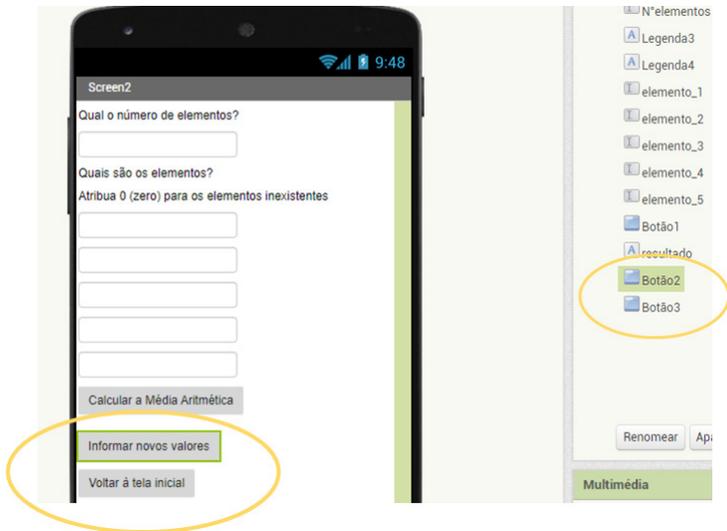


Figura 44 - Finalizando o aplicativo III.

Fonte: Elaborado pelos autores.

Na figura 45 acompanhe a programação desses dois botões acrescentados.

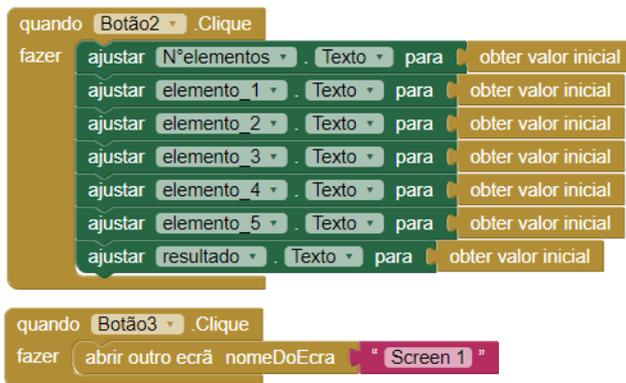


Figura 45 - Blocos do aplicativo III (f).

Fonte: Elaborado pelos autores.

Faça o mesmo nas demais telas e você pode utilizar a sua criatividade e tornar o aplicativo mais apresentável, assim como o exemplo trazido na figura 46.



**Média Aritmética**

Qual o número de elementos?

N° de elementos

Quais são os elementos?

Atribua 0 (zero) para os elementos inexistentes

Elemento 1

Elemento 2

Elemento 3

Elemento 4

Elemento 5

Calcular a Média Aritmética

Informar novos valores

Voltar à tela inicial

**Média Aritmética Ponderada**

Na tabela abaixo informe os elementos e os seus respectivos pesos

Para os elementos inexistentes atribua 0 (zero), bem como para o seu respectivo peso.

Elemento	Peso
x1	p1
x2	p2
x3	p3
x4	p4
x5	p5

Calcular a Média Aritmética Ponderada

Informar novos valores

Voltar à tela inicial

**Mediana**

Qual o número de elementos?

N° de elementos

Informe os elementos em ordem crescente ou decrescente.

Elemento 1

Elemento 2

Elemento 3

Elemento 4

Elemento 5

Determinar a mediana

Informar novos valores

Voltar à tela inicial

**Moda**

Informe os elementos e o número de vezes que ele aparece.

Para os elementos inexistentes atribua 0 (zero) ao elemento e a sua frequência.

Elemento	Frequência
x1	f1
x2	f2
x3	f3
x4	f4
x5	f5

Determinar a moda

Informar novos valores

Voltar à tela inicial

Figura 46 - Aplicativo III concluído.  
Fonte: Elaborado pelos autores.

Pronto! O aplicativo III está concluído e vamos continuar desenvolvendo mais alguns aplicativos.

# APLICATIVO IV: DESCOBRINDO O LADO DESCONHECIDO DE UM TRIÂNGULO RETÂNGULO

Vamos construir um aplicativo matemático, cuja função é descobrir a medida do lado desconhecido de um triângulo retângulo através da fórmula do Teorema de Pitágoras. Para tal acesse em seu navegador o App Inventor 2 e em seguida inicie um novo projeto e o nomeie, como sugestão pode nomear como “descobrimdo o lado desconhecido de um triângulo retângulo”.

## • Na aba designer

O triângulo possui três lados, por isso utilizamos três organizações horizontais, como na figura 47, de modo a inserir dentro de cada, uma legenda e uma caixa de texto, onde o usuário possa informar quais são as medidas conhecidas dos respectivos lados. Em relação às legendas e as caixas de textos, altere o texto das legendas para “hipotenusa”, “cateto 1” e “cateto 2”, respectivamente, depois exclua a dica das caixas de textos e selecione na propriedade de cada caixa, a opção “somente número”. Selecionando esta opção o aplicativo fornecerá ao usuário um teclado somente com números, evitando que o usuário informe por um equívoco letras, ou números negativos como medidas dos lados do triângulo.

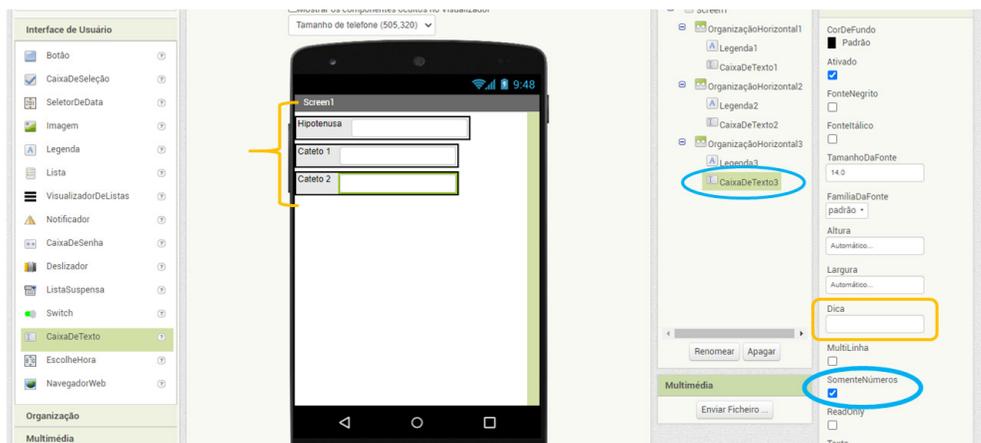


Figura 47 - Interface do aplicativo IV (a).

Fonte: Elaborado pelos autores.

Em seguida, de acordo com a figura 48 insira dois botões e mude os seus textos para “qual a medida do lado desconhecido?” e “limpar”, com o intuito de programá-los para efetuar o cálculo que resulta na medida procurada e limpar as informações da tela do usuário, que forem emitidas e recebidas pelo aplicativo. Insira também uma legenda para informar o resultado (medida procurada) e em propriedades, exclua o seu texto.

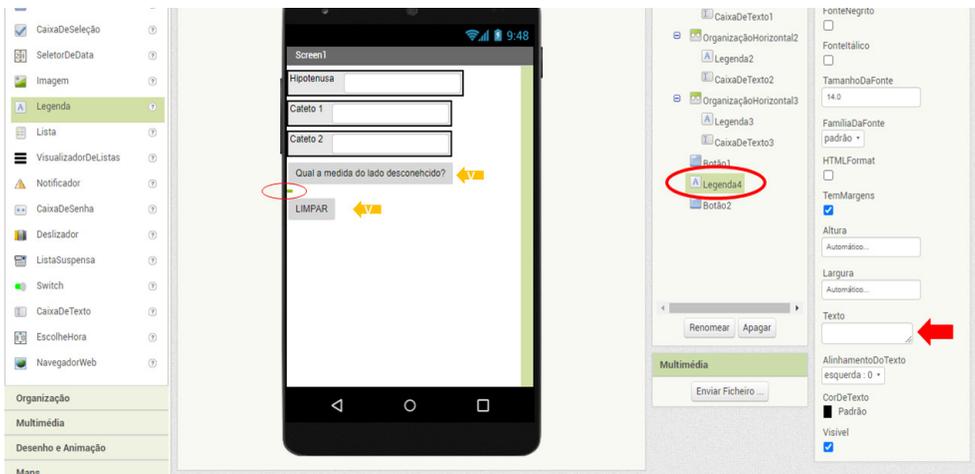


Figura 48 - Interface do aplicativo IV (b).  
Fonte: Elaborado pelos autores.

Com todos os componentes necessários para o funcionamento do aplicativo inseridos vamos para a aba blocos, a fim de programar as suas funções.

• **Na aba blocos**

Queremos que quando o “Botão 1” for clicado o aplicativo nos dê a medida do lado desconhecido e para isso iremos utilizar a fórmula do Teorema de Pitágoras  $(hipotenusa)^2 = (cateto\ 1)^2 + (cateto\ 2)^2$ , lembrando que a hipotenusa é o maior lado do triângulo, portanto a medida do cateto 1 e a medida do cateto 2 devem ser menores que a medida da hipotenusa. Logo temos quatro situações possíveis de lados desconhecidos:

- I. O lado desconhecido é a hipotenusa  $\rightarrow hipotenusa = \sqrt{(cateto\ 1)^2 + (cateto\ 2)^2}$
- II. O triângulo retângulo é um triângulo isósceles, ou seja, seus dois catetos possuem as mesmas medidas e que neste caso são desconhecidas  $\rightarrow cateto\ 1 = cateto\ 2 = \sqrt{\frac{(hipotenusa)^2}{2}}$ .
- III. O lado desconhecido é o cateto 1  $\rightarrow cateto\ 1 = \sqrt{(hipotenusa)^2 - (cateto\ 2)^2}$ .
- IV. O lado desconhecido o cateto 2  $\rightarrow cateto\ 2 = \sqrt{(hipotenusa)^2 - (cateto\ 1)^2}$ .

Para cada uma das situações utilizamos um bloco de controle “se-então” e as respectivas fórmulas dadas acima. Observe a programação para a primeira situação dada na figura 49.

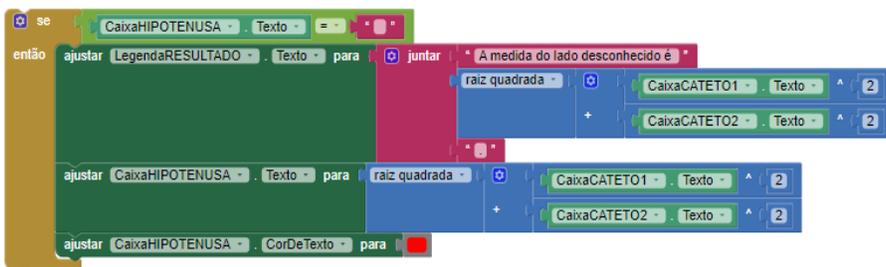


Figura 49 - Blocos do aplicativo IV (a).  
 Fonte: Elaborado pelos autores.

Na primeira situação o lado procurado é a hipotenusa, para isso utilizamos como condicionante que o texto na caixa destinada para a hipotenusa seja vazio, pois o usuário deve informar somente as medidas conhecidas, deixando a caixa da medida desconhecida sem nada digitado. Foi programado para que a medida procurada apareça na legenda destinada ao resultado, que se encontra abaixo do botão para calcular e também na caixa de texto do lado procurado, porém na cor vermelha.

O mesmo foi realizado para as demais situações conforme a figura 50.



Figura 50 - Blocos do aplicativo IV (b).  
 Fonte: Elaborado pelos autores.

Agora pegue o bloco “Quando botão calcular. Clique fazer”, esse bloco nos diz que quando o botão calcular for clicado ele irá realizar os passos determinados pelos blocos que serão encaixados dentro dele. Assim encaixamos os blocos como mostra a figura 51.

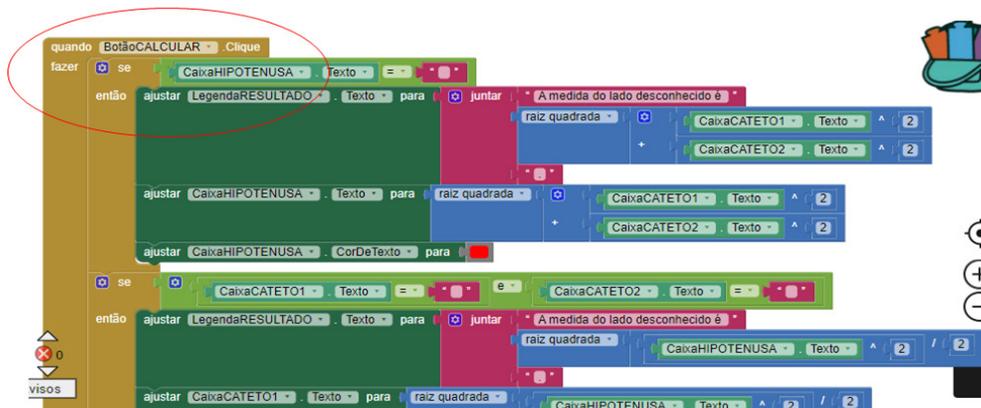


Figura 51 - Blocos do aplicativo IV (c).  
Fonte: Elaborado pelos autores.

O aplicativo já está programado para calcular e apresentar a medida desconhecida do lado triângulo retângulo, agora vamos programar o botão limpar conforme a figura 52. Queremos que quando o botão limpar for clicado, tudo o que foi digitado pelo usuário no aplicativo e o resultado que o aplicativo informou sejam excluídos, ou seja, o aplicativo apresentará novamente a tela de início.

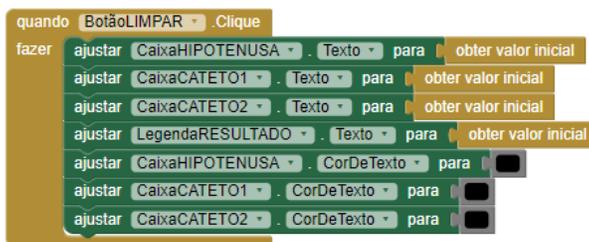


Figura 52 - Blocos do aplicativo IV (d).  
Fonte: Elaborado pelos autores.

Ao clicar nesse botão o texto das caixas de texto e da legenda destinada ao resultado voltarão para o valor inicial, onde o valor inicial corresponde a uma caixa de texto vazia. Importante salientar, que ao invés de utilizar o bloco de “obter valor inicial” pode-se utilizar o bloco dos textos destinado a escrever texto, porém com ele vazio (sem nada escrito). Como durante o uso do aplicativo as cores de algumas caixas de textos mudam para o vermelho, programamos para que a cor de todas as caixas de textos retorne ao preto.

Assim, nosso aplicativo já está pronto, agora você deve testá-lo para ver se o mesmo

está funcionando conforme o esperado e pode incrementá-lo. Observe a figura 53, onde a cor de alguns textos foi modificada, o Screen 1 foi renomeado para “lado desconhecido de um triângulo retângulo”, a organização horizontal foi alterada para se localizar no “centro”. Acima das três organizações horizontais inserimos uma legenda, com o intuito de orientar o usuário que deve informar as medidas dos lados conhecidos e abaixo dos componentes já inseridos foi inserida uma imagem, da qual não deixamos visível, de modo que se torne visível quando o botão calcular for clicado.

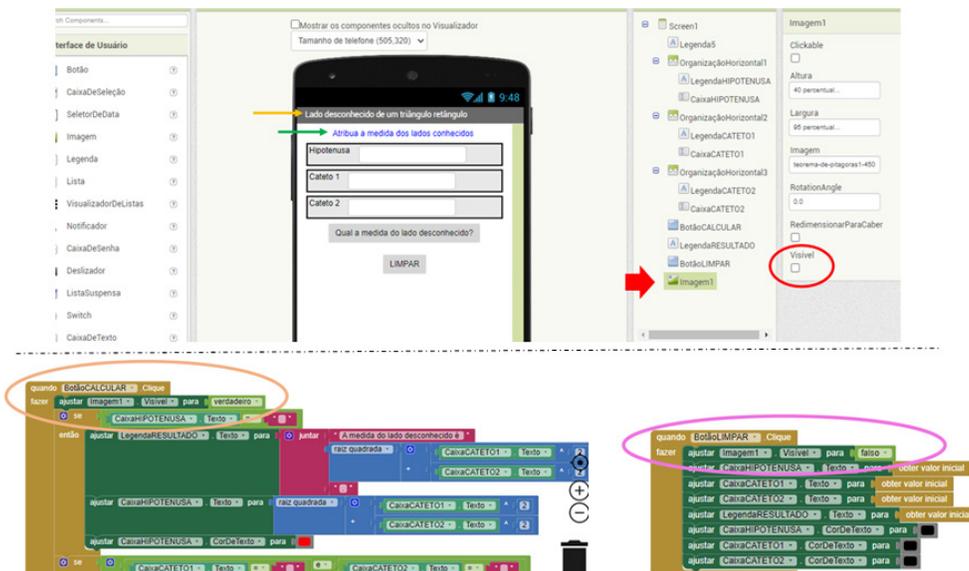


Figura 53 - Incrementando o aplicativo IV.  
Fonte: Elaborado pelos autores.

Além destas modificações, você pode alterar e acrescentar mais coisas deixando o aplicativo mais atraente, como mostra a figura 54.

Lado desconhecido de um triângulo retângulo

Atribua a medida dos lados conhecidos

Hipotenusa

Cateto 1

Cateto 2

Qual a medida do lado desconhecido?

A medida do lado desconhecido é 5.

LIMPAR

Lado desconhecido de um triângulo retângulo

Atribua a medida dos lados conhecidos

Hipotenusa

Cateto 1

Cateto 2

Qual a medida do lado desconhecido?

A medida do lado desconhecido é 32.72614.

LIMPAR

Figura 54 - Aplicativo IV concluído.  
 Fonte: Elaborado pelos autores.

## APLICATIVO V: ENCONTRANDO AS RAÍZES DA EQUAÇÃO DE 2º GRAU

Vamos construir agora um aplicativo matemático que encontre as raízes da equação de segundo grau, semelhante ao aplicativo IV que encontra a medida desconhecida do lado de um triângulo retângulo. Assim, acesse a plataforma do App Inventor e em seguida inicie um novo projeto e o nomeie, pode chamá-lo de “RaízesDaEquaçãoDe2Grau”.

### • Na aba Designer

O objetivo do aplicativo é informar quais são as raízes de uma equação do segundo grau após ele conhecer os seus coeficientes, os quais devem ser informados pelo usuário. Para criar um espaço destinado ao usuário informar os coeficientes insira uma organização horizontal, dentro da organização arraste três legendas e três caixas de texto, intercalando, primeiro legenda e depois caixa de texto. Depois altere o texto das legendas para “a”, “b” e “c”, respectivamente, correspondendo aos três coeficientes da equação de segundo grau  $ax^2 + bx + c = 0$ , também exclua o texto das caixas de texto e para facilitar a programação renomeie as caixas de texto, pode chamá-las de “caixaA”, “caixaB” e “caixaC”.

Em seguida, como na figura 54, acima da organização insira uma legenda e mude o texto para “Informe os coeficientes”. Além disso, o título da sua tela pode ser alterado, clicando em Screen1 que é o nome atual da sua tela e em suas propriedades altere o título, como sugestão pode colocar “Equação de 2º grau:  $ax^2 + bx + c = 0$ ”.

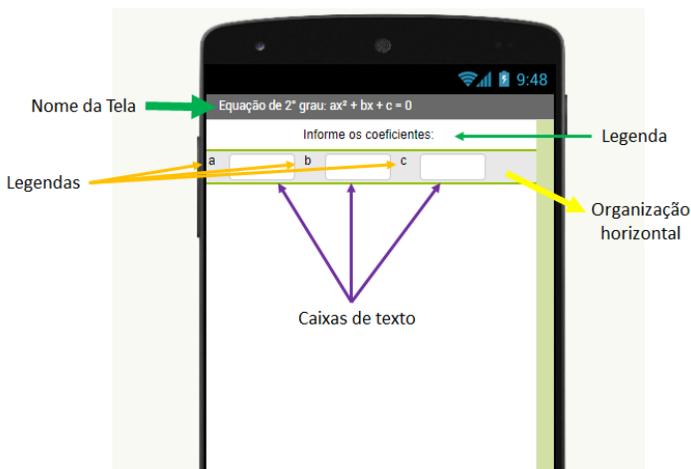


Figura 55 - Interface do aplicativo V (a).  
Fonte: Elaborado pelos autores.

O aplicativo já possui espaço para receber as informações do usuário, agora precisamos criar um espaço para o aplicativo emitir informações, ou seja, para ele informar

quais as raízes da equação cujos coeficientes foram informados pelo usuário. Para isso arraste um botão para a área de visualização, preferencialmente abaixo dos componentes já inseridos, este botão terá como função avisar o aplicativo que ele deve calcular as raízes e informá-las.

Ainda referente ao botão, mude o texto para “Quais são as raízes da equação?” e caso queira renomeie. Em seguida, abaixo do botão insira duas organizações horizontais e em cada organização arraste duas legendas, mude o texto das primeiras legendas dessas organizações, para “Raiz 1:” e “Raiz 2:”, respectivamente e das segundas legendas apague o texto e as renomeie, como mostra a figura 55.

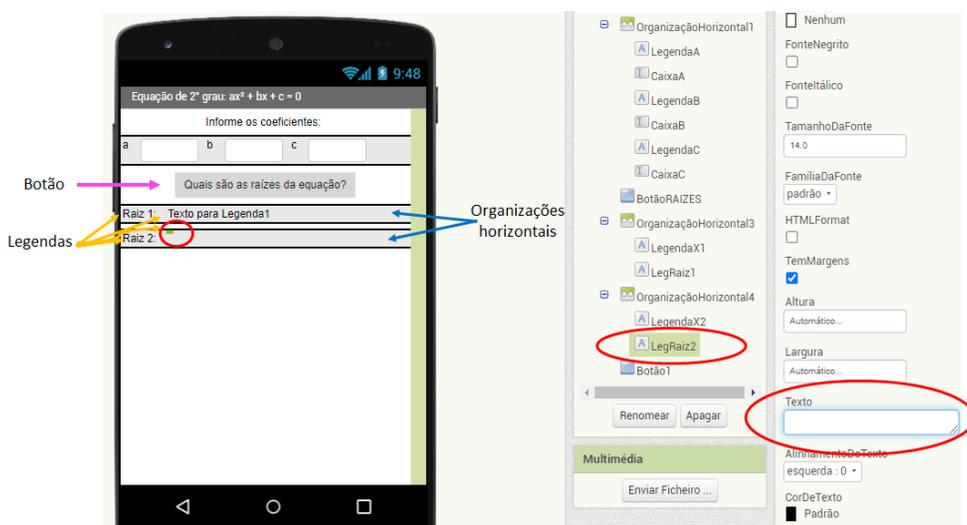


Figura 56 - Interface do aplicativo V (b).  
Fonte: Elaborado pelos autores.

Para facilitar o uso do aplicativo para uma nova equação insira mais um botão e mude o seu texto para “limpar”. Observe a figura 56 que traz o botão limpar inserido e algumas propriedades dos componentes que foram alteradas.

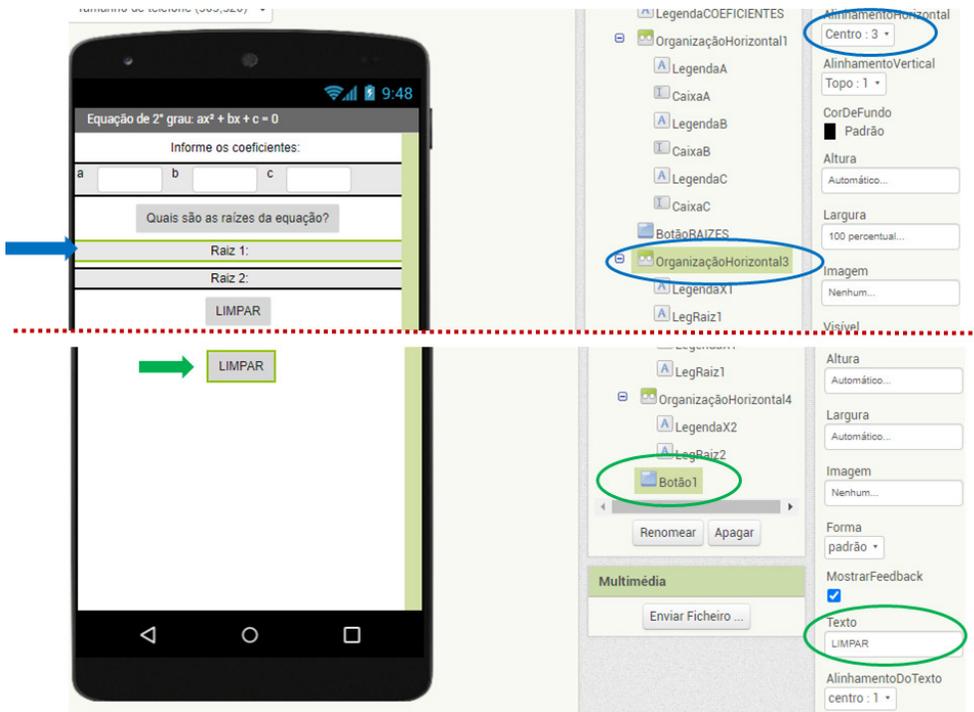


Figura 57 - Interface do aplicativo V (c).  
 Fonte: Elaborado pelos autores.

Nosso aplicativo já possui todos os componentes necessários para o seu funcionamento, agora vamos programá-los, mas para isso precisamos levar algumas condições em consideração.

#### • Condições

Sabendo que a equação de segundo tem o formato  $ax^2 + bx + c = 0$ , onde  $a$ ,  $b$  e  $c$  são coeficientes reais e  $a \neq 0$ , tem-se que:

- Se  $a = 0$  não teremos uma equação do segundo grau;
- Se  $b = 0$  teremos  $x = \pm \sqrt{\frac{-c}{a}}$ , isso implica que a raiz 1 será dada por  $\sqrt{\frac{-c}{a}}$  e a raiz 2 será  $-\sqrt{\frac{-c}{a}}$ . De modo que, para essas duas raízes pertencerem aos números reais, precisamos ter:
  - Se  $c$  for maior ou igual a zero, então  $a$  deve ser menor que zero;
  - Se  $c$  menor ou igual a zero, então  $a$  deve ser maior que zero;
- Se  $b = 0$  e a condições dadas em b) não forem cumpridas, a equação de 2º grau não possui raiz real;
- Se  $c = 0$  temos  $ax^2 + bx = 0 \rightarrow x(ax + b) = 0$ , assim a raiz 1 é nula (e a raiz 2 é

dada por  $\frac{-b}{a}$ ;

- e. Para facilitar a resolução das equações de segundo grau temos a fórmula resolvente, sendo x dado por:  $x = \frac{-b \pm \sqrt{\Delta}}{2a}$ , onde  $\Delta = b^2 - 4ac$ . A fórmula resolvente pode ser utilizada para resolver qualquer equação de 2º grau;
- f. Se  $\Delta < 0$  a equação não possui raízes reais;
- g. Se  $\Delta = 0$ , então  $raíz\ 1 = raíz\ 2 = \frac{-b}{2a}$ ;
- h. Se  $\Delta > 0$ , então  $raíz\ 1 = \frac{-b + \sqrt{\Delta}}{2a}$  e  $raíz\ 2 = \frac{-b - \sqrt{\Delta}}{2a}$ .

Depois de todas as condições e aspectos matemáticos sobre as equações de 2º grau serem analisados estamos prontos para programar o aplicativo.

#### • Na aba Blocos

Agora iremos apresentar dois modos de programar os componentes, cabe a você escolher o modo que achar conveniente, porém apesar de poder optar por um ou outro modo é importante que leia e procure compreender como ocorre a programação dos componentes nos dois modos, isso aumentará sua bagagem de conhecimento acerca da programação dos componentes podendo lhe auxiliar na programação de futuros aplicativos.

O modo I explora detalhadamente todas as condições matemáticas apresentadas anteriormente, sendo uma programação um tanto extensa. Já o modo II faz o uso da fórmula resolvente, onde criando um procedimento para o Delta a programação se torna simples e prática. Veja os dois modos a seguir.

#### Modo I

Clique no componente “bloco Raízes” e arraste o bloco “quando “BotãoRAIZES”. Clique fazer” para a área de visualização. Esse bloco nos diz que quando o botão for clicado ele irá realizar os passos determinados pelos blocos que serão encaixados a ele.

Agora iremos utilizar as condições citadas anteriormente, de modo que para condição utilizaremos um “bloco de controle - se então” que posteriormente serão encaixados dentro do “bloco raízes” já inserido.

Da condição 1, temos que se a for igual a zero a equação não possui raízes reais. Traduzindo isso para os componentes do aplicativo, segue que **se** o texto da “caixaA” for igual a zero, **então** as legendas “LegRaiz1” e “LegRaiz2” mudaram o seu texto para “ não é uma equação do 2º grau”, conforme a figura 57.



Figura 58 - Blocos do aplicativo V (a).  
Fonte: Elaborado pelos autores.

Seguindo as condições dadas em (b), monte os blocos de acordo com a figura 59.



Figura 59 - Blocos do aplicativo V (b).  
Fonte: Elaborado pelos autores.

Observe na figura 59 que se a condição dada em b) for cumprida, o texto das legendas (raiz 1 e 2) serão alteradas para  $\sqrt{\frac{-c}{a}}$  e  $-\sqrt{\frac{-c}{a}}$ , respectivamente, lembrando que os blocos azuis são do item matemática e os blocos em verde claro se encontram no item lógica. E da condição c), temos a montagem representada na figura 60.



Figura 60 - Blocos do aplicativo V (c).  
Fonte: Elaborado pelos autores.

Da condição d) segue a figura 61.



Figura 61 - Blocos do aplicativo V (d).  
Fonte: Elaborado pelos autores.

Agora realize a programação dos blocos em relação a condição e), observando a figura 61 que contém as duas primeiras partes da programação.

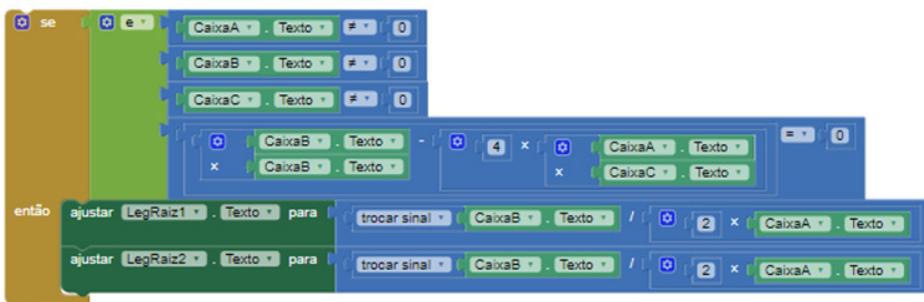


Figura 62 - Blocos do aplicativo V (e).  
 Fonte: Elaborado pelos autores.

Agora observando a figura 63, conclua a programação referente a condição e).

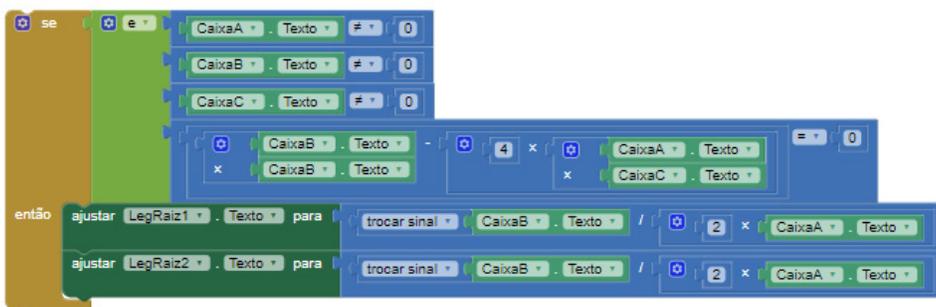


Figura 63 - Blocos do aplicativo V (f).  
 Fonte: Elaborado pelos autores.

Agora só precisamos juntar todos os blocos das figuras 61 e 62.

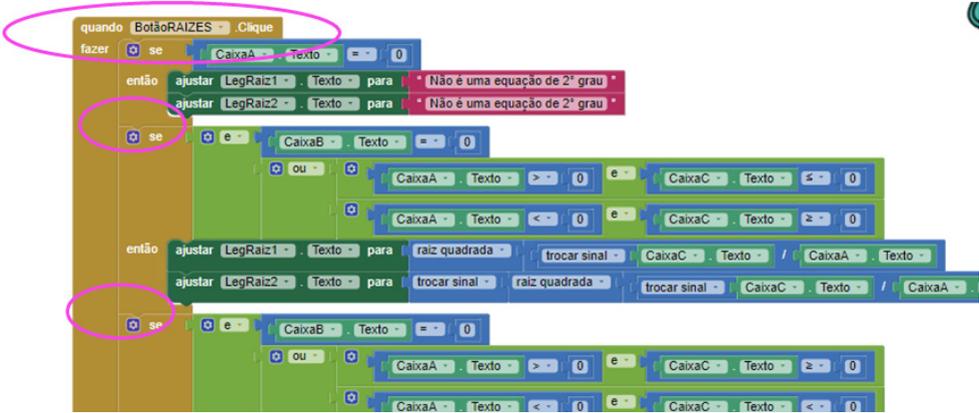


Figura 64 - Blocos do aplicativo V (e).  
 Fonte: Elaborado pelos autores.

O aplicativo já está pronto para informar quais são as raízes da equação de 2º grau, após os coeficientes serem informados e o botão raízes for clicado. Assim, o primeiro modo de programar os blocos foi concluído e vamos agora apresentar o segundo modo de programar os blocos.

### Modo II

Primeiramente crie um procedimento para o cálculo do delta, conforme apresentado na figura 64.

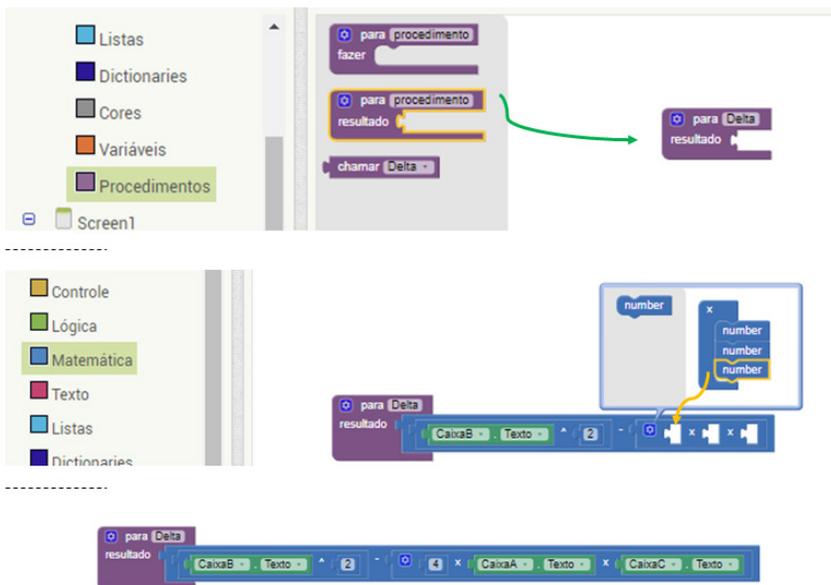


Figura 65 - Bloco do aplicativo V (f).  
 Fonte: Elaborado pelos autores.

Queremos que quando o “BotãoRAIZES” for clicado as raízes da equação sejam informadas, para isso reproduzimos a fórmula resolutiva com os blocos, conforme a figura 65.



Figura 66 - Bloco do aplicativo V (g).  
Fonte: Os autores.

O modo II de programação está concluído com o nosso aplicativo pronto para informar as raízes da equação de segundo grau dada pelo usuário.

• **Complementando o aplicativo**

Tanto no modo I quanto no modo II você deve programar o botão limpar como na figura 66, de modo que quando ele for clicado os dados informados pelo autor e pelo aplicativo sejam excluídos, tornando o aplicativo pronto para informar as raízes de uma nova equação de 2º grau.

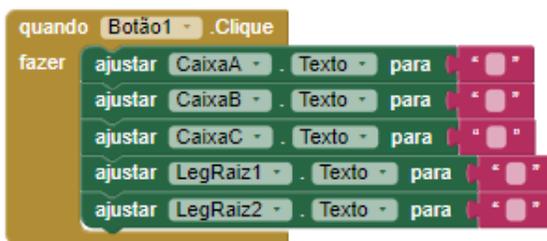


Figura 67 - Bloco do aplicativo V (h).  
Fonte: Elaborado pelos autores.

Veja na figura 67 a interface do aplicativo pronto, lembrando que você pode personalizar o seu aplicativo a fim de torná-lo atrativo visualmente. Além do mais, não esqueça de testar seu aplicativo, para saber se ele está funcionando conforme o desejado.

Equação de 2º grau:  $ax^2 + bx + c = 0$

Informe os coeficientes:

$a = 4$   $b = 12$   $c = 0$

Para descobrir as raízes da equação clique no botão abaixo:

Quais são as raízes da equação?

**Raiz 1:** 0  
**Raiz 2:** -3

LIMPAR

Equação de 2º grau:  $ax^2 + bx + c = 0$

Informe os coeficientes:

$a = 3$   $b = 0$   $c = 14$

Para descobrir as raízes da equação clique no botão abaixo:

Quais são as raízes da equação?

**Raiz 1:** A equação não possui  $x'$  real  
**Raiz 2:** A equação não possui  $x''$  real

LIMPAR

Figura 68 - Aplicativo V concluído.  
Fonte: Elaborado pelos autores.

# APLICATIVO VI: OPERAÇÕES ENTRE MATRIZES E O DETERMINANTE DE UMA MATRIZ

Sabemos quanto o estudo das matrizes é importante, bem como os demais conteúdos matemáticos e a fim de facilitar os cálculos envolvendo matrizes iremos construir aqui um aplicativo que resolve adição, subtração e multiplicação entre matrizes com no máximo 3 linhas e 3 colunas e que além disso calcule o determinante de matrizes  $1 \times 1$ ,  $2 \times 2$  e  $3 \times 3$ .

## • Informações a serem consideradas

Primeiramente há algumas informações que precisam ser levadas em consideração, sendo que para a elaboração deste aplicativo precisaremos trabalhar com seis telas no App Inventor 2, de modo a facilitar a programação e tornar o aplicativo organizado. As três primeiras telas consideradas como principais, terão a seguinte organização:

- Tela 1: Será a tela inicial, onde o usuário irá definir se ele deseja efetuar operações (adição, subtração e multiplicação) ou calcular o determinante;
- Tela 2: Contemplará as três operações entre matrizes, a adição, a subtração e a multiplicação;
- Tela 3: A terceira tela será direcionada ao cálculo do determinante.
- O elaborador do aplicativo deve possuir conhecimento acerca de matrizes, em especial sobre:
- Representação genérica de uma matriz;
- Adição e subtração de matrizes;
- Multiplicação entre matrizes, lembrando que a multiplicação de duas matrizes só é possível quando o número de colunas da primeira é igual ao número de linhas da segunda matriz;
- Cálculo do determinante de matrizes  $1 \times 1$ ,  $2 \times 2$  e  $3 \times 3$ , bem como a regra de Sarrus.

Agora levando em consideração as informações expostas vamos desenvolver o aplicativo, de acordo com cada uma das seis telas.

## • Tela 1 - Screen1

Siga os passos de acesso ao App Inventor 2, crie um novo projeto e o nomeie, assim você será direcionado para a aba de designer que contém a primeira tela (screen 1). Na aba designer, conforme representada na figura 69, arraste dois botões para a área de visualização, altere o texto do primeiro para “Adição, Subtração e Multiplicação de Matrizes” e o renomeie para “operações”, também altere o texto do segundo botão para “Determinante” e renomeie para “determinante”, depois acima dos botões arraste uma legenda e mude o

seu texto para “Faça a sua escolha clicando em um dos botões”.

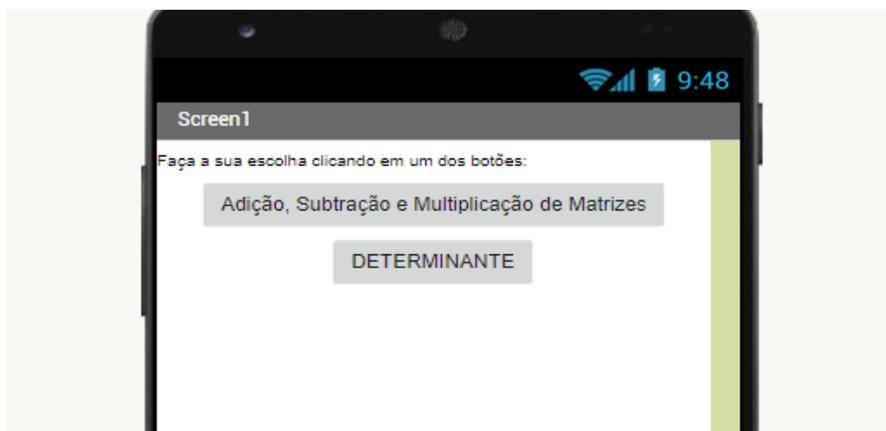


Figura 69 - Interface da tela 1 do aplicativo VI.  
Fonte: Elaboração dos autores.

Agora na aba blocos queremos que o “botão operações” nos leve a tela 2 e que o “botão determinante” nos leve a tela 3, para isso faça como na figura 70.

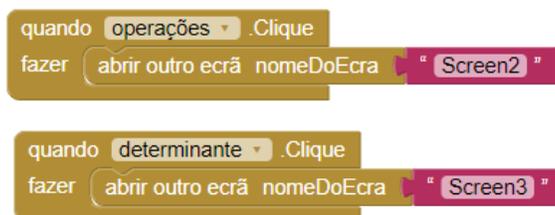


Figura 70 - Blocos do aplicativo VI (a).  
Fonte: Elaboração dos autores.

Assim, nossa primeira tela está programada, vamos agora programar a segunda tela.

#### • Tela 2 - Screen 2

Na aba Designer você precisa adicionar mais uma tela clicando em “Adicionar Ecrã”, como mostra a figura 71.



Figura 71 - Adicionando tela no aplicativo VI.  
Fonte: Elaborado pelos autores.

Nosso aplicativo efetuará operações entre duas matrizes com uma, duas ou três linhas e/ou colunas, ou seja, a maior ordem possível da matriz para o aplicativo realizar operações é 3x3. Chamaremos a primeira matriz de A e a segunda de matriz B, onde você deve saber a relembrar a forma de representação genérica de uma matriz.

Assim, precisamos criar na aba de designer um espaço para o usuário informar as duas matrizes, cuja representação se dá na figura 72, assim para a primeira matriz faça:

- Insira uma organização em tabela e em propriedades mude o número de linhas e de colunas para 3;
- Dentro da tabela insira nove caixas de texto;
- Como as caixas de texto excedem o espaço da tabela mude a largura das caixas de texto para 20 ou 30 por cento;
- Apague o texto e coloque como dica de cada caixa de texto o elemento genérico da matriz A o qual ela representa. Por exemplo, a primeira caixa de texto está na posição do elemento  $a_{11}$ , portanto a dica desta caixa de texto será  $a_{11}$ ;
- Renomeie as caixas de textos, sendo o seu novo nome o mesmo que a dica da caixa de texto;
- Acima da matriz A (organização horizontal), insira uma legenda e mude o seu texto para “Informe os coeficientes da 1ª matriz (matriz A), atribuindo 0 (zero) para os valores inexistentes”.



Figura 72 - Interface da tela 2 do aplicativo VI (a).  
Fonte: Elaborado pelos autores.

Agora vamos criar o espaço para a matriz B, para isso você pode seguir os mesmos passos realizados para a matriz A, ou copiar (Ctrl C) e colar (Ctrl V) a matriz A e somente alterar as dicas das caixas de textos e os seus respectivos nomes e nas propriedades da tela 2 (Screen 2) clique na opção rolável.

Agora o aplicativo já possui um espaço para receber as duas matrizes, a fim de efetuar as operações. Seguindo, precisamos proporcionar ao usuário a opção de escolha da operação desejada (adição, subtração ou multiplicação) além de um espaço para o aplicativo informar o resultado (matriz C). Então, observando a figura 73, faça:

- Insira três botões;
- Altere o texto e os renomeie para “adição”, “subtração” e “multiplicação”, respectivamente;
- Para a matriz resultado insira uma organização em tabela, dentro desta organização insira nove legendas e exclua o seu texto e as renomeie conforme os elementos de uma matriz genérica C.

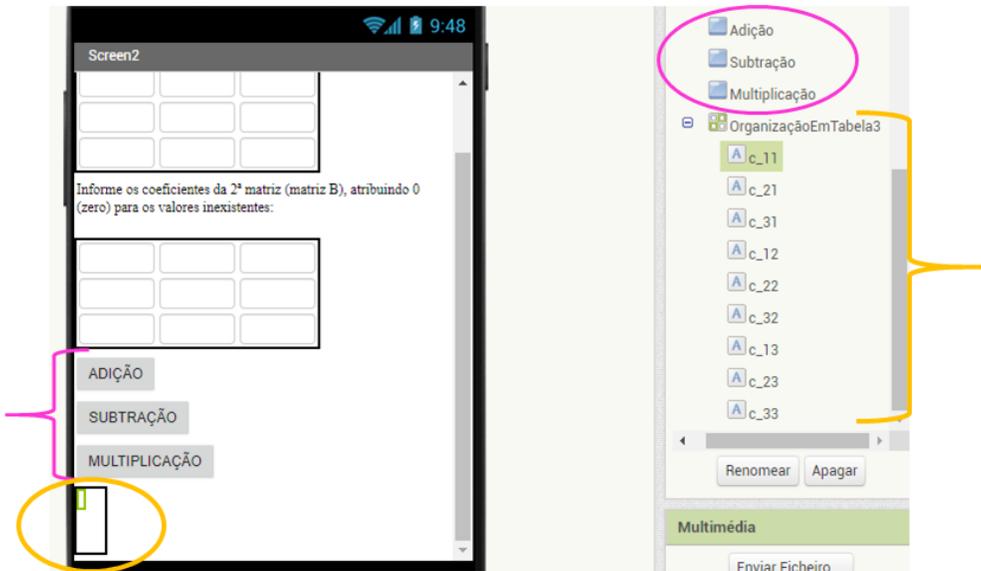


Figura 73 - Interface da tela 2 do aplicativo VI (b).  
 Fonte: Elaborado pelos autores.

Temos todos os componentes necessários para o funcionamento da tela 2, agora precisamos programá-los, mas antes lembre-se de como ocorre a adição, subtração e multiplicação entre duas matrizes A e B.

Seguindo, na aba blocos queremos que conforme cada botão for clicado apresente o respectivo resultado, desse modo reproduzimos com os blocos o processo das operações entre matrizes. Primeiramente programamos o botão adição, de acordo com a figura 74.

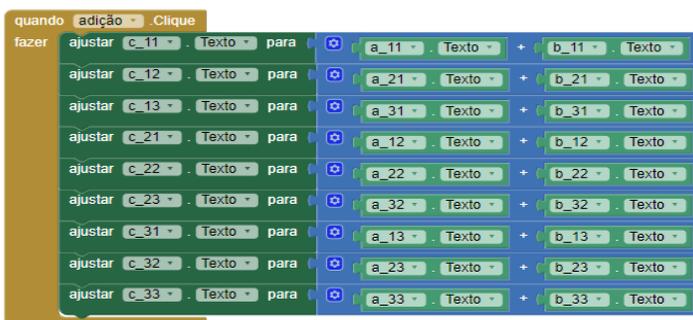


Figura 74 - Blocos do aplicativo VI (b).  
 Fonte: Elaboração dos autores.

Agora programamos o botão subtração como mostra a figura 75.

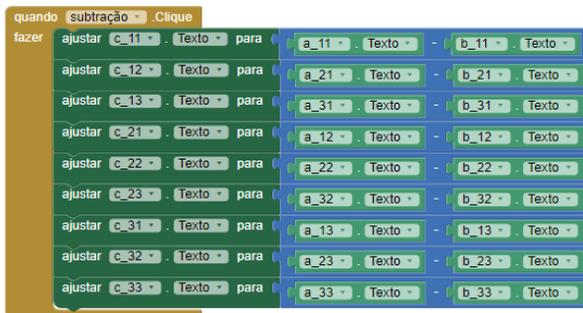


Figura 75 - Blocos do aplicativo VI (c).  
 Fonte: Elaborado pelos autores.

Agora programamos o botão multiplicação, de acordo com a figura 76.

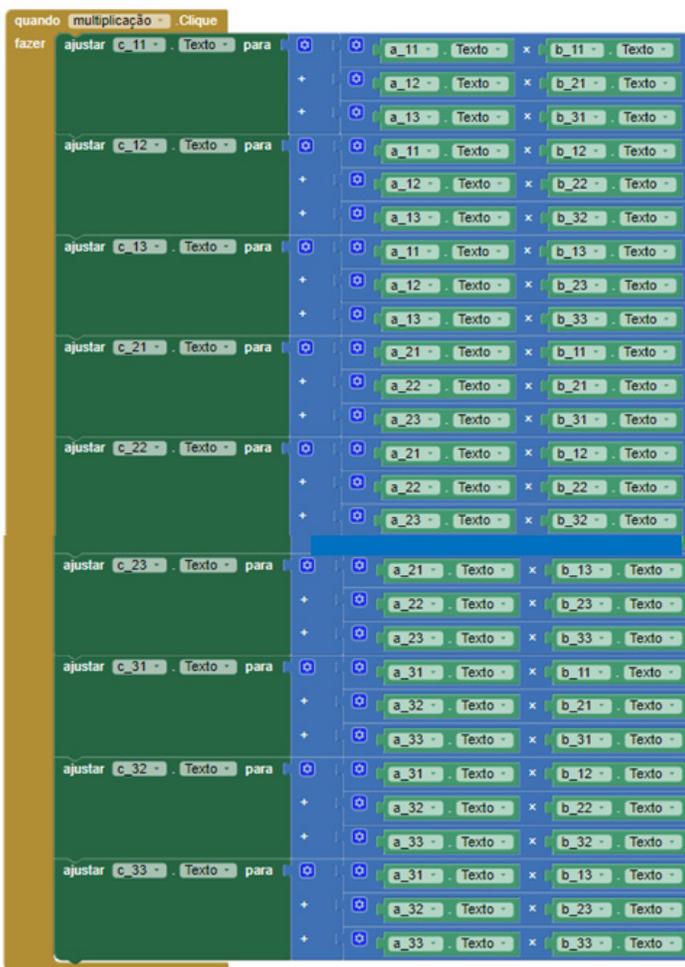


Figura 76 - Blocos do aplicativo VI (d).  
 Fonte: Elaborado pelos autores.

Enfim, a nossa tela 2 está programada. Vamos agora para a tela 3.

### • Tela 3 - Screen 3

Primeiramente crie uma nova tela, a tela 3. O nosso aplicativo trabalha com matrizes de ordem máxima 3x3 e o determinante só é calculado sobre matrizes quadradas (mesmo número de linhas e colunas), assim nosso aplicativo efetuará o cálculo do determinante de matrizes 1x1, 2x2 e 3x3.

Precisamos dar ao usuário a opção de escolher o tipo de matriz da qual deseja descobrir o determinante, portanto arraste para a área de visualização três botões e mude os seus textos para “matriz 1x1”, “matriz 2x2” e “matriz 3x3”, como mostra a figura 77, além disso você também pode renomeá-los.



Figura 77 - Interface da tela 3 do aplicativo VI.  
Fonte: Elaborado pelos autores.

Seguidamente adicione três telas e permaneça na tela 3. Permanecendo na tela 3, vá para a aba blocos e programe os botões segundo a figura 78.



Figura 78 - Blocos do aplicativo VI (e).  
Fonte: Elaborado pelos autores.

Nossa terceira tela está pronta, sendo que cada uma das três telas adicionadas se refere a um dos três tipos de matrizes quadradas trabalhadas aqui.

#### • Tela 4 - Screen 4

- Na aba designer arraste para a área de visualização na ordem dada:
- Uma legenda e mude o seu texto para “informe matriz”;
- Caixa de texto e mude a sua dica para “a\_11”;
- Um botão e altere o seu texto para “Calcular o Determinante”;
- Uma legenda e exclua o seu texto. Essa legenda será utilizada para informar o resultado;

Caso queira pode também renomear os componentes adicionados de acordo com a respectiva função, tendo que a interface da tela 4 pode ser visualizada na figura 79.

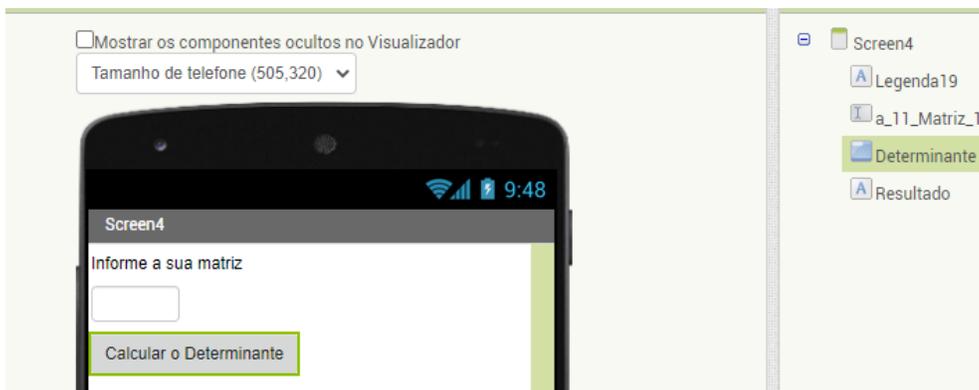


Figura 79 - Interface da tela 4 do aplicativo VI.  
Fonte: Elaboração dos autores.

O trabalho na aba de designer está pronto, vamos para aba blocos, lembrando que o determinante de uma matriz  $1 \times 1$  é o próprio elemento que compõe essa matriz. Portanto, programe a tela 4 segundo a figura 80.



Figura 80 - Blocos do aplicativo VI (f).  
Fonte: Elaborado pelos autores.

Com a tela 4 pronta seguimos para a tela 5.

#### • Tela 5 - Screen 5

Na aba Designer adicione os seguintes componentes na ordem descrita, lembrando que essa tela trabalhará com a matriz  $2 \times 2$  e que a sua interface está representada na figura 81.

- Uma legenda e mude o seu texto para “informe a matriz”;
- Uma organização em tabelas de duas linhas e duas colunas;
- Dentro de cada organização em tabelas insira quatro caixas de texto e altere a sua dica para o respectivo coeficiente que ela representa dentro de uma matriz 2x2;
- Um botão e mude o seu texto para “Calcular o Determinante”;
- Uma legenda e exclua o seu texto.

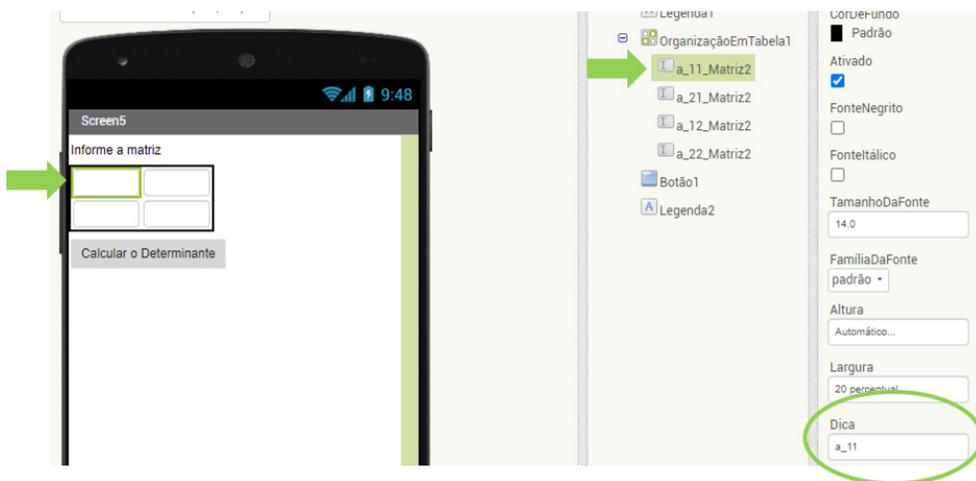


Figura 81 - Interface da tela 5 do aplicativo VI.  
Fonte: Elaborado pelos autores.

O próximo passo é programar os componentes desta tela, lembrando que o determinante de uma matriz 2x2 é dado pela diferença do produto das diagonais principais pelo produto das diagonais secundárias, assim siga a figura 82 e monte os blocos.



Figura 82 - Blocos do aplicativo VI (g).  
Fonte: Elaborado pelos autores.

Nossa tela 5 está pronta, vamos agora para a última.

### • Tela 6 - Screen 6

Essa tela, cuja interface está representada na figura 83, será responsável pelo cálculo do determinante de matrizes 3x3, para tanto arraste para a área de visualização da aba designer os seguintes componentes:

- Uma legenda e mude o seu texto para “Informe a matriz”;
- Uma organização em tabelas com três linhas e três colunas;
- Dentro da organização em tabelas, nove caixas de texto alterando a dica para o respectivo coeficiente que ela representa em uma matriz;
- Um botão e mude o seu texto para “Calcular o determinante”;
- Uma legenda e exclua o seu texto.



Figura 83 - Interface da tela 3 do aplicativo VI.  
Fonte: Elaborado pelos autores.

Na aba blocos vamos programar os componentes, utilizando a regra de Sarrus para o cálculo do determinante, assim monte os blocos de acordo com a figura 84.



Figura 84 - Blocos do aplicativo VI (h).  
 Fonte: Elaborado pelos autores.

Nossa tela 6 está pronta.

#### • Concluindo o Aplicativo

Para concluir o aplicativo volte a tela 2 e insira dois botões, um com a funcionalidade de proporcionar um novo cálculo e o outro para retornar à tela inicial. Na tela 3 acrescente um botão para retornar à tela inicial. Nas telas 4, 5 e 6 acrescente dois botões, um para um novo cálculo do determinante e outro para retornar à tela do determinante (escolha do tipo de matriz quadrada). Acrescentados estes botões eles devem ser programados de acordo com a figura 85.

### Tela 2

```

quando Novaoperação .Clique
fazer
  ajustar a_11 . Texto - para obter valor inicial
  ajustar a_12 . Texto - para obter valor inicial
  ajustar a_13 . Texto - para obter valor inicial
  ajustar a_21 . Texto - para obter valor inicial
  ajustar a_22 . Texto - para obter valor inicial
  ajustar a_23 . Texto - para obter valor inicial
  ajustar a_31 . Texto - para obter valor inicial
  ajustar a_32 . Texto - para obter valor inicial
  ajustar a_33 . Texto - para obter valor inicial
  ajustar b_11 . Texto - para obter valor inicial
  ajustar b_12 . Texto - para obter valor inicial
  ajustar b_13 . Texto - para obter valor inicial
  ajustar b_21 . Texto - para obter valor inicial
  ajustar b_22 . Texto - para obter valor inicial
  ajustar b_23 . Texto - para obter valor inicial
  ajustar b_31 . Texto - para obter valor inicial
  ajustar b_32 . Texto - para obter valor inicial
  ajustar b_33 . Texto - para obter valor inicial

quando TelaInicial .Clique
fazer abrir outro ecrã nomeDoEcrã Screen1
  
```

### Tela 4

Informe a sua matriz

Calcular o Determinante

Calcular novo Determinante

Voltar à tela principal do Determinante

```

quando Botão1 . Clique
fazer
  ajustar a_11 Matriz 1 . Texto - para obter valor inicial
  ajustar Resultado . Texto - para obter valor inicial

quando Botão2 . Clique
fazer abrir outro ecrã nomeDoEcrã Screen3
  
```

Figura 85 - Complementando o aplicativo VI.  
 Fonte: Elaborado pelos autores.

Seguindo a figura 85 programe os novos botões nas telas 3, 5 e 6 e nosso aplicativo estará pronto, mas você ainda poderá inserir mais alguns componentes e alterar algumas propriedades, como exemplo observe a figura 86. Ah! Não esqueça de testar o aplicativo.

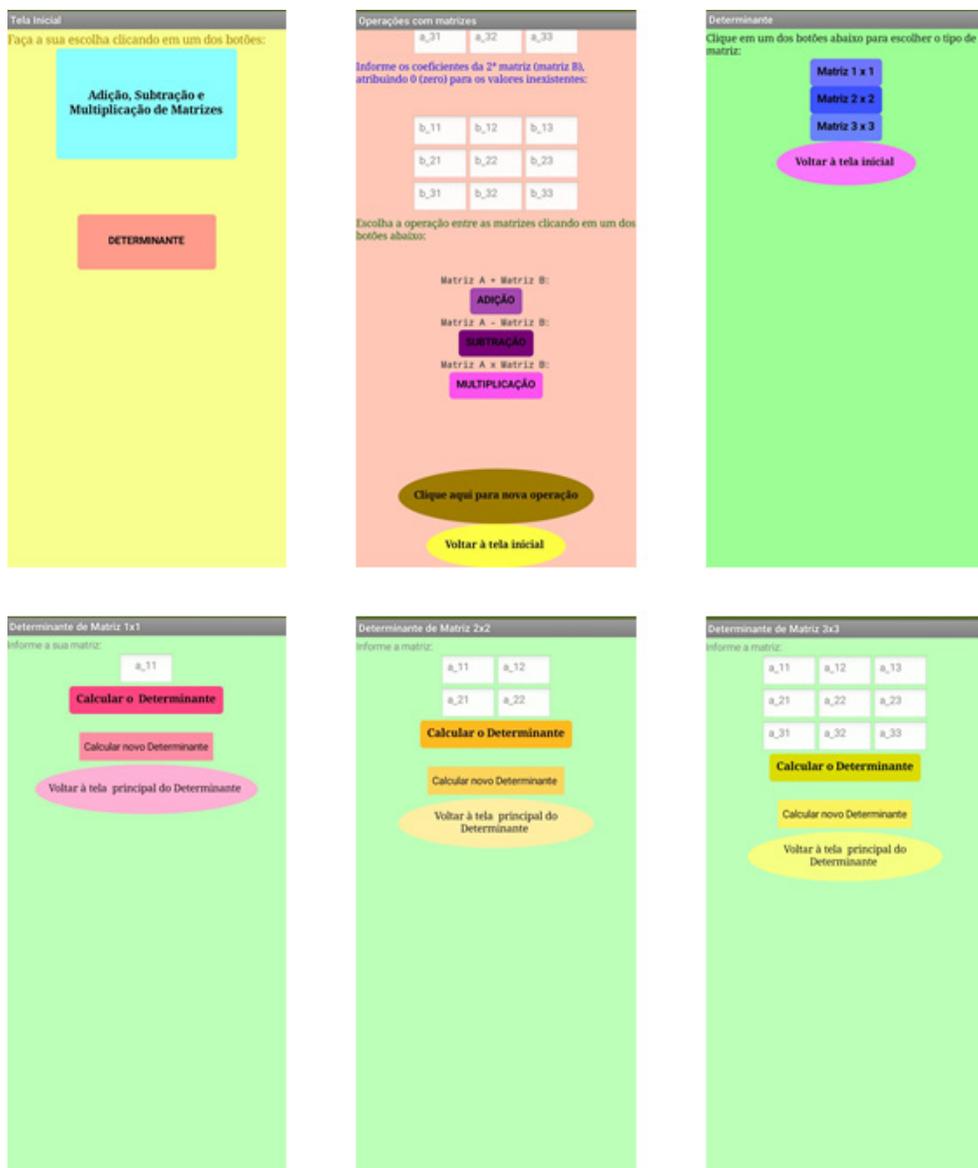


Figura 86 - Aplicativo VI concluído.  
Fonte: Elaborado pelos autores.

O aplicativo sobre matrizes está prontinho, então você pode prosseguir e desenvolver o próximo aplicativo.

## APLICATIVO VII: ANÁLISE COMBINATÓRIA

A seguir é explicitada a construção de um aplicativo sobre o conteúdo matemático Análise Combinatória, mais especificamente sobre os tipos de Combinatória, os arranjos, as permutações e as combinações. Vale salientar que antes de construir esse aplicativo é importante recordar os conceitos e as fórmulas de cada tipo de combinatória e depois acompanhe a construção descrita abaixo.

### • Na aba Designer

Primeiramente em sua conta do App Inventor inicie um novo projeto, em seguida insira duas organizações horizontais e dentro de cada arraste uma legenda e ao lado uma caixa de texto. Abaixo da segunda organização insira três botões e depois uma legenda, lembrando que cada botão indica um tipo de combinatória. Agora você pode nomear os componentes inseridos e alterar as propriedades conforme a figura 87.

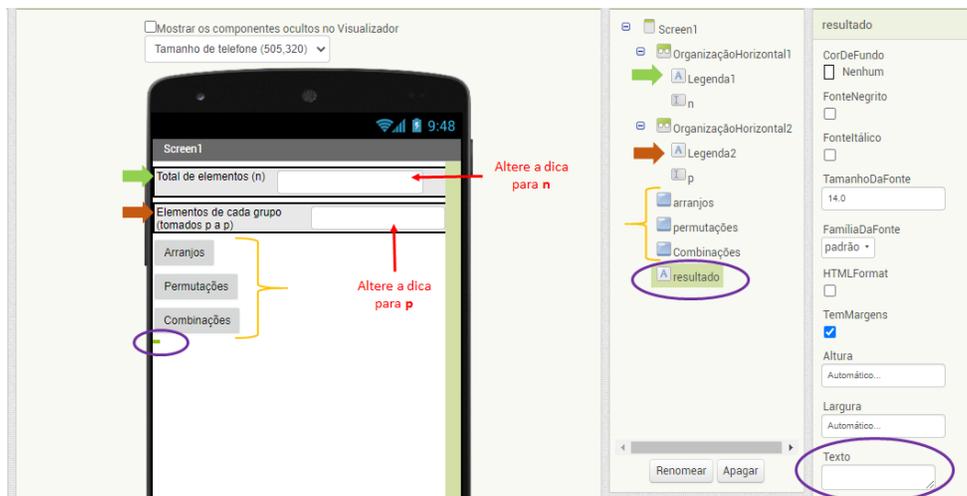


Figura 87 - Interface do aplicativo VII.

Fonte: Elaborado pelos autores.

Já temos os componentes que precisamos inseridos e organizados, o próximo passo é programá-los.

### • Na aba Blocos

Para facilitar o processo de programação, vamos utilizar duas opções oferecidas pelos blocos: a criação de listas e de procedimentos. De modo que para formar as expressões para encontrar arranjo, permutação e combinação precisamos somente dos seguintes elementos:  $n!$ ,  $(n-p)!$  e  $p!$ , então primeiramente programamos os blocos para determinarem estes elementos.

Para armazenar o  $n!$  inicie uma global fatorial com lista vazia e em seguida crie um procedimento para determinar o fatorial  $n$  ( $n!$ ), para isso você precisa dos blocos que estão na figura 88.

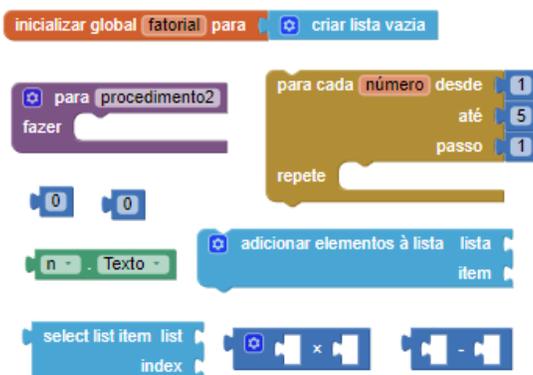


Figura 88 - Blocos do aplicativo VII (a) .  
 Fonte: Elaborado pelos autores.

O procedimento para encontrar  $n!$  inicia em 2 e vai até o número  $n$ , caminhando no passo 1, ou seja, após o número 2 vem o número 3, depois o 4 até  $n$ . De modo que a global fatorial será composta pela multiplicação do número  $n$  selecionado até o antecessor de 2, conforme os blocos da figura 89.



Figura 89 - Blocos do aplicativo VII (b).  
 Fonte: Elaborado pelos autores.

De acordo com a figura 90 crie um procedimento que informe o resultado de  $n!$ .



Figura 90 - Blocos do aplicativo VII (c).  
 Fonte: Elaborado pelos autores.

Para determinar o fatorial  $(n-p)!$  e  $p!$  siga os mesmos passos utilizados para determinar  $n!$ . Logo a programação para encontrar  $(n-p)!$  é dada pela figura 91.

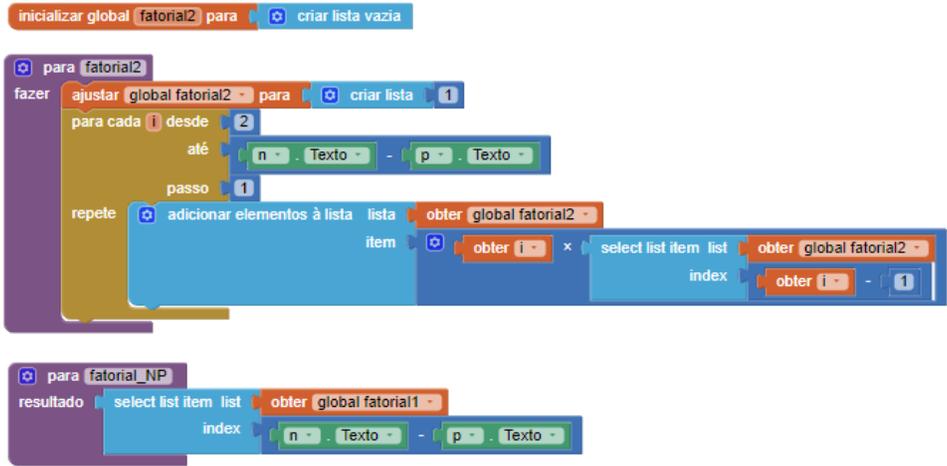


Figura 91 - Blocos do aplicativo VII (d).  
Fonte: Elaborado pelos autores.

E a programação para encontrar  $p!$  é dada na figura 92.

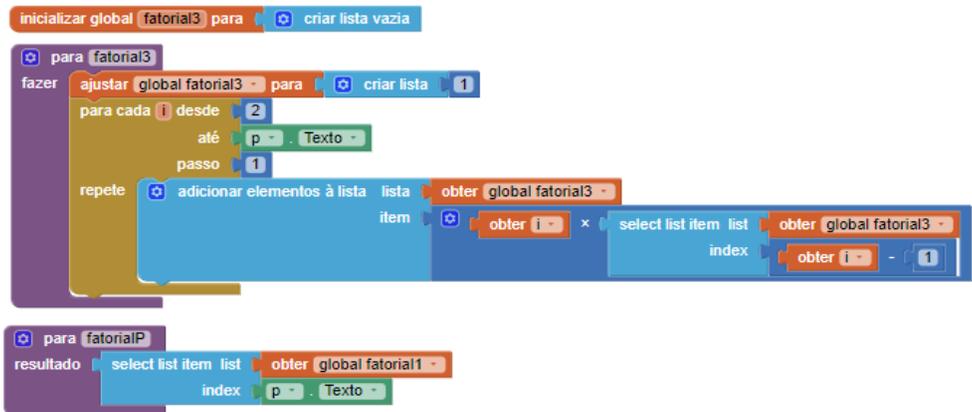


Figura 92 - Blocos do aplicativo VII (e).  
Fonte: Elaborado pelos autores.

O último passo é programar os botões do aplicativo, para isso devemos expressar as fórmulas do arranjo, da permutação e da combinação utilizando os procedimentos criados e demais blocos necessários. Lembre-se de programar os botões para que eles primeiramente ativem/chamem os procedimentos que serão utilizados por eles para determinar o resultado, conforme mostra a figura 93.

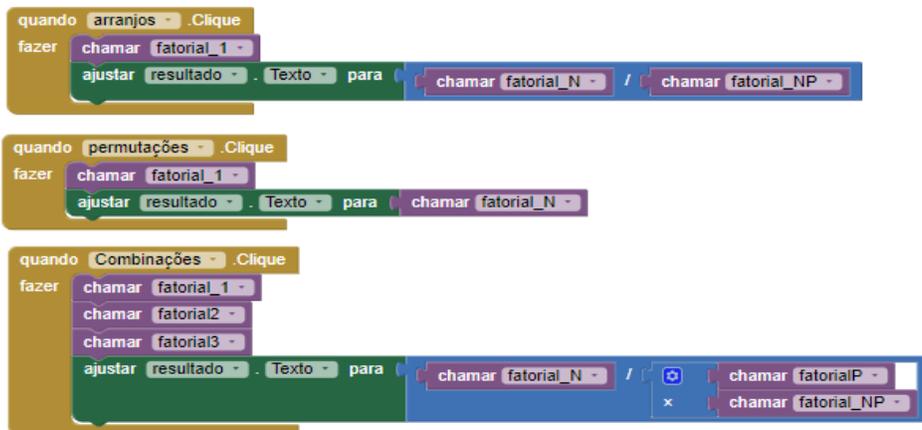


Figura 93 - Blocos do aplicativo VII (f).  
 Fonte: Elaborado pelos autores.

O aplicativo sobre Análise Combinatória está pronto, mas você ainda pode melhorá-lo, ajustando o designer e inserindo alguns componentes, observe a figura 94, sendo que o mais importante é realizar o teste do aplicativo.

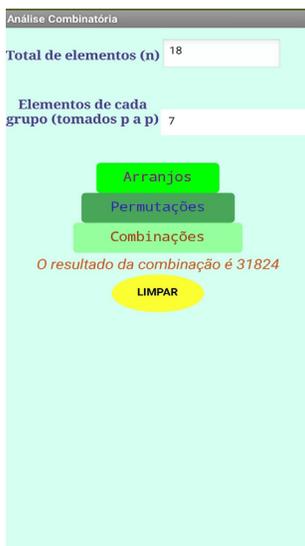


Figura 94 - Aplicativo VII concluído.  
 Fonte: Elaborado pelos autores.

## APLICATIVO VIII: QUIZ SOBRE FUNÇÕES DO 1º E DO 2º GRAU

O aplicativo construído aqui é um quiz e o tema escolhido para esse quiz foi funções do 1º e 2º grau, mas você pode escolher os mais variados temas, tanto conteúdos matemáticos como sobre qualquer outro assunto, como filmes e futebol, por exemplo.

O quiz apresentado é formado somente por três questões, onde cada questão contém uma imagem, assim acesse a sua conta no App Inventor, inicie um novo projeto e siga os passos descritos.

### • Na aba designer

Selecione a opção rolável do Screen1 e em seguida arraste para a área de visualização um componente imagem para apresentar as imagens, altere a dimensão para 80% de altura e 100% de largura, depois insira uma legenda, exclua o seu texto e a renomeie para “pergunta”. Em seguida insira uma organização horizontal e dentro dela insira uma caixa de texto e exclua a sua dica, ao lado insira dois botões, abaixo da organização insira uma legenda e exclua o seu texto e em seguida insira um botão. Observe a figura 95 e renomeie e mude o texto dos componentes.

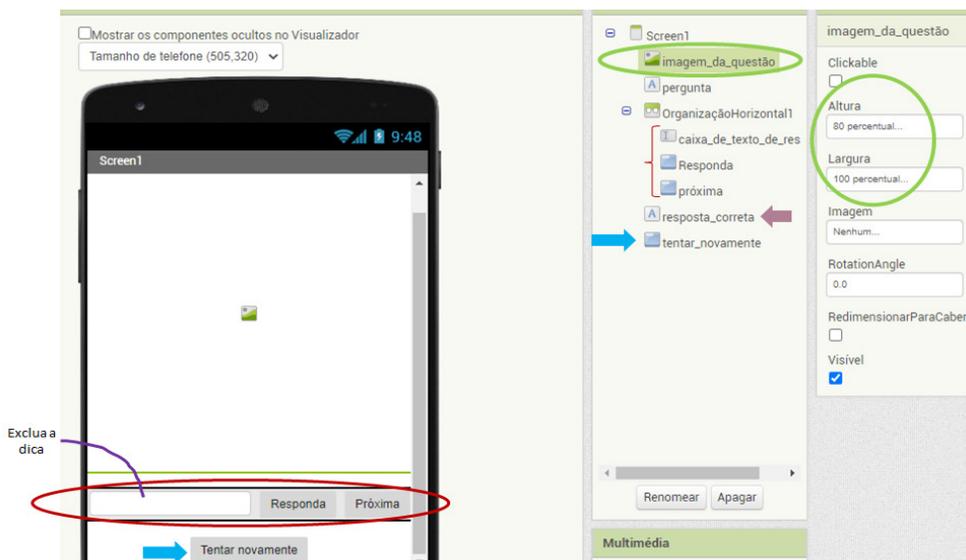


Figura 95 - Interface do aplicativo VIII.  
Fonte: Elaborado pelos autores.

Agora no campo “multimídia” insira as imagens que você irá utilizar no quiz em “enviar ficheiro”. Neste quiz utilizaremos as seguintes questões:

- “Qual o tipo de função representada no gráfico?”, cujo gráfico é dado pela figura

96 e a resposta é “segundo grau”.

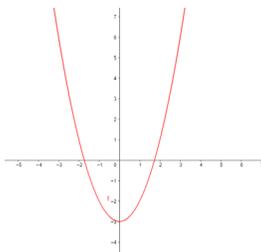


Figura 96 - Questão 1 do aplicativo VIII.  
Fonte: Elaborado pelos autores.

- O gráfico representa uma função do 1º grau. Ela é crescente ou decrescente?”, cuja imagem para essa questão é a figura 87 e a resposta para a pergunta é “decrescente”.

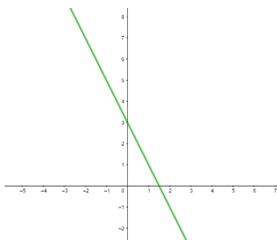


Figura 97 - Questão 2 do aplicativo VIII.  
Fonte: Elaborado pelos autores.

- “Os lados de um triângulo são segmentos de reta de qual tipo de função?”, sendo “primeiro grau” a resposta e a figura 98 a imagem utilizada.

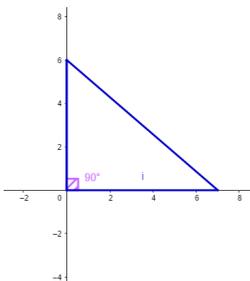


Figura 98 - Questão 3 do aplicativo VIII.  
Fonte: Elaborado pelos autores.

Após inserir as imagens na multimídia, vamos programar o aplicativo.

#### • Na aba Blocos

Primeiramente vamos criar uma variável global para cada elemento (imagens, perguntas e respostas) que compõem o quiz, tal que cada variável irá armazenar a lista do

elemento. Então, faça conforme na figura 99.



Figura 99 - Blocos do aplicativo VIII (a).  
Fonte: Elaborado pelos autores.

Crie uma variável global índice para possibilitar que ocorra a mudança de imagens, perguntas e respostas e depois ajuste a caixa de texto da pergunta para a pergunta 1, de acordo com a figura 100.



Figura 100 - Blocos do aplicativo VIII (b).  
Fonte: Elaborado pelos autores.

Agora programe os botões como a figura 101. Primeiramente o botão para passar para a próxima questão, onde da questão 1, passará para a 2, depois para a 3 e retornará a questão 1.



Figura 101 - Blocos do aplicativo VIII (c).  
Fonte: Elaborado pelos autores.

Agora o botão para responder a questão, de modo que onde ele compare a resposta dada ao aplicativo com a resposta informada pelo usuário, onde para não ocorrerem erros em relação a variância de letras maiúsculas e minúsculas ele analisa os dois textos exclusivamente em letras maiúsculas e posteriormente informa ao usuário se a resposta está correta ou não. Observe a figura 102.

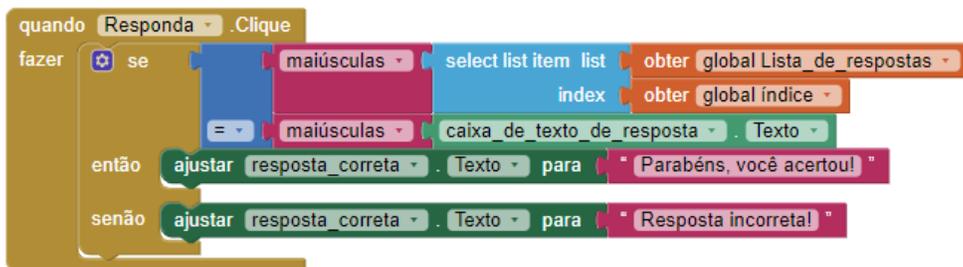


Figura 102 - Blocos do aplicativo VIII (d).  
Fonte: Elaborado pelos autores.

Agora só programar o botão tentar novamente, sendo que esse botão tem a funcionalidade de limpar os campos para o usuário responder a mesma pergunta novamente, caso não tenha acertado na tentativa anterior. Faça como na figura 103.

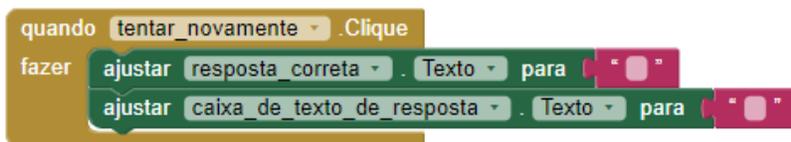


Figura 103 - Blocos do aplicativo VIII (e).  
Fonte: Elaborado pelos autores.

Perfeito! Nosso quiz sobre funções do 1° e 2° grau está pronto, agora teste-o e deixe-o a sua cara, um exemplo está ilustrado na figura 104 e lembre-se que você pode acrescentar mais questões e utilizar este modelo para elaborar quiz sobre os assuntos do seu interesse.

Screen1

Qual o tipo de função representada no gráfico?

segundo grau

Resposta Próxima

Parabéns, você acertou!

Tentar novamente

Screen1

O gráfico representa uma função do 1º grau. Ela é crescente ou decrescente?

crescente

Resposta Próxima

Resposta incorreta!

Tentar novamente

Screen1

Os lados de um triângulo são segmentos de reta de qual tipo de função?

Informe a resposta aqui e em seguida clique no botão "responda"

Resposta Próxima

Tentar novamente

Figura 104 - Aplicativo VIII concluído.  
 Fonte: Elaborado pelos autores.

Parabéns, você concluiu mais um aplicativo matemático!

## CONSIDERAÇÕES FINAIS

Esta obra trouxe uma explanação sobre a composição e o funcionamento do software App Inventor, entretanto nem todos os componentes e funcionalidades, como nem todos os blocos puderam ser explorados detalhadamente, por isso recomenda-se que o usuário explore o software, pois quanto mais trabalhar com ele mais domínio e facilidade terá para desenvolver aplicativos.

Os aplicativos apresentados são sugestões, especialmente para professores de Matemática da Educação Básica e esperamos que a partir do desenvolvimento destes possamos contribuir para que o professor consiga elaborar outros, se apropriando cada vez mais do App Inventor.

## REFERÊNCIAS

APP INVENTOR. (2020). Disponível em <http://ai2.appinventor.mit.edu/> . Acesso em 05 dez. 2020.

REBOUÇAS, Ayla Débora Dantas de Souza. **Minicurso: Introdução a Programação e Criação de Aplicativos com o MIT App Inventor 2**. DCX-CCAE/UFPB, 2019.

**JANICE TERESINHA REICHERT** - Professora da Universidade Federal da Fronteira Sul (UFFS). Possui Graduação em Matemática pela Universidade Federal de Santa Maria (UFSM), Mestrado em Matemática e Computação Científica pela Universidade Federal de Santa Catarina (UFSC), Doutorado em Engenharia Mecânica pela Universidade Federal do Rio Grande do Sul (UFRGS) e realizou estágio pós-doutoral no Departamento de Informática Aplicada do Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS).

**BRUNA MIECOANSKI** - Licenciada em Matemática pela Universidade Federal da Fronteira Sul – UFFS. Bolsista de iniciação científica durante o período de 2020 a 2022.

**MILTON KIST** - Professor da Universidade Federal da Fronteira Sul (UFFS). Possui Graduação em Matemática pela Universidade Federal de Santa Catarina (UFSC), Mestrado em Matemática e Computação Científica pela Universidade Federal de Santa Catarina (UFSC), Doutorado em Engenharia Mecânica e de Materiais pela Universidade Tecnológica Federal do Paraná (UTFPR) – *Campus* Curitiba.

[www.atenaeditora.com.br](http://www.atenaeditora.com.br) 

[contato@atenaeditora.com.br](mailto:contato@atenaeditora.com.br) 

@atenaeditora 

[www.facebook.com/atenaeditora.com.br](https://www.facebook.com/atenaeditora.com.br) 

# Desenvolvimento de aplicativos matemáticos com o **APP INVENTOR**

[www.atenaeditora.com.br](http://www.atenaeditora.com.br) 

[contato@atenaeditora.com.br](mailto:contato@atenaeditora.com.br) 

[@atenaeditora](https://www.instagram.com/atenaeditora) 

[www.facebook.com/atenaeditora.com.br](https://www.facebook.com/atenaeditora.com.br) 

# Desenvolvimento de aplicativos matemáticos com o **APP INVENTOR**