



UNIVERSIDADE ESTADUAL DE SANTA CRUZ
DEPARTAMENTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
MESTRADO EM MATEMÁTICA EM REDE NACIONAL – PROFMAT

MARCOS ALBERTO BARBOSA

DESENVOLVENDO APLICATIVOS PARA DISPOSITIVOS MÓVEIS
ATRAVÉS DO MIT APP INVENTOR 2 NAS AULAS DE MATEMÁTICA

ILHÉUS – BA

2016

MARCOS ALBERTO BARBOSA

**DESENVOLVENDO APLICATIVOS PARA DISPOSITIVOS MÓVEIS
ATRAVÉS DO MIT APP INVENTOR 2 NAS AULAS DE MATEMÁTICA**

Dissertação de mestrado apresentada ao programa Mestrado Profissional em Matemática em Rede Nacional – PROFMAT ofertado pela Universidade Estadual de Santa Cruz e coordenado pela Sociedade Brasileira de Matemática, como requisito final à obtenção do título de mestre em Matemática.

Orientadora: Profa. Dra. Mirela Vanina de Mello.

Coorientador: Prof. Me. André Malvezzi Lopes.

ILHÉUS – BA

2016

MARCOS ALBERTO BARBOSA

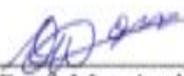
**DESENVOLVENDO APLICATIVOS PARA DISPOSITIVOS MÓVEIS
ATRAVÉS DO MIT APP INVENTOR 2 NAS AULAS DE MATEMÁTICA**

Dissertação de mestrado apresentada ao programa Mestrado Profissional em Matemática em Rede Nacional – PROFMAT ofertado pela Universidade Estadual de Santa Cruz e coordenado pela Sociedade Brasileira de Matemática, como requisito final à obtenção do título de mestre em Matemática.

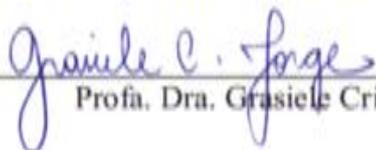
Trabalho aprovado em 14 de Janeiro de 2016.



Profª. Dra. Mirela Vanina de Mello
UESC - Orientadora



Prof. Me. André Malvezzi Lopes
UESC - Coorientador



Profª. Dra. Grasielle Cristiane Jorge

ILHÉUS – BA

2016

B238 Barbosa, Marcos Alberto.
Desenvolvendo aplicativos para dispositivos móveis através do MIT App inventor 2 nas aulas de matemática / Marcos Alberto Barbosa. – Ilhéus, BA: UESC, 2016.
141 f. : il. ; anexos.

Orientadora: Mirela Vanina de Mello.
Coorientador: André Malvezzi Lopes.
Dissertação (Mestrado) – Universidade Estadual de Santa Cruz. Programa de Mestrado Profissional em Matemática em Rede Nacional – PROFMAT.
Referências bibliográficas: f. 116-119.

1. Matemática. 2. Sistemas lineares. 3. Matrizes (Matemática). 4. Android (Recurso eletrônico). I. Título.

CDD 510

Ao meu avô (*in memoriam*) e Diego (*in memoriam*).

Agradeço

ao Célio, Elias e Léo pelas divertidas viagens,

aos meus amigos e colegas de trabalho pelo apoio,

aos meus professores, Mirela e André pela orientação,

em especial a minha esposa Magaly, por ter sido mãe e pai.

RESUMO

Ano após ano, a educação pública vem registrando baixos índices de proficiência em Matemática, contrariando as próprias metas estabelecidas para serem alcançadas em 2022. O número de vendas de smartphones e tablets por sua vez está em franco crescimento, atingindo consumidores de todas as idades e classes sociais. Diante desta realidade, é apresentado este trabalho composto por três sequências didáticas, cujo objetivo é facilitar o processo de ensino-aprendizagem dos conteúdos de equações do segundo grau, matrizes e sistemas lineares através da construção de aplicativos, usando para isso os aparelhos móveis dos alunos e a plataforma MIT App Inventor 2 criada para desenvolver programas para o sistema operacional Android.

Palavras chaves: Android, Aplicativo, App Inventor, Bhaskara, Equações do 2º Grau Matrizes, Sistemas Lineares.

ABSTRACT

Year after year, public education has been recording low levels of proficiency in Mathematics contradicting its own goals set to be achieved in 2022. The number of sales of smartphones and tablets in turn is rapidly growing and reaching consumers of all ages and social classes. Face of this reality, it is presented this work, composed by three didactic sequences, which aims to facilitate the teaching-learning process of second degree equations, matrices and linear systems by building applications using mobile devices of the student and the platform MIT App Inventor 2, created to develop programs for the Android operating system.

Key words: Android, App, App Inventor, Bhaskara, Linear Systems, Matrices, Second Degree Equations.

LISTA DE SIGLAS E ABREVIACOES

BBN – BBN Technologies (Bolt, Beranek and Newman)

Inep – Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira

IBGE – Instituto Brasileiro de Geografia e Estatística Kcal – Quilocalorias

MEC – Ministério da Educação

MIT – Massachusetts Institute of Technology (Instituto Tecnológico de Massachusetts)

MAI – MIT App Inventor 2

PNAD – Pesquisa Nacional por Amostra de Domicílios

TPE – Todos Pela Educação

Saeb – Sistema Nacional de Avaliação da Educação Básica

ÍNDICE DE ILUSTRAÇÕES

Figura 1 – Médias nacionais das proficiências de Português e Matemática entre	19
Figura 2 – Médias regionais e baiana das proficiências de Matemática entre os anos.....	19
Figura 3 – Médias baianas das proficiências de Matemática entre os anos de 1995 a 2013 dos alunos do 4º/5º e 8º/9º ano do Ensino Fundamental e do 3º Ano do Ensino Médio. Fonte: TPE.	20
Figura 4 – A tartaruga original do "Mindstorms", Seymour Papert, Basic Books, Inc., 1980 (esquerda), e o TI 99/4 ao lado do Apple II. Ambos podiam ser ligados em televisores (direita).	25
Figura 5 – Exemplo de um bloco de programação referente à tarefa de potenciação.	26
Figura 6 – Anúncios de produtos tecnológicos em diferentes décadas.	28
Figura 7 – Ciclo das sequências lógicas de tarefas no ato da programação.	33
Figura 8 – Exemplos de funções quadráticas representadas por parábolas.	35
Figura 9 – Representação gráfica de funções quadráticas com $a > 0$ e $a < 0$, respectivamente.	36
Figura 10 – Representações gráficas de funções quadráticas de acordo com o sinal de a e de Δ	37
Figura 11 – Pontos de mínimos e máximos de gráficos de funções quadráticas.....	38
Figura 12 – A parábola, o eixo de simetria e o vértice.	39
Figura 13 – Gráfico da função $f(x) = x^2 - 3x - 10$	40
Figura 14 – Gráfico da função $g(x) = 3x^2 + 2x + 4$	40
Figura 15 – Discriminação das funções de cada componente.	42
Figura 16 – Etiquetas auxiliam o uso do aplicativo pelo usuário.	43
Figura 17 – Escrita linear da fórmula de Bhaskara.	43
Figura 18 – Relação entre as caixas de textos e os coeficientes da função quadrática.	43
Figura 19 – Sequência de passos a ser realizado pelo aplicativo para solucionar a função.	44
Figura 20 – Componentes adicionados na tela do aplicativo.	45
Figura 21 – Componentes adicionados na tela do aplicativo após as mudanças nas suas propriedades.	46
Figura 22 – Itens necessários para o cálculo de $(4 \cdot a \cdot c)$	47
Figura 23 – Procedimento para adicionar termos no bloco da multiplicação.	47
Figura 24 – Bloco responsável pelo cálculo de $(b^2 - 4 \cdot a \cdot c)$	47
Figura 25 – Comparação das formas de representar potência fracionária.....	48
Figura 26 – Bloco responsável pelo cálculo de $(b^2 - 4 \cdot a \cdot c)^{(1/2)}$	48
Figura 27 – Blocos necessários pelos cálculos de cada uma das raízes.	48

Figura 28 – Composição dos numeradores de cada raiz.	49
Figura 29 – Representação completa da fórmula de Bhaskara em blocos.	49
Figura 30 – Conjunto responsável pela análise do sinal de Δ	49
Figura 31 – Conjunto responsável pela exibição da mensagem nas etiquetas das respostas. ...	50
Figura 32 – Composição final da parte lógica do aplicativo.	51
Figura 33 – Capturas de telas do smartphone exibindo o aplicativo solucionando os exemplos propostos.	52
Figura 34 – Ilustração das últimas modificações realizadas nos componentes.	53
Figura 35 – Captura de tela do smartphone exibindo as modificações feitas no aplicativo.	54
Figura 36 – Conjunto responsável por limpar todas as caixas de textos e etiquetas.	54
Figura 37 – Exemplo de matriz aplicada no cotidiano. Fonte: Google.	56
Figura 38 – Visualização da aplicação da Regra de Sarrus.	59
Figura 39 – Modelo de arranjo das múltiplas telas do aplicativo.	63
Figura 40 – (a) Esboço da tela inicial do programa. (b) Esboço de tela para operação de soma entre matrizes.	63
Figura 41 – Tela inicial do aplicativo “Calculadora de Matrizes”.	65
Figura 42 – Demonstração do procedimento para adicionar novas telas.	65
Figura 43 – Demonstração do procedimento para alternar entre as telas.	65
Figura 44 – Ilustração das telas “tela_soma” e “tela_produto” após a inclusão dos componentes.	66
Figura 45 – Visão dos componentes inseridos na “tela_soma2”.	67
Figura 46 – Captura de tela do smartphone durante a montagem de “tela_soma2”.	68
Figura 47 – Agrupamento responsável pela soma dos elementos das matrizes A e B com posterior exibição na matriz C.	69
Figura 48 – Bloco responsável por abrir/retornar a “tela_soma”.	69
Figura 49 – Capturas de telas do smartphone exibindo o aplicativo solucionando os exemplos propostos.	70
Figura 50 – Visão dos componentes inseridos na tela “tela_soma3”.	71
Figura 51 – Agrupamento responsável pela soma dos elementos das matrizes A e B com posterior exibição na matriz C e a função do “botao_voltar”.	71
Figura 52 – Captura de tela do smartphone exibindo o aplicativo funcionando.	72
Figura 53 – Visão dos componentes inseridos na “tela_produto2”.	73
Figura 54 – Bloco que calcula e exhibe o valor de "c11".	73
Figura 55 – Agrupamento responsável pelo cálculo de todos os elementos da matriz C.	73
Figura 56 – Agrupamento responsável pelo cálculo da matriz C.	74
Figura 57 – Conjunto responsável por abrir a "tela_produto" através do "botao_voltar".	74
Figura 58 – Capturas de tela do smartphone exibindo o aplicativo solucionando os exemplos dados.	74
Figura 59 – Visão dos componentes inseridos na “tela_produto3”.	75
Figura 60 – Bloco que calcula o valor de "c11".	75
Figura 61 – Procedimento para aumentar fatores no bloco de adição.	76
Figura 62 – Demonstração da função <i>External Inputs</i> que facilita o manuseio de blocos grandes.	76
Figura 63 – Bloco que calcula o valor do elemento "c11".	76
Figura 64 – Conjunto final dos blocos de todos termos da matriz C e a função do “botao_voltar”.	77

Figura 65 – Captura de tela com o aplicativo em funcionamento.....	77
Figura 66 – Captura de tela exibindo os componentes inseridos nos passos anteriores.....	78
Figura 67 – Captura de tela do celular exibindo as modificações realizadas.....	80
Figura 68 – Parte do conjunto que calcula determinantes de matrizes 2×2	80
Figura 69 – Conjunto que calcula determinantes de matrizes 2×2	80
Figura 70 – Capturas de telas do smartphone exibindo o aplicativo solucionando os exemplos propostos.....	81
Figura 71 – Conjunto de blocos dos produtos de cada diagonal para determinantes de matrizes 3×3	81
Figura 72 – Conjunto de soma dos produtos de cada diagonal para determinantes de matrizes 3×3	82
Figura 73 – Bloco que calcula determinantes de matrizes 3×3	82
Figura 74 – Bloco responsável por abrir a tela inicial “Screen1”.....	82
Figura 75 – Captura de telas do smartphone exibindo o aplicativo solucionando os exemplos propostos.....	83
Figura 76 – Blocos responsáveis por abrir as telas “tela_soma”, “tela_produto” e “tela_determinantes”.....	83
Figura 77 – Captura de tela do smartphone exibindo a “tela_soma” e os blocos responsáveis por abrirem as telas.....	84
Figura 78 – Captura de tela do smartphone exibindo a “tela_produto” e os blocos responsáveis por abrirem as telas desejadas.	84
Figura 79 – Captura de algumas telas do aplicativo exibindo as modificações aplicadas.	86
Figura 80 – Ilustração da inserção do “botao_limpar” nas telas “tela_soma2” e “tela_determinantes”.....	86
Figura 81 – Exemplos de blocos responsáveis por limpar caixas de textos e etiquetas.....	87
Figura 82 – Demonstração prática da aplicação de sistemas lineares no cotidiano.	88
Figura 83 – Classificação dos sistemas de equações lineares de acordo com suas soluções. ..	90
Figura 84 – Esquema das múltiplas telas do aplicativo.....	92
Figura 85 – Esboço da tela inicial do aplicativo.....	93
Figura 86 – Esboço das telas que conterão as calculadoras.	93
Figura 87 – Tela inicial do aplicativo com os componentes inseridos.....	94
Figura 88 – Demonstração do procedimento para adicionar novas telas.	95
Figura 89 – Demonstração do procedimento para alternar entre as telas.....	95
Figura 90 – Componentes inseridos na “tela_2”.....	97
Figura 91 – Captura da “tela_2” com os novos componentes inseridos.	97
Figura 92 – Composição do cálculo de $\det(S)$	99
Figura 93 – Composição do cálculo de $\det(S_1)$ e $\det(S_2)$	99
Figura 94 – Demonstração dos determinantes $\det(S)$, $\det(S_1)$ e $\det(S_2)$ transformados em procedimentos.....	99
Figura 95 – Bloco que guarda a informação a ser exibida quando o $\det(S)$ é igual a zero. ...	100
Figura 96 – Procedimento para adicionar itens <i>else</i>	100
Figura 97 – Agrupamento do bloco <i>if/then</i> com o bloco <i>click</i> do “botao_calcular”.	101
Figura 98 – Parte inicial da composição que soluciona os sistemas lineares.....	101
Figura 99 – Composição final dos cálculos para solução dos sistemas lineares de duas incógnitas.....	102
Figura 100 – Bloco responsável por abrir a tela inicial “Screen1”.....	102

Figura 101 – Captura de tela do celular exibindo as soluções encontradas pelo aplicativo...	103
Figura 102 – Componentes inseridos na “tela_3”	105
Figura 103 – Captura da “tela_3” com os novos componentes inseridos.	106
Figura 104 – Demonstração dos determinantes $\det(S)$, $\det(S_1)$, $\det(S_2)$ e $\det(S_3)$ transformados em procedimentos.	107
Figura 105 – Demonstração da função <i>External Inputs</i> que facilita o manuseio de blocos grandes.....	108
Figura 106 – Composição final dos cálculos para solução dos sistemas lineares de três incógnitas e a função do “botao_voltar”.....	108
Figura 107 – Captura de tela do celular exibindo as soluções encontradas pelo aplicativo...	108
Figura 108 – Ilustração da inserção do “botao_limpar”.....	109
Figura 109 – Exemplos de blocos responsáveis por limpar as caixas de textos e etiquetas...	110
Figura 110 – Composições do “botao_2” e “botao_3”.....	110
Figura 111 – Captura das telas do aplicativo exibindo as modificações aplicadas.....	112
Figura 112 – Exemplos de aplicativos desenvolvidos paralelamente ao projeto.....	113
Figura 113 – Cadeia de tarefas para serem realizadas em grupo durante o trabalho com o MAI.....	114
Figura 114 – Exemplos de aplicativos que podem ser trabalhados na escola.....	115
Figura 115 – Relação das três opções para testar os projetos em desenvolvimento.....	121
Figura 116 – Página que disponibiliza os links para a instalação do aplicativo do MAI que executa o projeto em desenvolvimento no smartphone/tablet.....	122
Figura 117 – Demonstração da escolha <i>All Companion</i> para testar o projeto através do wi-fi.....	122
Figura 118 – Capturas de telas do smartphone demonstrando o programa instalado e em execução.....	123
Figura 119 – Código gerado pela plataforma para que o dispositivo baixe e execute o aplicativo em desenvolvimento.....	123
Figura 120 – Página que disponibiliza os links para a instalação do emulador no computador.....	124
Figura 121 – Demonstração da escolha <i>Emulator</i> para testar o projeto.....	124
Figura 122 – Ilustração da plataforma iniciando a execução do emulador no computador...	125
Figura 123 – Ilustração da plataforma executando o emulador no computador.....	125
Figura 124 – Links de downloads dos programas para serem instalados no computador e no dispositivo móvel.....	126
Figura 125 – Demonstração da escolha <i>USB</i> para testar o projeto.....	126
Figura 126 - Ilustração da ativação da função <i>DESENVOLVEDOR</i> em diferentes versões do Android.....	127
Figura 127 – Destaque para a opção que exporta a instalação do aplicativo.....	127
Figura 128 – Página inicial do MIT App Inventor.....	128
Figura 129 – Página de acesso à conta do Google.....	128
Figura 130 – Página inicial da plataforma exibindo o menu de opções e os projetos desenvolvidos.....	129
Figura 131 – Página da <i>Gallery</i> (Galeria) do MIT App Inventor 2 disponibilizando aplicativos desenvolvidos por outros usuários.....	129
Figura 132 – Demonstração do procedimento de criar um novo projeto.....	129
Figura 133 – Caixa de diálogo para entrada do nome escolhido do novo projeto.....	130

Figura 134 – Ambiente de trabalho do MAI.	130
Figura 135 – Procedimento de arrastar e soltar componentes na tela <i>Viewer</i>	130
Figura 136 – Imagem aponta os locais onde foram realizadas as modificações no componente <i>Button</i>	131
Figura 137 – Ambiente do MAI para manuseio dos blocos responsáveis pelas funções dos componentes inseridos na tela <i>Designer</i>	131
Figura 138 – Os blocos laranjas são da função <i>Control</i> (Controle), os azuis.....	132
Figura 139 – Exemplo de uma montagem de blocos associando uma regra ao botão.	132
Figura 140 – Página inicial da plataforma do MAI.	133
Figura 141 – Ambiente de trabalho do MAI.	133
Figura 142 – Componentes inseridos na tela inicial do aplicativo.	134
Figura 143 – Lista de componentes inseridos e editados.	134
Figura 144 – Área de trabalho para edição dos blocos.....	135
Figura 145 – Bloco de soma.	135
Figura 146 – Bloco relacionado ao atributo texto do componente “valor01”.	135
Figura 147 – Composição de soma dos componentes “valor01” e “valor02”.....	136
Figura 148 – Bloco relacionado à seleção do componente “resultado”.	136
Figura 149 – Composição de soma dos componentes “valor01” e “valor02” para ser exibido em “resultado”.	136
Figura 150 – Função <i>clique</i> do componente botão “calcular”.....	136
Figura 151 – Composição final do procedimento de soma entre os componentes “valor01” e “valor02”.	137
Figura 152 – Capturas de tela do smartphone exibindo o aplicativo funcionando.....	137
Figura 153 – Ilustração das modificações aplicadas nos componentes.....	138
Figura 154 – Captura de tela do smartphone exibindo o aplicativo funcionando além das edições realizadas.	139
Figura 155 – Modelo diferente de layout do aplicativo.....	139
Figura 156 – Procedimento para exportar o projeto construído.	140
Figura 157 – Código de barra com o atalho para download do projeto direto para o dispositivo móvel.	140
Figura 158 – Janela de diálogo indicando o início do download do aplicativo para o computador.	140
Figura 159 – Processo de instalação do aplicativo no smartphone.	141

SUMÁRIO

1 APRESENTAÇÃO.....	16
2 INTRODUÇÃO.....	18
2.1 Alguns dados nacionais sobre Matemática.....	18
2.2 Os assuntos escolhidos.....	22
2.3 A linguagem Logo	23
2.4 Do App Inventor para o MIT App Inventor 2	26
2.5 O MIT App Inventor na sociedade	27
2.6 Por que o MIT App Inventor?.....	28
2.7 A tecnologia disponível	29
2.8 Por que programar?.....	31
2.9 Antes de começar.....	33
3 EQUAÇÕES DO 2º GRAU	35
3.1 Principais resultados.....	36
3.2 Exemplos.....	39
3.3 Sequência didática para construção do aplicativo.....	41
3.3.1 Esboço da estrutura do aplicativo e escolha dos componentes da tela.....	42
3.3.2 Início da construção do aplicativo com a montagem da tela.....	44
3.3.3 Montagem dos blocos responsáveis pelos cálculos usando o método de Bhaskara.....	46
3.3.4 Montagem dos blocos responsáveis pela análise do sinal do discriminante Δ	49
3.3.5 Conclusão da montagem dos blocos.....	50
3.3.6 Finalização do projeto com modificações nas propriedades dos componentes.....	52
3.3.7 Desafio.....	55
4 MATRIZES	56
4.1 Principais resultados.....	57
4.2 Exemplos.....	60
4.3 Sequência didática para construção do aplicativo.....	61
4.3.1 Esboço da estrutura do aplicativo e escolha dos componentes de algumas telas.....	62

4.3.2	Início da construção do aplicativo com a inclusão das primeiras telas.....	64
4.3.3	Construção da calculadora de soma entre matrizes de ordem 2.....	66
4.3.4	Construção da calculadora de soma entre matrizes de ordem 3.....	70
4.3.5	Construção da calculadora de produto de matrizes de ordem 2.....	72
4.3.6	Construção da calculadora de produto de matrizes de ordem 3.....	75
4.3.7	Construção da tela da calculadora de determinantes de matrizes.	78
4.3.8	Montagem dos blocos da calculadora de determinantes de matrizes 2×2	80
4.3.9	Montagem dos blocos da calculadora de determinantes de matrizes 3×3	81
4.3.10	Finalização do projeto com modificações nas propriedades dos componentes.	83
4.3.11	Desafio.	87
5	SISTEMAS DE EQUAÇÕES LINEARES.....	88
5.1	Principais resultados.	88
5.2	Exemplos.	90
5.3	Sequência didática para construção do aplicativo.	91
5.3.1	Esboço da estrutura do aplicativo e escolha dos componentes de algumas telas.....	92
5.3.2	Início da construção do aplicativo com a inclusão das telas.....	93
5.3.3	Construção da tela da calculadora de que soluciona sistemas lineares de duas incógnitas.	95
5.3.4	Esboço do cálculo para determinar as incógnitas e montagem dos blocos para sistemas lineares de duas incógnitas.	98
5.3.5	Esboço do cálculo para determinar as incógnitas para sistemas lineares de três incógnitas.	103
5.3.6	Construção da tela da calculadora de que soluciona sistemas lineares de três incógnitas.	104
5.3.7	Montagem dos blocos da calculadora de que soluciona sistemas lineares de três incógnitas.	107
5.3.8	Finalização do projeto com modificações das propriedades dos componentes.	109
5.3.9	Desafio.	112
6	CONSIDERAÇÕES FINAIS.....	113
7	REFERÊNCIAS BIBLIOGRÁFICAS.....	116
8	ANEXOS.....	120
	ANEXO A – O QUE INSTALAR.....	120
	ANEXO B – CONHECENDO A PLATAFORMA.	128
	ANEXO C – CONSTRUINDO O PRIMEIRO APLICATIVO.	133

1 APRESENTAÇÃO

Com o mercado de tecnologia caminhando para o acesso pleno aos produtos individualizados de conexão, surgiram duas novas ferramentas que podem auxiliar o processo de ensino-aprendizagem: o smartphone e o tablet. Estes aparelhos dão o devido acesso ao mundo virtual e são capazes de armazenar diferentes tipos de multimídias¹, agregado com as múltiplas funções incluídas em cada aparelho.

Atividades vinculadas à informática, que antes eram envolvidas de complexidade e desinteresse por parte dos docentes, podem ser feitas a qualquer hora e lugar com estes dispositivos, que permitem acesso à Internet, registros de imagens e vídeos, acompanhados de aplicativos de pesquisa, utilizando centrais de processamento muitas vezes superiores aos dos computadores instalados nos laboratórios escolares.

Porém, o que nota-se é uma relutância de professores e administradores em aceitar a entrada pacífica deste aparelho no ambiente escolar, usando como principal argumento que o uso inapropriado de smartphones e tablets levam os estudantes à distração e conseqüentemente ao baixo rendimento na aprendizagem.

O cerne deste trabalho é oferecer três exemplos significativos do uso do smartphone/tablet no processo de educação, através de seqüências didáticas que ensinam a construir aplicativos capazes de determinar as raízes de equações do segundo grau, operar matrizes e solucionar sistemas lineares.

¹ Multimídia é a combinação, controlada por computador (computador pessoal, periférico e dispositivo móvel), de pelo menos um tipo de mídia estática (texto, fotografia, gráfico), com pelo menos um tipo de mídia dinâmica (vídeo, áudio, animação).

O texto é dividido em três partes, iniciando com a apresentação de alguns dados publicados recentemente sobre a proficiência em matemática na rede pública de ensino, extraídos de relatórios de avaliações nacionais como o Sistema de Avaliação da Educação Básica (Saeb) e a Avaliação Nacional do Rendimento Escolar (também conhecida como Prova Brasil), ambos coordenados pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep).

A seguir vem a apresentação do Logo, considerado o primeiro projeto computacional voltado para a educação, co-desenvolvido por Seymour Papert na década de 1960, que não por acaso é o criador do construcionismo, teoria intimamente ligada com o construtivismo, já que Papert trabalhou com Piaget enquanto estavam na Europa. Neste mesmo contexto, segue um breve relato sobre o MIT App Inventor 2 (MAI), uma plataforma virtual de desenvolvimento de aplicativos para o sistema operacional Android², mantida pelo Instituto de Tecnologia de Massachusetts (MIT) e coordenado também por Papert, cujo objetivo é disponibilizar um ambiente para os mais diversos fins educacionais de iniciação, modelagem e prática de computação.

No terceiro momento vêm as sequências didáticas, inspiradas na idéia da Logo, que culminam na construção de aplicativos, acompanhados de vasto anexo com imagens e tutoriais para auxiliar o professor na aplicação da atividade.

² O Android é o sistema operacional móvel mais utilizado do mundo. Foi desenvolvido pela empresa de tecnologia Google baseado no núcleo Linux. Possui uma interface de usuário baseada na manipulação direta, pois é projetado principalmente para dispositivos móveis com tela sensível ao toque como smartphones e tablets; também é utilizado em consoles de videogames, câmeras digitais, computadores e outros dispositivos eletrônicos.

2 INTRODUÇÃO

2.1 Alguns dados nacionais sobre Matemática.

Num relatório publicado pelo Todos Pela Educação (TPE)³ lançado no mês de julho de 2015, é possível ter o contorno exato da qualidade da educação brasileira. O documento é resultado do acompanhamento das 5 Metas estabelecidas pelo movimento: (1) Toda criança e jovem de 4 a 17 anos na escola; (2) Toda criança plenamente alfabetizada até os 8 anos; (3) Todo aluno com aprendizado adequado ao seu ano; (4) Todo jovem de 19 anos com Ensino Médio concluído e (5) Investimento em Educação ampliado e bem gerido.

Os dados correspondentes a Meta 3, consideram os resultados obtidos na Prova Brasil e no Saeb, ambos desenvolvidos pelo Inep, que avaliam a qualidade do ensino a partir de provas padrões e questionários socioeconômicos. De acordo com o relatório, a média nacional de proficiência dos alunos do 3º ano do Ensino Médio em Português foi de 27,2% enquanto a de Matemática ficou em 9,3%, bem aquém da meta determinada para o período de 2013 que seria 39,0% para Português e 28,3% para Matemática.

Focando na disciplina de matemática, os dados revelaram que a Região Norte teve o pior rendimento alcançando 3,6%, atrás da Região Nordeste com 5,7%, da Região Centro-oeste com 9,5%, da Região Sudeste com 12,1% e da Região Sul com 12,2%. O estado da Bahia também repetiu os baixos índices marcando 4,6% de proficiência na disciplina, ficando em sexto lugar entre os nove estados nordestinos.

³ Fundado em 2006, o Todos Pela Educação é um movimento da sociedade brasileira que tem como missão contribuir para que até 2022, ano do bicentenário da Independência do Brasil, o País assegure a todas as crianças e jovens o direito a Educação Básica de qualidade. Fonte: <<http://www.todospelaeducacao.org.br/quem-somos/otpe/>>.

Veja os gráficos a seguir.

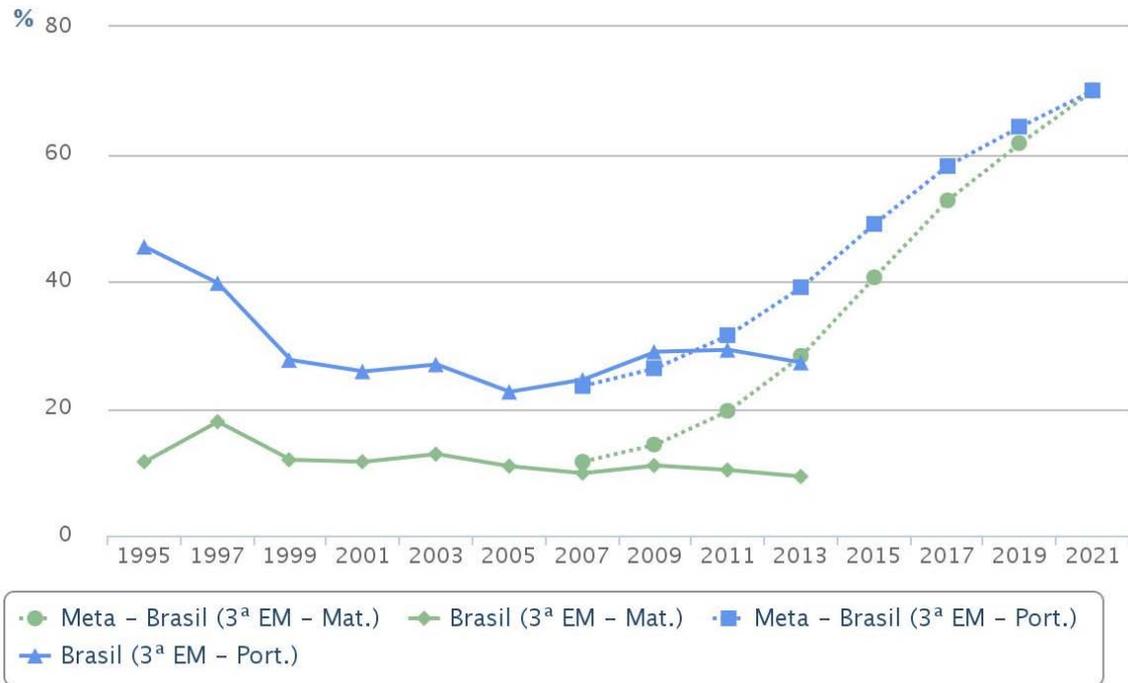


Figura 1 – Médias nacionais das proficiências de Português e Matemática entre os anos de 1995 a 2013 dos alunos do 3º Ano do Ensino Médio. Fonte: TPE.

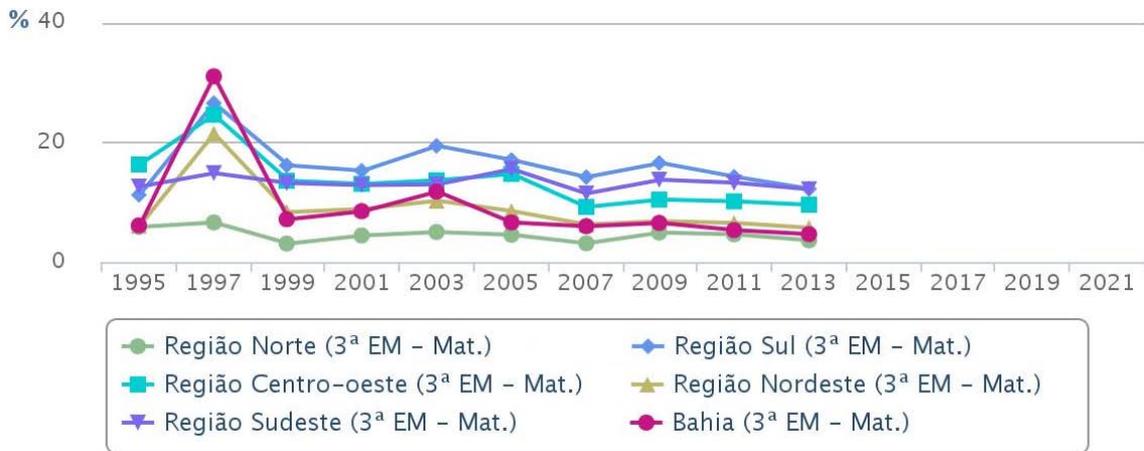


Figura 2 – Médias regionais e baiana das proficiências de Matemática entre os anos de 1995 a 2013 dos alunos do 3º Ano do Ensino Médio. Fonte: TPE.

Na Bahia, apenas os resultados das competências dos alunos do 4º/5º ano do Ensino Fundamental estão próximos das metas determinadas. A proficiência para as turmas do 8º/9º ano também provocam preocupação.

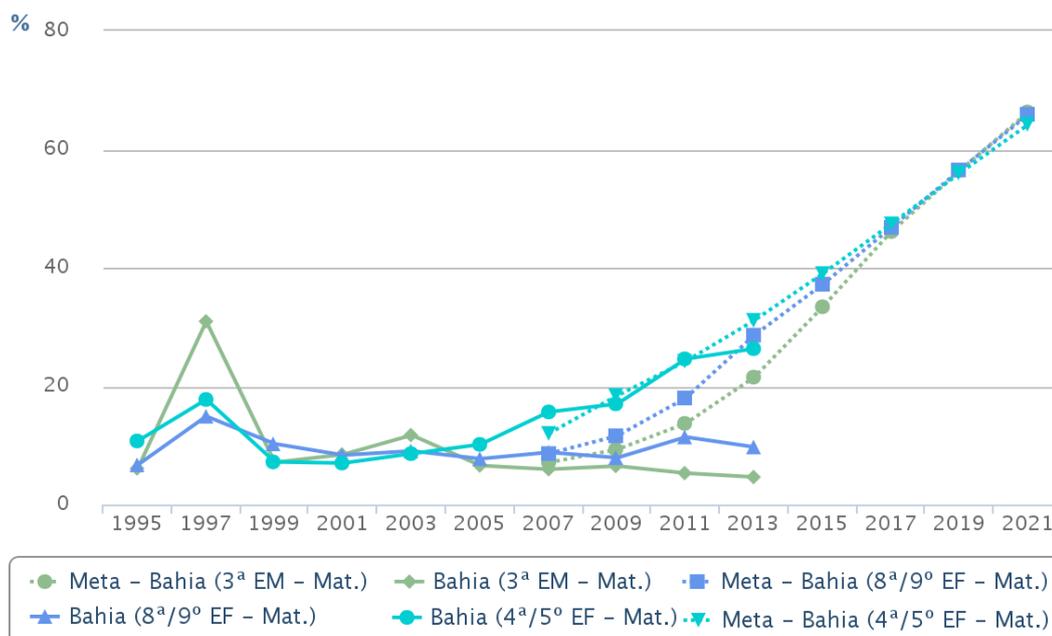


Figura 3 – Médias baianas das proficiências de Matemática entre os anos de 1995 a 2013 dos alunos do 4º/5º e 8º/9º ano do Ensino Fundamental e do 3º Ano do Ensino Médio. Fonte: TPE.

Percebe-se que à medida que o estudante avança nos anos escolares, seu rendimento na disciplina de Matemática cai. Isso se deve a vários fatores: mudança no ambiente escolar, onde uma professora é substituída por vários professores com métodos de ensinamentos diferentes; mudanças internas no indivíduo devido ao amadurecimento corporal; à abstração dos conteúdos matemáticos; às dificuldades impostas pelos docentes através da complexidade dos conteúdos, entre outros. Até mesmo os PCNs ventilam hipóteses à respeito da queda do rendimento escolar do aluno atrelado ao desinteresse na disciplina de Matemática:

Como esses conteúdos geralmente são abordados de forma linear e hierarquizada, apenas em função de sua complexidade, os alunos acabam tendo poucas oportunidades de explorá-los em contextos mais amplos. Mais ainda, as situações-problema raramente são colocadas aos alunos numa perspectiva de meio para a construção de conhecimentos. Essa organização linear e bastante rígida dos conteúdos, que vem sendo mantida tradicionalmente na organização do ensino de Matemática, é um dos grandes obstáculos que impedem os professores de mudar sua prática pedagógica numa direção em que se privilegie o recurso à resolução de problemas e a participação ativa do aluno (PCNs, 1998, p. 38).

Na pesquisa⁴ realizada com 2.632 pessoas, encomendada pelo Instituto Tim sob a coordenação do Ciclo da Matemática do Brasil⁵, revelou que 75% dos entrevistados não sabem

⁴ Para saber mais, acesse: <<http://institutotim.org.br/2015/11/06/como-e-a-matematica-dos-adultos/>>.

⁵ O Círculo da Matemática do Brasil é um projeto apoiado pelo Instituto TIM para melhorar o aprendizado da Matemática de crianças estudando em escolas públicas das cinco regiões do Brasil. O projeto tem como objetivo

médias simples, 63% não conseguem responder a perguntas sobre porcentuais e 75% não entendem frações. Note que são três exemplos de aplicações diretas e constantes da matemática no dia a dia de qualquer indivíduo inserido numa sociedade bem informada. Mas, 89% dos entrevistados alegaram que não usam a matemática no cotidiano.

Na contra mão destes dados negativos, outras pesquisas desenvolvidas por entidades e organizações privadas conseguem demonstrar a influência positiva da matemática na vida social do indivíduo, refutando o argumento do “não ver aplicação prática” — uma das principais causas do desinteresse do aluno — que agrava mais o rendimento escolar.

Segundo o Ipea⁶, num relatório⁷ publicado em outubro de 2013, alegou que o rendimento médio real de pessoas com maior escolaridade cresceu 6,3% de acordo com os dados do Pesquisa Nacional por Amostra de Domicílios (PNAD de 2012).

A Fundação Itaú Social⁸ foi mais longe e analisou⁹ duas gerações nascidas em 1977-78 e em 1987-88, observando três etapas da vida dos indivíduos: dos 4 aos 5 anos (infância); dos 17 a 18 anos (fase escolar) e dos 23 aos 24 (mercado de trabalho), através de dados do PNAD, do Instituto Brasileiro de Geografia e Estatística (IBGE), referente aos anos de 1982, 1992, 1995 e 2005 e do Censo Demográfico, realizado pelo mesmo órgão, além das informações do Saeb e do Inep. A conclusão foi que o aumento de 10% na nota de proficiência em Língua Portuguesa é responsável por um salário 5% maior, em média, cinco anos depois que o estudante concluiu a Educação Básica e está no mercado de trabalho. Já em Matemática, uma elevação de 10% na proficiência origina, em média, um salário 4,6% maior no mesmo período.

despertar nas crianças o gosto pela matemática e potencializar a sua aprendizagem. Fonte: <<http://www.ocirculodamatematica.com.br/>>.

⁶ O Ipea é uma fundação pública federal vinculada ao Ministério do Planejamento, Orçamento e Gestão. Suas atividades de pesquisa fornecem suporte técnico e institucional às ações governamentais para a formulação e reformulação de políticas públicas e programas de desenvolvimento brasileiros. Fonte: <http://www.ipea.gov.br/portal/index.php?option=com_content&view=article&id=1226&Itemid=68>.

⁷ Para saber mais, acesse: <http://www.ipea.gov.br/portal/index.php?option=com_content&view=article&id=20067>.

⁸ Criada especificamente para estruturar e implementar os investimentos sociais do Itaú, a Fundação Itaú Social, com atuação em todo o território brasileiro, tem como foco o apoio, elaboração e fortalecimento de programas que visam o progresso das políticas públicas de educação, a potencialização do engajamento de seus colaboradores – e da sociedade em geral – por meio da cultura de voluntariado, e a avaliação sistematizada de projetos sociais. Fonte: <<https://www.itausocial.org.br/crianca/a-fundacao/>>.

⁹ Para saber mais, acesse: <www.redeitausocialdeavaliacao.org.br/blog/o-impacto-da-educacao-nos-salarios-dos-jovens-brasileiros/>.

2.2 Os assuntos escolhidos

Por experiência, é possível afirmar que existe uma provocação velada à respeito da aplicabilidade de vários conteúdos matemáticos. Além disso, se torna um grande motivador da rejeição do aluno com a disciplina e ao mesmo tempo serve de justificativa para várias correntes de pesquisa que defendem a exclusão de alguns conteúdos das matrizes curriculares.

Para responder a este desafio foram escolhidos três assuntos: equações do segundo grau, matrizes e sistemas lineares. Estes conteúdos são, entre outros, considerados abstratos e apontados como irrelevantes pela pouca utilização prática no cotidiano do aluno, e por isso são bons recursos para mostrar o lado eficiente da Matemática. Além disso, há outros motivos:

a) Possuem resoluções mecanizadas: O propósito deste projeto é construir aplicativos para serem executados em dispositivos móveis. Mas a construção será realizada por professores e estudantes que em sua maioria não possuem destrezas em informática, ou ainda, nunca participaram de algum projeto relacionado à programação. Deste modo, fica inviável conceber projetos que não usam métodos conhecidos (fórmulas) em seus algoritmos. Portanto, equações do segundo grau, operações com matrizes e sistemas lineares permitem usar regras predefinidas facilitando a depuração dos programas.

b) São considerados abstratos e enfadonhos: De fato, os métodos de resoluções destes conteúdos são tediosos e dificilmente vem acompanhados de aplicações práticas cotidianas capazes de convencer o aluno da importância daqueles assuntos. Assim, cada sequência didática vem acompanhada de um exercício-desafio, levando o estudante a interiorizar os mecanismos de resolução para alcançar o resultado esperado na atividade paralela (desafio).

c) São recorrentes: As definições dos três temas são conhecimentos âncoras e periodicamente são resgatados para colaborar com a fixação de novos conteúdos, seja no ensino médio ou em cursos superiores. Portanto é necessário criar uma boa retenção destes assuntos para melhorar o rendimento do estudante.

2.3 A linguagem Logo

O marco histórico no qual este trabalho foi inspirado se deu na década de 1960, com a criação da linguagem computacional chamada Logo, desenvolvida e aprimorada pela equipe de cientistas formada por Daniel G. Bobrow, Wallace Feurzeig, Richard Grant e Cynthia Solomon, todos estes integrantes da Bolt, Beranek and Newman Technologies (BBN)¹⁰, e Seymour Papert do MIT.

Em 1966, a linguagem Logo começou a ser escrita por Bobrow quando trabalhava na BBN. De acordo com a empresa, a Logo deveria permitir que alunos do ensino básico com pouca preparação pudessem ser capazes de usá-la para tarefas simples, tivesse sua estrutura incorporada com conceitos importantes da matemática com o mínimo de interferência de ajustes de programação, bem como usassem a expressão matematicamente não numérica, e ainda algoritmos numéricos.

A BBN também estava envolvida ativamente no trabalho de inteligência artificial durante os anos 60. Eles colaboraram com pesquisadores do MIT estudando a compreensão da linguagem natural, reconhecimento de padrões, e instrução assistida por computador.

Um dos objetivos do projeto original da Logo eram que a linguagem fosse usada para apoiar a aprendizagem. Contudo, a Logo não foi criada apenas para ensinar habilidades de programação. Também foi projetada com finalidades mais elevadas, tais como ajudar a ensinar habilidades metacognitivas¹¹ e resolução de problemas, permitindo que os alunos aprendessem de forma construtivista, criando projetos interessantes com o computador.

Algumas das principais características da linguagem Logo são modularidade, extensibilidade, interatividade e flexibilidade. Estas características de projeto estão intimamente relacionados com os objetivos educacionais do software. Isto se dá porque essas peculiaridades permitem que os estudantes possam rapidamente aprender a usá-lo e programadores experien-

¹⁰ A BBN é uma empresa de alta tecnologia americana que fornece serviços de pesquisa e desenvolvimento.

¹¹ A metacognição é um conceito cujo propósito é levar o indivíduo a discutir e a pensar sobre como faz as coisas, e como se aprende, ou ainda analisar e avaliar como se conhece. A expressão é formada pelas palavras *meta* “reflexão sobre si” e *cognição* “processo ou faculdade de adquirir um conhecimento” ou “percepção, conhecimento”.

tes possam fazer uso de seus recursos avançados para construir programas complexos.

Nas palavras de FREIRE (1999),

A modularidade possibilita definir procedimentos de tal forma que possam ser reaproveitados em diferentes programas computacionais (um mesmo procedimento quadrado pode ser usado para fazer a janela de uma casa ou a composição de uma figura com vários quadrados); a extensibilidade permite expandir os comandos do Logo via programação de novos procedimentos que ficam disponíveis para serem utilizados por aquele usuário particular (além dos comandos disponíveis no Logo, o sujeito pode definir procedimentos que funcionam como novos comandos como, quadrado, triângulo, círculo, etc.); a interatividade, proporciona ao usuário um feedback imediato dos comandos utilizados (na forma de ações produzidas pela Tartaruga ou mensagens de erros) e a flexibilidade, permite resolver um mesmo problema de diferentes formas (uma mesma figura pode ser desenhada de várias maneiras, dependendo da sequência de comandos utilizada).

Não por acaso que a Logo se tornou pioneira no uso de softwares nos processos de ensino-aprendizagem. Antes de vir para Grupo de Inteligência Artificial do MIT em 1964, Seymour Papert estudou com Jean Piaget em Genebra, Suíça, durante cinco anos. Piaget é o fundador do construtivismo, uma forte teoria educacional que afirma que as crianças aprendem melhor fazendo. Ele viu os estudantes como construtoras de suas próprias estruturas intelectuais. Papert foi fortemente influenciado pelas teorias de Piaget sobre cognição infantil.

O interesse de Papert estava no desenvolvimento infantil, na aprendizagem e na natureza do pensamento. Ele percebeu que os computadores e a computação eram de longe a mais rica fonte conhecida de atividades construtivistas que poderia dar às crianças um poder sem precedentes de inventar e realizar projetos interessantes. Papert partiu da hipótese de que se as crianças ensinassem os computadores como pensar (na forma de programar o computador), eles explorariam a forma como a se pensar. Ele afirmou que a melhor maneira de aprender é ensinar alguém — no caso programação — o conteúdo. Assim, uma linguagem que tornou fácil para as crianças a programar um computador deve oferecer oportunidades de aprendizagem imensas.

Embora Papert tenha contribuído para o Logo original como consultor da BBN no final dos anos 1960, ele é provavelmente mais conhecido pela a adição da famosa tartaruga no

Logo em 1971 e pelo seu livro *Mindstorms*¹² lançado em 1980. Na primeira versão, a tartaruga da Logo era um equipamento que estava conectado ao computador. Estudantes colocavam a tartaruga em um grande pedaço de papel e, em seguida, davam ordens (executavam comandos) no computador, como esquerda, direita, para frente e para trás, de modo que a tartaruga desenhasse no papel. Mais tarde, a tartaruga se transformou num símbolo gráfico na tela que poderia desenhar linhas e imagens respondendo aos mesmos comandos que direcionavam as tartarugas de metal.



Figura 4 – A tartaruga original do "Mindstorms", Seymour Papert, Basic Books, Inc., 1980 (esquerda), e o TI 99/4 ao lado do Apple II. Ambos podiam ser ligados em televisores (direita).

O livro *Mindstorms* elevou o uso da Logo a novos níveis, inspirando professores a pensar sobre seus estudantes como crianças com potencial ilimitado. A obra discutiu como a Logo poderia ser usado em salas de aula e explicou como as teorias pedagógicas de Papert e os computadores poderiam ajudar os alunos a aprender. Nesta década, os microcomputadores estavam se tornando mais baratos, viabilizando a inserção destas máquinas nas salas de aula. Naquela época, escolas públicas estadunidenses foram ganhando computadores tais como o TI 99/4 e o Apple II que poderiam rodar o software Logo.

¹² *Mindstorms: Children, Computers and Powerful Ideas* é um livro de Papert lançado em 1980, onde o autor propõe um ambiente único de aprendizado baseado em computação, que denomina de "*Microworld*" (micro mundo, numa tradução literal). A sua crença principal sobre o "design" do Microworld é que ele complementa os mecanismos naturais de construção de conhecimento das crianças, conhecidos como abordagem construtivista para conhecer e aprender. A sua primeira implicação é que a aprendizagem Microworld afetará profundamente a qualidade do conhecimento adquirido. Esta obra é uma das primeiras grandes tentativas de mediar a tecnologia educacional baseada em computador com as teorias de Jean Piaget sobre a aprendizagem e o conhecimento. Fonte: Wikipédia.

2.4 Do App Inventor para o MIT App Inventor 2

O *App Inventor* (nome original até ser transferido para o MIT) é uma plataforma de desenvolvimento, que permite pessoas com qualquer nível de experiência em programação criarem programas (aplicações) para o sistema operacional Android. Ele usa uma interface gráfica onde a funcionalidade dos componentes é exposta aos desenvolvedores via blocos de código permitindo construir o aplicativo sem ter que escrever código tradicional, tal como montar um quebra-cabeça.

O *App Inventor* foi desenvolvido sobre um trabalho anterior, feito em linguagens de programação gráfica como a *StarLogoTNG* e a biblioteca *Openblocks* para fornecer um quadro de desenvolvimento de aplicativos que oferece aos usuários sem experiência em codificação a capacidade de criar aplicações móveis.



Figura 5 – Exemplo de um bloco de programação referente à tarefa de potenciação.

No outono de 2009, o *App Inventor* foi introduzido por um programa piloto em uma dúzia de universidades estadunidenses como uma ferramenta para ajudar a ensinar os alunos sobre vários temas relacionados com a ciência da computação, privacidade digital e a importância da tecnologia na sociedade.

Durante o semestre, os alunos criaram várias aplicações para smartphones e exploraram algumas das dificuldades em desenvolver em plataforma móvel. Na sequência deste programa piloto, o *Google Research* aprimorou o projeto, tendo sido efetivamente lançado em julho de 2010. Naquela época, o *App Inventor* era fornecido ao público como parte dos serviços do *Google Labs*.

Em agosto de 2011 o *Center For Mobile Learning* (Centro de Aprendizagem Móvel) foi criado no *MIT Media Lab* (Instituto de Tecnologia de Massachusetts - Laboratório de Mídia) a fim de continuar a oferecer o *App Inventor* ao público. No quarto trimestre de 2011, o *Center For Mobile Learning* começou a trabalhar no *App Inventor*, e em março de 2012 uma nova versão foi lançada, passando a chamar-se *MIT App Inventor 2* (MAI).

Segundo o próprio *MIT App Inventor*¹³, ele é uma ferramenta que visa tornar a programação e a criação de aplicativos acessíveis a uma grande variedade de públicos como: educadores formais e informais que usam para apresentar a programação aos seus alunos de ciência da computação; educadores que desenvolvem aplicativos para apoiar seus próprios objetivos instrucionais; governos e entidades civis que criam aplicativos personalizados, muitas vezes hiper-locais na resposta a desastres naturais e necessidades; pesquisadores que criam aplicativos para satisfazer as suas necessidades de coleta e análise de dados para sua pesquisa em uma grande variedade de campos, além de entusiastas e empreendedores.

2.5 O MIT App Inventor na sociedade

Muitas organizações e indivíduos com diferentes objetivos utilizam o MAI. Isto é possível graças a sua acessibilidade, tanto no sentido de habilidades de programação como no sentido de ter acesso à tecnologia. O MAI está sendo usado em uma variedade de configurações, incluindo a educação em salas de aula que vão desde a escola primária à faculdade além de programas de pós-graduação.

Enquanto muitos ensinam ciência da computação usando o MAI, há também inúmeros educadores, utilizando-o como uma ferramenta para envolver os alunos no ensino de qualquer assunto. Esses programas muitas vezes se concentram em incentivar grupos que normalmente são sub-representados em tecnologia para se sentirem habilitadas para serem as criadoras da tecnologia (escolas públicas).

Além disso, o MAI também está sendo usado para fomentar o espírito empresarial,

¹³ Fonte: <http://appinventor.mit.edu/explore/about-us.html>

onde gestores estão aproveitando a capacidade de colocar de forma rápida e barata protótipos de suas ideias, que podem ser usados para lançar no mercado. Usando o MAI elimina-se tempo e trabalho que é feito muitas vezes antes que haja ainda qualquer financiamento para um projeto, pois não é mais necessário saber programar, a fim de publicar uma idéia com um exemplo funcional.

2.6 Por que o MIT App Inventor?

Dispositivos móveis estão se infiltrando em todos os aspectos de nossas vidas e, atualmente, simbolizam uma era de prosperidade e democratização das tecnologias de ponta para indivíduos de todas as classes sociais. Portanto, a escola como elo entre o estudante e a realidade social, não pode rejeitar o uso destes aparelhos privilegiando apenas o computador como representante tradicional de tecnologia.

A Figura 6 compara os preços de computadores e aparelhos móveis. À esquerda e acima, ela mostra um computador sendo vendido em 1991 por Cr\$ 280.000,00, enquanto o salário mínimo era de Cr\$ 12.325,60. Ao lado está o anúncio de outro computador no panfleto do Extra, publicado em novembro de 2000, que custava R\$ 1.290,00 para um salário mínimo de R\$ 151,00, e abaixo, um smartphone intermediário sendo negociado por R\$ 593,00 onde o salário é de R\$ 788,00.

The figure consists of three panels illustrating the evolution of technology products and their prices over time:

- Top Left:** A video frame showing a PC advertisement from 1991. The text reads "Comercial Computador PC XT e PC AT 1991" and "Cr\$ 280.000,00".
- Top Right:** A flyer for a computer from 2000. The text includes "COMPUTADOR METRON CELERON 550 MHz MULTIMEDIA 32 MB MEMORIA RAM HD 6.4 GB. CD-ROM 4X E MONITOR 15\" and "A VISTA R\$ 1.290,00 OU 12X (0+12) R\$ 121,67".
- Bottom:** A screenshot of a mobile phone advertisement for a Moto G. The text includes "Celular Desbloqueado Moto G™ 2ª Geração Dual Preto com Tela de 5\" and "Preço: R\$ 593,10".

Figura 6 – Anúncios de produtos tecnológicos em diferentes décadas.

Utilizar estas ferramentas – que são acessíveis aos usuários de todos os níveis de experiência, é democratizar o processo de ensino, ratificando para o aluno e para o professor que objetos de sua propriedade, por mais simples que sejam, também são úteis para alcançar os objetivos propostos. Além disso, desenvolver os próprios métodos para resolver problemas, é um elemento essencial para ampliar suas habilidades.

O MAI une ambos desafios, pois ele é projetado para pessoas sem experiência e a codificação é livre para usar. Se um indivíduo – professor ou aluno, tem acesso a um computador com ingresso à Internet, então eles podem executar o MAI, sem nenhum custo adicional. Não é apenas o MAI, uma plataforma livre para usar, como também o sistema operacional para a qual ele faz aplicativos – Android, também é livre, permitindo que qualquer desenvolvedor crie um programa sem ter que pagar para usar o sistema operacional.

Outro motivo da escolha do MAI é que um de seus objetivos coincide perfeitamente com o deste projeto, qual seja, incentivar as pessoas a tornarem-se produtores, em vez de apenas os consumidores da tecnologia. O MAI atinge perfeitamente este fim, fornecendo a plataforma gratuitamente e sendo mantido por uma das mais conceituadas instituições de ensino superior do planeta.

2.7 A tecnologia disponível

Segundo pesquisa¹⁴ publicada em abril de 2015 pela IDC Brasil¹⁵, em 2014 foram comercializados 54.5 milhões de aparelhos smartphones, demonstrando um crescimento de 55% na comparação com 2013. Além disto, no mesmo período o mercado brasileiro consumiu 70.3 milhões de aparelhos *feature phones*¹⁶, registrando alta de 7% sobre 2013. Estas

¹⁴ Para saber mais, acesse: <<http://br.idclatin.com/releases/news.aspx?id=1801>>.

¹⁵ A International Data Corporation Pesquisa de Mercado e Consultoria Ltda – IDC Brasil é uma empresa do Ramo de inteligência de mercado e consultoria nas indústrias de tecnologia da informação, telecomunicações e mercados de consumo em massa de tecnologia. Analisa e prediz as tendências tecnológicas para que os profissionais, investidores e executivos possam tomar decisões de compra e negócios nestes setores.

¹⁶ Um *feature phone* (tradução: **telefone de recurso**) é o termo normalmente usado para descrever uma classe de telefone celular de baixo nível, que são limitados em capacidades comparados com uma moderna do smartphone. Tipicamente fornecem chamadas de voz e funcionalidade de mensagens de texto, além de básico multimídia e internet, e outros serviços oferecidos pelo provedor de serviços sem fio do usuário. (Fonte Wikipédia).

vendas colocaram o Brasil na 4ª colocação entre os maiores mercados do mundo, atrás da China, Estados Unidos e Índia. Em março, a mesma empresa havia publicado¹⁷ que, naquele período de 2014, os brasileiros compraram cerca de 9,5 milhões de tablets, ante 8,4 milhões em 2013, fincando uma alta de 13% em volume de vendas.

Esta possibilidade de acesso às tecnologias de mídia e comunicação: smart tv, telefonia móvel, internet banda larga, smartphones, tablets e notebooks não só representa a acessibilidade das novidades pelas massas, como também garante a liberdade do indivíduo diante de antigos padrões de coletividade. Ter e estar representado por determinados aparelhos e contas em redes sociais, posiciona o sujeito em grupos ou classes das quais ele não participaria há alguns anos.

Junto com este movimento que permite uma aproximação do que é considerado ser rico ou ser pobre através do que se pode comprar, vem o lado negativo do exagero consumista, proporcionado pela estabilidade econômica e oferta de crédito acessível, estimulado pela mídia que vende a idéia do quanto mais caro melhor.

Naturalmente, não cabe nestas linhas discutir a repercussão das tendências sociais e seu consumismo. Porém, deve-se lembrar que o ambiente escolar reflete quase que imediatamente o comportamento social na qual os alunos estão inseridos. Observe que possuir dispositivos móveis e ter acesso a contas sociais é um cenário nunca antes experimentado pela parcela social cliente da rede pública de ensino. Estamos falando de estudantes que já no terceiro ano escolar possuem seu próprio aparelho celular, enquanto seus pais tiveram os primeiros acessos somente na fase adulta.

Além disto, o laboratório de informática não representa mais o ápice do encontro entre estudante e tecnologia. Veja que, enquanto era necessário distribuir os alunos em grupos para otimizar as poucas máquinas disponíveis, o estudante atual possui sua própria estação de trabalho no bolso, devidamente atualizado, operando, e principalmente, personalizado. Portanto está aí um recurso de alta tecnologia, em grande quantidade e subutilizado. Então porque não pode ser aproveitado?

¹⁷ Para saber mais, acesse: <<http://br.idclatin.com/releases/news.aspx?id=1785>>.

2.8 Por que programar?

Na voz de VALENTE (1999), o processo de aprendizagem pode ocorrer basicamente de duas maneiras: a informação é memorizada ou é processada pelos esquemas mentais e esse processamento acaba enriquecendo esses esquemas. A proposta deste trabalho está alicerçada no processo de esquematização mental, através do uso das tecnologias de informações, onde, o estudante cria seu próprio programa, utilizando conceitos, estratégias e um estilo de resolução de problemas. O diferencial está no uso das novas tecnologias inseridas no cotidiano do estudante na última década: dispositivos móveis.

Além disso, “a ferramenta computacional pode ser o instrumento que permita romper com a abordagem instrucionista que caracteriza a educação tradicional em prol de uma educação progressista”, (Papert, 1994, p. 20 apud ALMEIDA, 1999, p. 30), chamando a atenção do aluno para o novo, possibilitando aflorar habilidades ocultas que não são reveladas no método clássico. Isto se deve ao fato no “qual alunos e educadores se engajem num trabalho de investigação científica, em que ocorre: o processo cíclico ação-testagem-depuração-generalização; o autodomínio na representação e o estabelecimento de conexões entre conhecimentos que o aluno possui – o velho – para a construção de um novo conhecimento” (ALMEIDA, 1999, p. 28).

Mas é prudente alertar que

(...) os professores precisam entender que a entrada da sociedade na era da informação exige habilidades que não têm sido desenvolvidas na escola, e que a capacidade das novas tecnologias de propiciar aquisição de conhecimento individual e independente implica num currículo mais flexível, desafia o currículo tradicional e a filosofia educacional predominante, e depende deles a condução das mudanças necessárias (STAHL, 2008, p. 299 apud ALTOÉ, FUGIMOTO, 2009, p. 6),

eis que “o professor da disciplina curricular deve ter conhecimento sobre os potenciais educacionais do computador e ser capaz de alternar adequadamente atividades tradicionais de ensino-aprendizagem e atividades que usam o computador” (VALENTE, 1999).

Porém o que mais vemos é a relutância de boa parte dos docentes em utilizar ferra-

mentas computacionais para auxiliar seu trabalho. Conforme Martinez (2004, p. 105, apud ALTOÉ, FUGIMOTO, 2009, p. 3) “a maioria dos professores em serviço não tem conhecimento sobre como se utilizam essas ferramentas ou quais são suas possibilidades na sala de aula”. Não obstante, quando eles usam qualquer meio para incrementar sua aula, acabam por tê-la como “um recurso perpetuador do ensino baseado na instrução e no controle, que não é adequado ao equacionamento de problemas sociais emergentes e ao desenvolvimento da autonomia e da autocrítica”. (ALMEIDA, 1999, p. 38).

Assim sendo, “a atividade de programação permite observar e descrever as ações do aluno enquanto ele resolve problemas que envolvem abstrações, aplicações de estratégias, estruturas e conceitos já construídos, ou a criação de novas estratégias, estruturas e conceitos” (Papert apud ALMEIDA, 1999, p. 33). É o que diz também Valente:

Quando o aluno usa o computador para construir o seu conhecimento, o computador passa a ser uma máquina para ser ensinada, propiciando condições para o aluno descrever a resolução de problemas, usando linguagens de programação, refletir sobre os resultados obtidos e depurar suas idéias por intermédio da busca de novos conteúdos e novas estratégias. (VALENTE, 1999).

Para finalizar, ainda nos argumentos de Valente, o ato de programar permite ao aluno estabelecer uma sequência lógica de tarefas descritas da seguinte forma:

- **DESCRICHÃO:** é a exposição da resolução do problema em termos da linguagem de programação, utilizando a estrutura de conhecimento para representar.
- **EXECUÇÃO:** é a construção da DESCRICHÃO na linguagem lógica do computador. Essa execução fornece um retorno fiel e imediato, desprovido de qualquer animosidade ou afetividade que possa haver entre o aluno e o computador. O resultado obtido é fruto somente do que foi solicitado à máquina.
- **REFLEXÃO:** é a análise do que foi produzido pelo computador. O processo de refletir sobre o resultado do programa pode acarretar uma das seguintes ações alternativas: ou o aprendiz não modifica o seu procedimento porque as suas idéias iniciais sobre a resolução daquele problema correspondem aos resultados apresentados pelo computador e, então, o problema está

resolvido; ou depura¹⁸ o procedimento, quando o resultado é diferente da sua intenção original;

- **DEPURAÇÃO**: é o último passo do ciclo, onde o aprendiz pode buscar informação sobre conceitos de uma determinada área, alguma convenção da linguagem de programação, computação ou estratégias. Essa informação é assimilada pela estrutura mental (passa a ser conhecimento) e utilizada no programa para modificar a descrição anteriormente definida.



Figura 7 – Ciclo das sequências lógicas de tarefas no ato da programação.

2.9 Antes de começar

Provavelmente, este deve ser o primeiro contato do leitor com a plataforma MIT App Inventor 2, por isso alguns ajustes devem ser feitos para a plena execução do trabalho com a observação de dois detalhes. O primeiro trata-se da configuração do computador e do smartphone/tablet com a instalação dos programas necessários, e o segundo é conhecer a plataforma na qual o projeto será concebido.

Vale reforçar que antes de aplicar o projeto na escola, o docente deve testar toda a sequência para experimentar os possíveis cenários que sugerem dúvidas para conseguir fazer o devido acompanhamento dos estudantes no laboratório. Além disso, é interessante iniciar a aula apresentando a plataforma, seus elementos e componentes concluindo com um experi-

¹⁸ Depurar: (Informática) examinar o código de um programa, a fim de identificar, localizar e suprimir falhas ou erros. Fonte: *Wikcionário: O dicionário livre*.

mento simples, por exemplo, construir um aplicativo de soma e testá-lo no celular. Abaixo segue a lista de anexos que devem ser consultados:

- ANEXO A – O QUE INSTALAR – Tutorial para configurar o computador;
- ANEXO B – CONHECENDO A PLATAFORMA. – Apresentação dos componentes da plataforma;
- ANEXO C – CONSTRUINDO O PRIMEIRO APLICATIVO – Tutorial para construir um aplicativo que executa a soma entre dois números inseridos pelo usuário.

3 EQUAÇÕES DO 2º GRAU

As funções polinomiais do segundo grau ou funções quadráticas ocupam importante espaço no currículo de Matemática no Ensino Fundamental e no Ensino Médio. Isto se dá aos inúmeros empregos que estas funções possuem no cotidiano, disfarçadas pela aplicação direta das parábolas geradas pela parte algébrica das funções. Além disto, funções de modo geral, como as afins (primeiro grau), exponenciais, logarítmicas, seno e cosseno acompanham o estudante em cursos das ciências exatas e biológicas.



Figura 8 – Exemplos de funções quadráticas representadas por parábolas.

Contudo é perceptível uma grande quantidade de estudantes que não conseguem assimilar as principais definições e resultados do conteúdo de equações do segundo grau, entre outros tópicos matemáticos. Prova disso, são os resultados apresentados no início deste trabalho que apontam o déficit de proficiência na disciplina de Matemática. E sendo a função quadrática a mais tratada no Ensino Médio, é imprescindível que tais conceitos sejam bem fixados no processo de ensino-aprendizagem.

Este capítulo contém uma sequência didática cujo objetivo é construir um aplicativo destinado a encontrar as raízes de equações do segundo grau, dentro do conjunto dos números reais, utilizando para isso, o método de Bháskara.

3.1 Principais resultados.

Nesta seção serão listadas as principais definições e conceitos de funções quadráticas que nortearão a sequência didática.

Definição 3.1: Uma função $f: \mathbb{R} \rightarrow \mathbb{R}$ chama-se quadrática quando existem números reais a , b , c com $a \neq 0$ tais que $f(x) = ax^2 + bx + c$, para todo $x \in \mathbb{R}$.

A representação gráfica da função quadrática é uma parábola, tendo concavidade voltada para cima quando o coeficiente a é positivo, e concavidade voltada para baixo quando o coeficiente a é negativo.

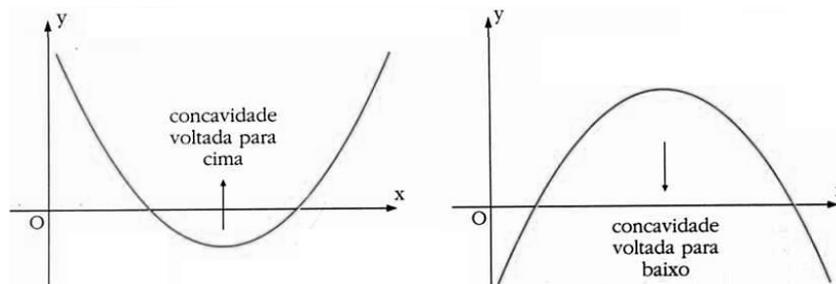


Figura 9 – Representação gráfica de funções quadráticas com $a > 0$ e $a < 0$, respectivamente.

O ponto $P(0, y)$, no qual o gráfico da função quadrática intersesta o eixo das coordenadas, ocorre quando $y = c$, eis que, substituindo a variável x por zero, em $f(x) = ax^2 + bx + c$ tem-se que $f(0) = a(0^2) + b(0) + c = c$.

Definição 3.2: Um número x^{19} chama-se raiz da equação $f(x) = 0$ ou zero da função $f(x)$, se $f(x)=0$.

A fórmula de Bhaskara é um método prático para determinar as raízes da equação $f(x) = 0$, onde $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$. Uma forma de chegar a esta equação é utilizando o método para completar os quadrados procedendo da seguinte maneira: dada a equação $ax^2 + bx + c = 0$, é possível dizer que $4a(ax^2 + bx + c) = 0$, assim $4a^2x^2 + 4abx + 4ac = 0$. Completan-

¹⁹ Neste trabalho, serão consideradas apenas as soluções pertencentes ao conjunto dos números reais.

do os quadrados, fica:

$$4a^2x^2 + 4abx + 4ac = 0 \Rightarrow 4a^2x^2 + 4abx + 4ac + b^2 - b^2 = 0$$

$$4a^2x^2 + 4abx + 4ac + b^2 - b^2 = 0 \Rightarrow (2ax + b)^2 + 4ac - b^2 = 0.$$

Então,

$$(2ax + b)^2 + 4ac - b^2 = 0 \Rightarrow (2ax + b)^2 = b^2 - 4ac$$

$$2ax + b = \pm\sqrt{b^2 - 4ac} \Rightarrow 2ax + b = \begin{cases} \sqrt{b^2 - 4ac} \\ -\sqrt{b^2 - 4ac} \end{cases}$$

logo

$$x = \begin{cases} \frac{\sqrt{b^2 - 4ac} - b}{2a} \\ \frac{-\sqrt{b^2 - 4ac} - b}{2a} \end{cases}.$$

Atribuindo ao discriminante Δ o valor de $(b)^2 - 4.a.c$, tem-se: $x = \frac{-b \pm \sqrt{\Delta}}{2a}$.

Desta equação, conclui-se então que: se $\Delta > 0$, haverá duas soluções reais e distintas para a equação $f(x) = 0$, o que significa que o gráfico da função $f(x)$ intersecará o eixo x duas vezes; quando $\Delta = 0$, existirá apenas uma solução real, e portanto a função tocará o eixo x apenas uma vez, sem intersecá-lo; e quando $\Delta < 0$, o gráfico da função não cruzará o eixo x , pois não existe solução no conjunto dos números reais para raiz quadrada de números negativos.

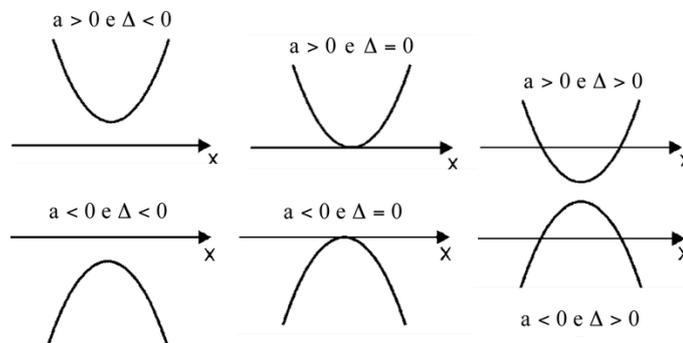


Figura 10 – Representações gráficas de funções quadráticas de acordo com o sinal de a e de Δ .

Quando a concavidade do gráfico for voltada para cima, a função possuirá ponto de mínimo absoluto. Caso a concavidade esteja voltada para baixo, a função terá máximo absoluto. Estes locais, máximos e mínimos, são chamados de vértice da parábola.

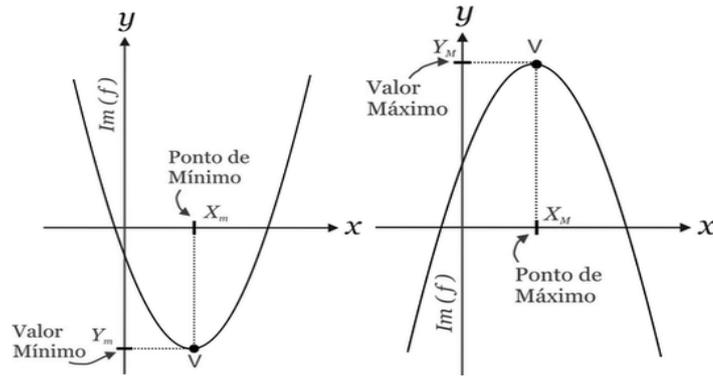


Figura 11 – Pontos de mínimos e máximos de gráficos de funções quadráticas.

A parábola pode ser definida como sendo uma curva plana formada pelo conjunto dos pontos que são equidistantes de uma reta e um ponto dados, chamados de diretriz e foco respectivamente (LIMA, 2012, p. 125). Isto implica imediatamente na posição do vértice em relação às raízes (ou possíveis raízes) da equação. Observe que, se os conjuntos de pontos que formam a parábola são equidistantes (Figura 12), então as raízes x_0 e x_1 também são equidistantes da reta diretriz ou eixo de simetria.

Isto leva a concluir que o valor da abscissa do vértice, que será chamado de ponto $V(x_V, y_V)$, está à meia distância de x_0 e x_1 , ou seja, a média aritmética.

Assim, como $x_0 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ e $x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$ são os zeros da função quadrática, fica então que $x_V = \frac{\frac{-b + \sqrt{b^2 - 4ac}}{2a} + \frac{-b - \sqrt{b^2 - 4ac}}{2a}}{2} = \frac{2b}{2} = -\frac{b}{2a}$.

Como trata-se de um par ordenado, sendo o valor de $x_V = -\frac{b}{2a}$, é possível substituir na função genérica de $y = f(x) = ax^2 + bx + c$, onde:

$$y_V = a\left(-\frac{b}{2a}\right)^2 + b\left(-\frac{b}{2a}\right) + c = a\left(\frac{b^2}{4a^2}\right) - \frac{b^2}{2a} + c$$

$$y_V = \frac{b^2}{4a} - \frac{b^2}{2a} + c = \frac{b^2 - 2b^2 + 4ac}{4a} = \frac{-b^2 + 4ac}{4a}.$$

Veja que $-b^2 + 4ac$ é o valor oposto do discriminante Δ , logo, $y_V = -\frac{\Delta}{4a}$. Então, o par

ordenado do ponto V é:

$$V(x_V, y_V) = \left(-\frac{b}{2a}, -\frac{\Delta}{4a}\right).$$

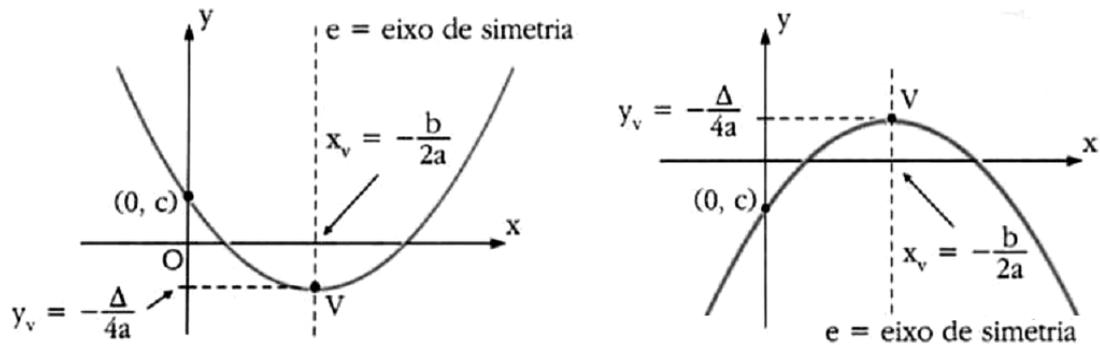


Figura 12 – A parábola, o eixo de simetria e o vértice.

3.2 Exemplos.

Exemplo 3.1: Dadas as funções quadráticas abaixo determine, se possível: a) os pontos que o gráfico intersecta o eixo x pelo método de Bhaskara; b) o vértice da parábola; c) o esboço do gráfico.

i) $f(x) = x^2 - 3x - 10$.

ii) $g(x) = 3x^2 + 2x + 4$.

Solução:

i) $f(x) = x^2 - 3x - 10$.

a) As raízes da equação $x^2 - 3x - 10 = 0$ pelo método de Bhaskara são:

$$x_0 = \frac{-(-3) + \sqrt{(-3)^2 - 4(1)(-10)}}{2(1)} = 5.$$

$$x_1 = \frac{-(-3) - \sqrt{(-3)^2 - 4(1)(-10)}}{2(1)} = -2.$$

b) O vértice da parábola:

$$V(x_V, y_V) = \left(-\frac{b}{2a}, -\frac{\Delta}{4a}\right) = \left(\frac{-(-3)}{2(1)}, -\frac{((-3)^2 - 4(1)(-10))}{4(1)}\right) = \left(\frac{3}{2}, -\frac{49}{4}\right).$$

c) Esboço do gráfico:

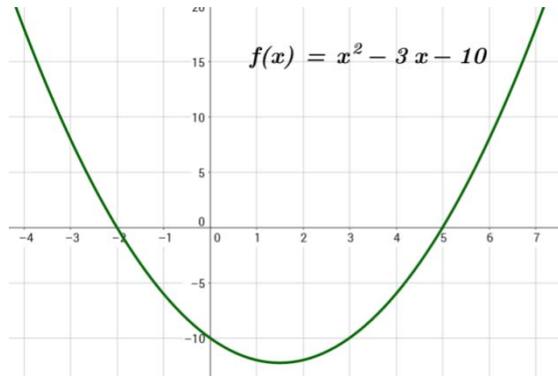


Figura 13 – Gráfico da função $f(x) = x^2 - 3x - 10$.

ii) $g(x) = 3x^2 + 2x + 4$.

a) As raízes da equação $3x^2 + 2x + 4 = 0$ pelo método de Bhaskara:

$$x_0 = \frac{-(-2) + \sqrt{(-2)^2 - 4(3)(4)}}{2(3)} = \frac{-2 + \sqrt{-44}}{6}$$

$$x_1 = \frac{-(-2) - \sqrt{(-2)^2 - 4(3)(4)}}{2(3)} = \frac{-2 - \sqrt{-44}}{6}$$

Veja que o valor do discriminante Δ é negativo, e portanto não há soluções dentro do conjunto dos números reais, e deste modo o gráfico da função não interseca o eixo x .

b) O vértice da parábola:

$$V(x_V, y_V) = \left(-\frac{b}{2a}, -\frac{\Delta}{4a}\right) = \left(\frac{-(-2)}{2(3)}, -\frac{((-2)^2 - 4(3)(4))}{4(3)}\right) = \left(-\frac{1}{3}, \frac{44}{12}\right) \text{ ou } \left(-\frac{1}{3}, \frac{11}{3}\right).$$

c) O esboço do gráfico:

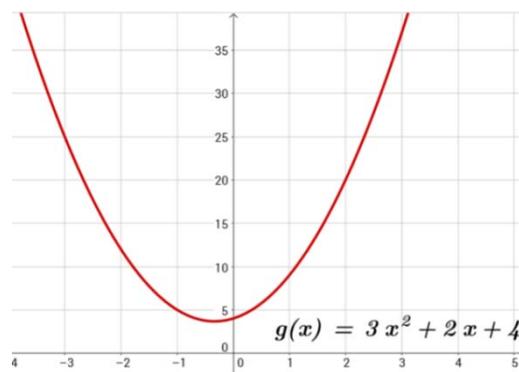


Figura 14 – Gráfico da função $g(x) = 3x^2 + 2x + 4$.

3.3 Sequência didática para construção do aplicativo.

São objetivos desta sequência didática:

- criar um programa que calcule as raízes reais de uma equação do segundo grau e fazê-lo funcionar em dispositivos móveis como smartphones e tablets;
- propiciar ao estudante meios de fixar as definições mais usadas nos processos de resolução de problemas envolvendo equação do segundo grau;
- conhecer aplicações dos conectivos *se, ou, então, senão, se e somente se, falso, verdadeiro*, entre outros;

Consideram-se competências (conhecimentos e habilidades) desta sequência didática:

- dominar as propriedades de potenciação e radiciação;
- reconhecer uma equação do segundo grau;
- identificar os coeficientes de uma equação do segundo grau;
- determinar as raízes de uma equação do segundo grau pelo método de Bhaskara;
- perceber a existência de raízes reais da equação a partir da análise do valor do discriminante Δ ;

Os recursos necessários para a execução desta sequência didática são:

- laboratório de informática;
- acesso à internet;
- rede sem fio (wireless);
- smartphones ou tablets que utilizem o sistema operacional Android;
- cabo de conexão tipo USB;
- aplicativo *MIT AI2 Companion* instalado nos smartphones ou tablets;
- conta de usuário no Google;
- lápis, caderno, borracha para anotações.

3.3.1 Esboço da estrutura do aplicativo e escolha dos componentes da tela.

Inicie a aula conversando com os alunos à respeito do modo que eles vêem o futuro aplicativo e as possíveis telas que ele poderá ter. Crie com o grupo um roteiro através de desenhos e esquemas que ajude a idealizar como o aplicativo funcionará. O esboço auxiliará durante a montagem dos componentes de interação do usuário com o aplicativo bem como na criação dos blocos relacionados à parte lógica do programa. A ideia é ter um programa simples e eficiente.

Comece desenhando no quadro, um retângulo que simule a tela de um smartphone. Em seguida, docente e alunos deverão discutir quais elementos serão necessários para compor a tela do aplicativo. Basicamente, deverá haver locais para serem digitados os coeficientes numéricos e locais para serem exibidos as possíveis soluções (raízes). A discussão deve concluir também que precisa ser adicionado um componente que iniciará o processo de cálculo no aplicativo. Este elemento poderá ser um botão.

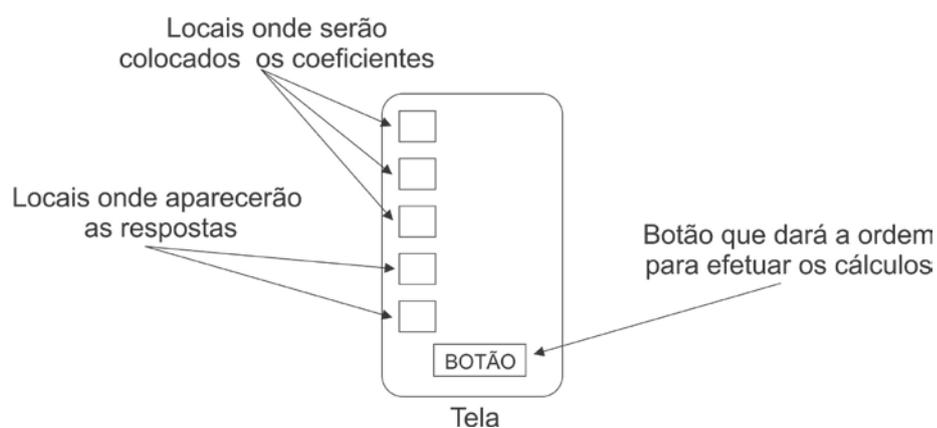


Figura 15 – Discriminação das funções de cada componente.

Outros componentes que podem ser incorporados na tela do programa são as etiquetas que darão dicas onde devem ser inseridos os coeficientes. Estas etiquetas podem ser os símbolos matemáticos e a variável da função quadrática.

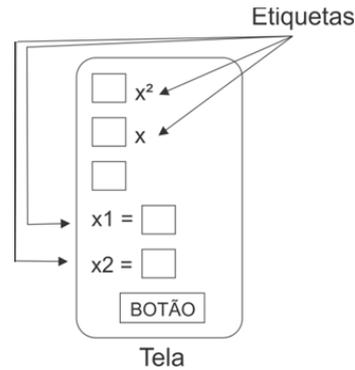


Figura 16 – Etiquetas auxiliam o uso do aplicativo pelo usuário.

Vale ressaltar que a aparência do aplicativo será melhorada posteriormente, devendo apenas preocupar-se com a funcionalidade do programa.

Com um novo retângulo no quadro (Figura 17), mantendo o primeiro, o professor ajudará os alunos a prepararem a sequência de cálculos que o programa fará para resolver a função. Comece escrevendo a fórmula de Bhaskara. Naturalmente não é assim que o programa entende, portanto reescreva em forma de expressão numérica. Seguindo, relacione os componentes desenhados no primeiro retângulo com os termos da fórmula de Bhaskara (Figura 18).

$$\text{raiz} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

↓

$$[-b \pm (b^2 - 4ac)^{1/2}] / (2a)$$

Blocos

Figura 17 – Escrita linear da fórmula de Bhaskara.

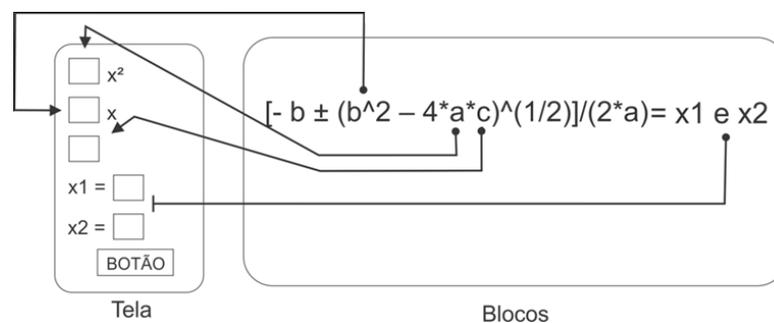


Figura 18 – Relação entre as caixas de textos e os coeficientes da função quadrática.

Oriente os alunos descreverem no caderno a sequência que o aplicativo resolverá a fórmula de Bhaskara. A divisão da expressão numérica em blocos auxilia no raciocínio.

$$[-b \pm (b^2 - 4ac)^{1/2}] / (2a)$$

Blocos

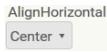
Figura 19 – Sequência de passos a ser realizado pelo aplicativo para solucionar a função.

Pelo esquema anterior, percebe-se que a ordem de resolução será de dentro para fora, da seguinte maneira:

- (1) Simultaneamente será calculado $A = b^2$ e $B = 4ac$;
- (2) Em seguida subtrairá o valor de B em A: $C = A - B$;
- (3) Do valor de C, será extraído a raiz quadrada: $D = C^{(1/2)}$;
- (4) Após, será somado ou subtraído de $(-b)$ o valor de D: $E = (-b) \pm D$;
- (5) O valor do denominador será determinado: $F = 2a$;
- (6) Finalmente, a divisão do numerador E pelo denominador F será feito: $G = E / F$.

3.3.2 Início da construção do aplicativo com a montagem da tela.

Agora, os alunos construirão os aplicativos na plataforma, iniciando com a inclusão da tela do programa, as caixas de textos, as etiquetas e o botão. Em seguida montarão os blocos que repetirão o procedimento ilustrado na Figura 19.

Comece indo em *Projects*²⁰ (Projetos), e em seguida *Start new project* (Começar novo projeto) e escolha um nome para o trabalho. Após, selecione o objeto “*Screen1*” listado na coluna *Components* (Componentes), e edite no menu *Properties* (Propriedades) alterando a opção *Center* (Centro) em *AlignHorizontal* (Alinhamento Horizontal) . Em *AppName* (Nome do Aplicativo) coloque o nome que preferir. Neste tutorial foi usado “E-

²⁰ Havendo dúvidas, consulte o ANEXO B – **Conhecendo a plataforma**, onde está detalhado o processo para iniciar um novo projeto.

quação do 2º Grau”. Esta frase é a que aparecerá abaixo do ícone do aplicativo quando estiver instalado no dispositivo. Marque a opção *Scrollable* (Rolar) **Scrollable**, isto permite a tela rolar à medida que mais componentes são acrescentados. Em *Title* (Título) coloque um título para esta tela do aplicativo. No exemplo foi colocado “Equação do 2º Grau” também.

Dando prosseguimento, adicione três caixas de textos **TextBox**, oito etiquetas **Label** e um botão **Button**, conforme Figura 20.

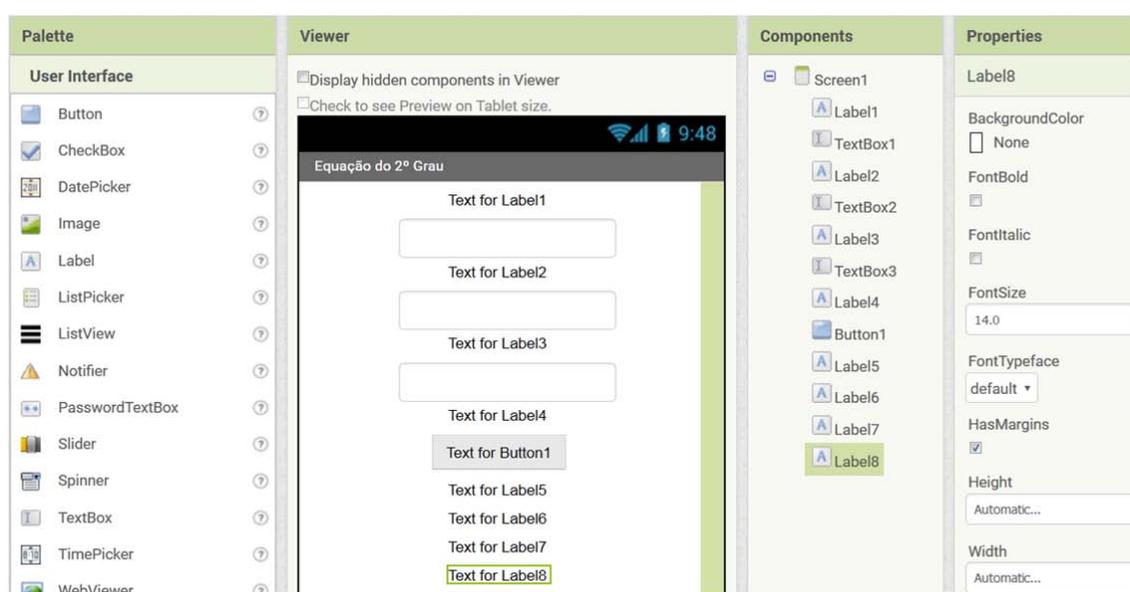


Figura 20 – Componentes adicionados na tela do aplicativo.

Em seguida, clique sobre cada objeto listado em *Components* e no menu *Properties* faça as seguintes modificações:

- (1) Selecione cada caixa de texto e renomeie²¹, nesta ordem, para : “A”, “B” e “C”. Aproveite e retire a frase “*Hint for TextBox*” em *Hint* (Sugestão), pois ela é exibida dentro da caixa de texto quando o aplicativo é executado, atrapalhando a digitação, e marque a opção *NumbersOnly* (Somente Números) para dar agilidade na digitação dos valores nas caixas de textos.
- (2) Nas etiquetas, altere os nomes para: “etiqueta_A”, “etiqueta_B”, “etiqueta_C”, “etiqueta_igual”, “etiqueta_raiz1”, “raiz_1”, “etiqueta_raiz2” e “raiz_2”;

²¹ Para renomear um componente, selecione-o no menu *Components* e clique no botão *Rename* (Renomear) **Rename**, que aparecerá uma caixa de diálogo solicitando a nova nomenclatura.

- (3) Nas quatro primeiras etiquetas, nesta mesma ordem, vá na propriedade *Text*, e substitua por: “(”, “)X² + (”, “)X + (”, “) = 0”. Para “etiqueta_raiz1” e “etiqueta_raiz2” coloque “Raiz 1:” e “Raiz 2:”. Já nas etiquetas “raiz_1” e “raiz_2” deixe sem texto;
- (4) No componente botão, mude o nome para “botao_calcular” e edite o *Text* substituindo por “CALCULAR”.

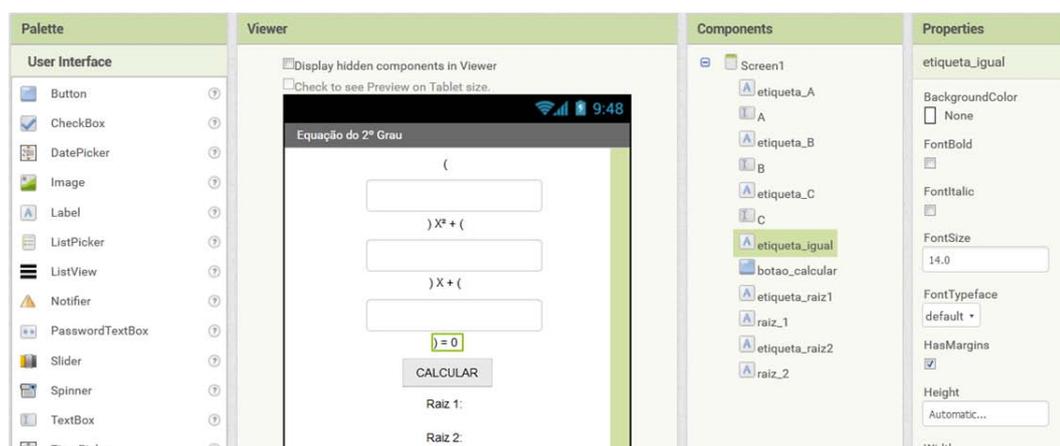


Figura 21 – Componentes adicionados na tela do aplicativo após as mudanças nas suas propriedades.

3.3.3 Montagem dos blocos responsáveis pelos cálculos usando o método de Bhaskara.

Prosseguindo, vá para o ambiente de montagem dos blocos, clicando no botão *Blocks* no canto superior direito. O propósito é recriar o diagrama desenhado no quadro (Figura 19) usando os blocos de programação. Pelos registros, comece montando a parte (**b²**). Então vá no menu *Math* (Matemática) , na lateral esquerda, escolha o bloco azul correspondente a potenciação , arraste e solte-o dentro da tela *Viewer* (Visualizador).

Em seguida no menu *Blocks*, selecione o componente “B” e procure pelo bloco verde correspondente a *Text*  e o bloco azul correspondente a número  (em *Math*). Substitua o valor de 0 por 2 e monte o seguinte conjunto: .

Para montar o cálculo da multiplicação do número 4 pelos componentes “A” e “C”, adicione o bloco azul da operação de multiplicação  (menu *Math*), os blocos verdes referentes a *Text* dos componentes “A” e “C” além de outro bloco correspondente a número,

atribuindo o valor 4.



Figura 22 – Itens necessários para o cálculo de $(4*a*c)$.

No bloco azul da multiplicação existe um ícone no canto superior esquerdo. Ao clicá-lo aparecerá a opção de incluir ou retirar fatores de multiplicação. Para aumentar basta arrastar o bloco *number* (número) para o lado direito.

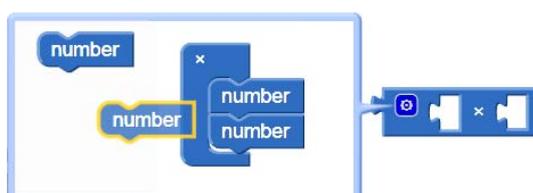


Figura 23 – Procedimento para adicionar termos no bloco da multiplicação.

A montagem da composição fica: 

A operação $(b^2 - 4*a*c)$ será com inserção dos grupos feitos anteriormente num bloco de subtração .

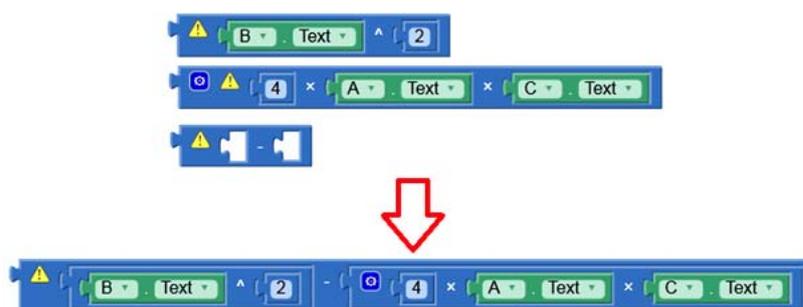


Figura 24 – Bloco responsável pelo cálculo de $(b^2 - 4*a*c)$.

A etapa seguinte é montar o conjunto que calculará a raiz quadrada do resultado obtido anteriormente em $b^2 - 4*a*c$ (Figura 24). Para isto, adicione outro bloco azul de potenciação. Aqui há duas opções: o expoente pode ser o número decimal 0.5 ou pode usar o bloco da divisão com os números 1 e 2, conforme ilustra a imagem a seguir:

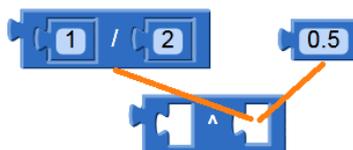


Figura 25 – Comparação das formas de representar potência fracionária.

Fica a critério do professor escolher a opção. Porém é necessário ressaltar que ao usar a notação decimal, os resultados serão mostrados com poucas casas decimais, portanto tal estratégia ficaria comprometida se deparasse com uma fração do tipo $5/6$ ou $7/8$. Portanto este tutorial seguirá usando a primeira opção resultando na composição da Figura 26.



Figura 26 – Bloco responsável pelo cálculo de $(b^2 - 4*a*c)^{(1/2)}$.

Para representar o termo $(-b)$ use a operação de subtração com o número zero. Veja:

Como é possível encontrar até duas raízes distintas, precisa-se de dois blocos e dois dos blocos da Figura 26. A fim de evitar repetir todos os passos anteriores, faça cópia de cada um. Após, adicione um bloco azul de adição e outro de subtração.

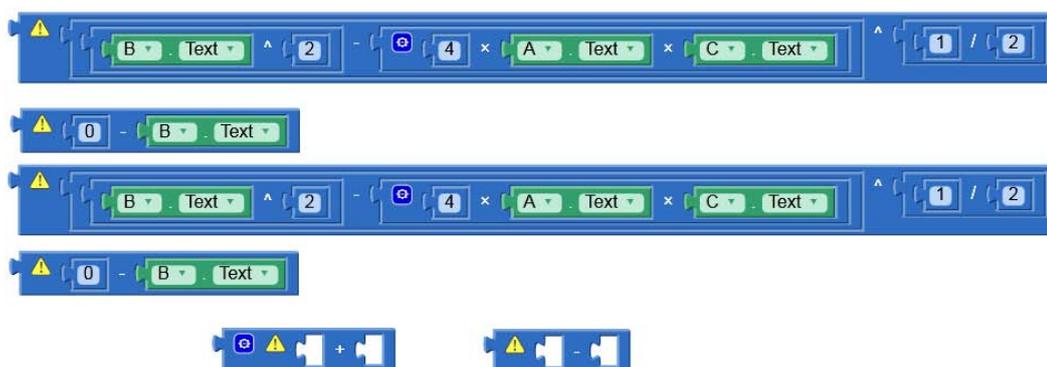


Figura 27 – Blocos necessários pelos cálculos de cada uma das raízes.

Divida as operações nos blocos de adição e subtração como ilustrado na Figura 28.

O produto $(2*a)$ do denominador é formado pelo bloco de multiplicação, o bloco numérico para o 2 e o item verde de texto referente ao componente “A”: Faça dois deste bloco.



Figura 28 – Composição dos numeradores de cada raiz.

Finalize criando a fração através de dois blocos de divisão e distribuindo os conjuntos construídos, ficando da seguinte forma:

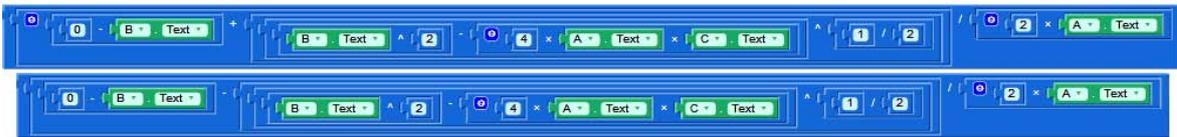


Figura 29 – Representação completa da fórmula de Bhaskara em blocos.

3.3.4 Montagem dos blocos responsáveis pela análise do sinal do discriminante Δ .

Como o objetivo é encontrar apenas as raízes reais, deve-se condicionar os cálculos apenas a equações com discriminante positivo ou igual a zero. Para isso, neste passo, será visto um pouco de programação básica.

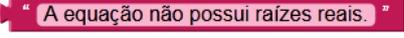
Para o aplicativo efetuar a análise do discriminante Δ , serão usados os conectivos lógicos *se*, *então* e *senão* (*If*, *then*, *else*) da seguinte forma:

*Se $\Delta < 0$, então imprimir "a equação não possui raízes REAIS",
senão, calcular as raízes.*

Comece então providenciando o bloco que calcula o valor do discriminante $\Delta = b^2 - 4ac$. Como foi feito anteriormente, Figura 24, basta apenas fazer uma cópia. A tarefa é verificar se $\Delta < 0$, assim é preciso de um bloco  e um bloco  (em *Math*), substituindo o sinal de “=” por “<”. A montagem ficará assim:



Figura 30 – Conjunto responsável pela análise do sinal de Δ .

Se o discriminante for negativo, não haverá raízes reais, e o aplicativo deverá imprimir esse aviso para o usuário. Esta informação poderá ser transmitida na tela através de uma frase. Uma solução é adicionar o bloco  disponível no menu *Text* (Texto)  *Text*, e colocar a frase “A equação não possui raízes reais.”: . Esta frase poderá aparecer aonde o usuário espera as respostas das raízes, nas etiquetas “raiz_1” e “raiz_2”. Deste modo, procure o componente “raiz_1” no menu *Blocks*, e selecionando-o aparecerá um submenu onde estará o bloco . Encaixe-o com bloco lilás que foi construído. Faça o mesmo com o componente “raiz_2”. Veja:

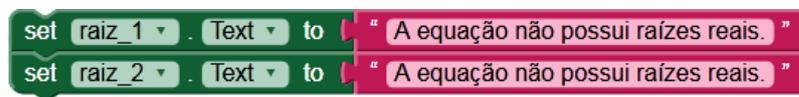
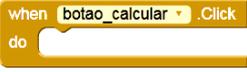


Figura 31 – Conjunto responsável pela exibição da mensagem nas etiquetas das respostas.

3.3.5 Conclusão da montagem dos blocos.

Agora atribua ao botão a função para dar a ordem de início dos cálculos. Selecione o componente “botao_calcular” e escolha a função *Click* . No item *Control* (Controle), escolha o bloco *if/then* (se/então) , encaixando-o dentro do bloco do botão.

O aplicativo deverá cumprir as seguintes instruções: Se $\Delta < 0$, imprima que a função não possui raízes reais, caso contrário, calcule as raízes. Sendo assim, em frente à palavra *if* do bloco marrom, coloque o bloco da Figura 30, responsável pela análise do sinal de Δ . Uma vez verdadeira a comparação, então (*then*) retornará a mensagem de “A equação não possui raízes reais.”. Assim acople no espaço *then* os blocos ilustrados na Figura 31. Entretanto (se não = *else*), acaso o resultado da verificação seja positiva, $\Delta \geq 0$, o programa prosseguirá com os cálculos, conforme montagem da Figura 29. Veja o resultado:

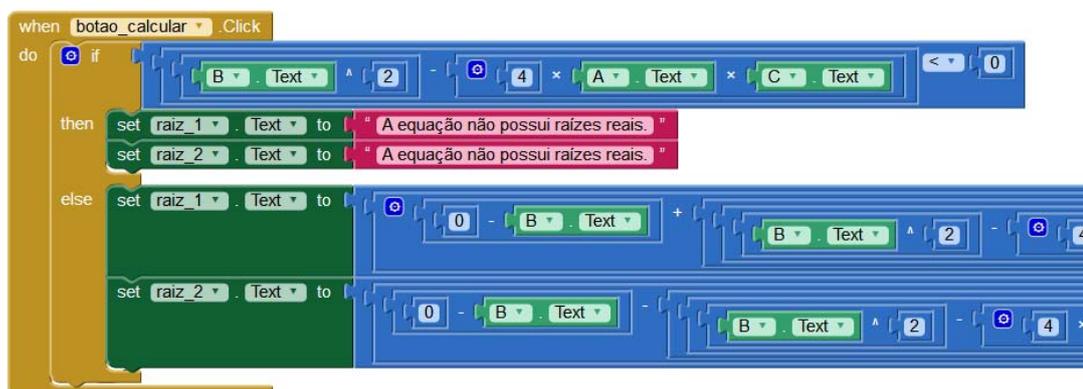


Figura 32 – Composição final da parte lógica do aplicativo.

Peça aos alunos para testarem²² os aplicativos com os exemplos abaixo para averiguar se há algum problema, corrigindo o que for necessário. Normalmente os problemas são oriundos de equívocos durante as montagens dos blocos, como a troca dos componentes, ou a escolha da função errada do item. Por exemplo, ao invés de ter escolhido função *Text* do componente “a11”  pode ter sido escolhido a função  (Tamanho da Fonte). O professor pode delegar estudantes com mais habilidades para acompanhar os demais auxiliando na monitoria do projeto.

Exemplos de equações para testes:

Exemplo 3.2: $x^2 + 3x + 2 = 0$, com $x_1 = -1$ e $x_2 = -2$.

Exemplo 3.3: $x^2 - x - 12 = 0$, com $x_1 = 4$ e $x_2 = -3$.

Exemplo 3.4: $2x^2 - 4x + 3/2 = 0$, com $x_1 = 3/2$ e $x_2 = 1/2$.

Exemplo 3.5: $x^2 + 1 = 0$. Não possui raízes reais.

Exemplo 3.6: $x^2 + 2x + 3 = 0$. Não possui raízes reais.

²² Para testar o aplicativo, consulte o ANEXO A – O que instalar, onde está descrito detalhadamente as três formas de testes.

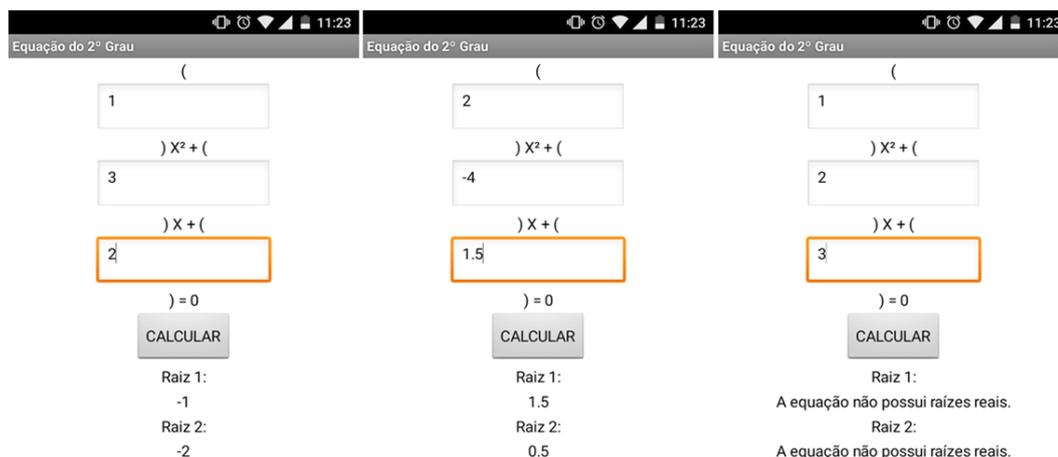


Figura 33 – Capturas de telas do smartphone exibindo o aplicativo solucionando os exemplos propostos.

3.3.6 Finalização do projeto com modificações nas propriedades dos componentes.

Veja que o projeto desenvolvido já consegue atingir os objetivos propostos, que é determinar as raízes de equações do 2º grau. Porém é necessário efetuar ajustes nas propriedades de alguns componentes para que a aparência do aplicativo fique mais atraente. Existem inúmeras formas fazê-lo. Abaixo segue a sequência de edições usadas no aplicativo deste tutorial, que podem inspirar a deixar o seu projeto personalizado.

Retorne à tela *Designer* (Desenhador), clicando no botão  no canto superior direito para iniciar as edições, começando com a inclusão de uma etiqueta de nome “etiqueta_titulo”, marcando a opção *FontBold* (Negrito), alterando o *FontSize* (Tamanho da Fonte) para 20 e colocando a frase “INSIRA OS COEFICIENTES NAS CAIXAS DE TEXTOS” em *Text*. Altere também a posição em *TextAlignment* (Alinhamento do texto) para *Center* e a cor em *TextColor* (Cor do Texto).

No menu da lateral esquerda, *Layout*, escolha o componente *HorizontalArrangement* (Arranjos na Horizontal)  e coloque após a “etiqueta_titulo” dando-lhe o nome de “caixa_equacao” e altere a propriedade *AlignHorizontal* e *AlignVertical* para *Center*, e *Width* (Largura) para 90 em *percent* (por cento).

Depois arraste as caixas de textos “A”, “B”, e “C”, e as etiquetas “etiqueta_A”, “eti-

queta_B”, “etiqueta_C”, “etiqueta_igual”, para dentro da “caixa_equacao” mantendo a ordem. Se acaso não couber, vá modificando a propriedade *Width* das caixas de textos colocando 17 em *percent*. Nas caixas de textos e nas etiquetas, marque *FontBold* e *TextAlignment* para *Center*.

Coloque outra etiqueta, “etiqueta_aviso” sob “caixa_equacao”, e nas propriedades marque *FontBold*, *FontItalic* (Itálico), mude o *FontSize* para 12, escreva em *Text* a frase “Coloque zero nos coeficientes que faltarem.”, alterando também a posição em *TextAlignment* para *Center* e a cor em *TextColor*.

Modifique as características do botão em *BackgroundColor* (Cor de Fundo), marque *FontBold*, mude o *FontSize* para 20, escolha alguma opção em *Shape* (Forma) e a posição do *TextAlignment* para *Center*.

Adicione uma *TableArrangement* (Arranjos em Tabela)  *TableArrangement* depois do botão, de nome “caixa_respostas” e altere as propriedades deixando 2 em *Columns* (Colunas) e 2 em *Rows* (Linhas), edite o *Width* digitando 90 em *percent*. Nela coloque as etiquetas “etiqueta_raiz1”, “raiz_1”, “etiqueta_raiz2” e “raiz_2”.

Seguindo, nas etiquetas, marque *FontBold*, altere o *FontSize* para 20 apenas na “etiqueta_raiz1” e “etiqueta_raiz2”. Altere o *TextAlignment* para *Center* e deixe com a cor azul a “etiqueta_raiz1” e “etiqueta_raiz2” e vermelho em “raiz_1” e “raiz_2”.

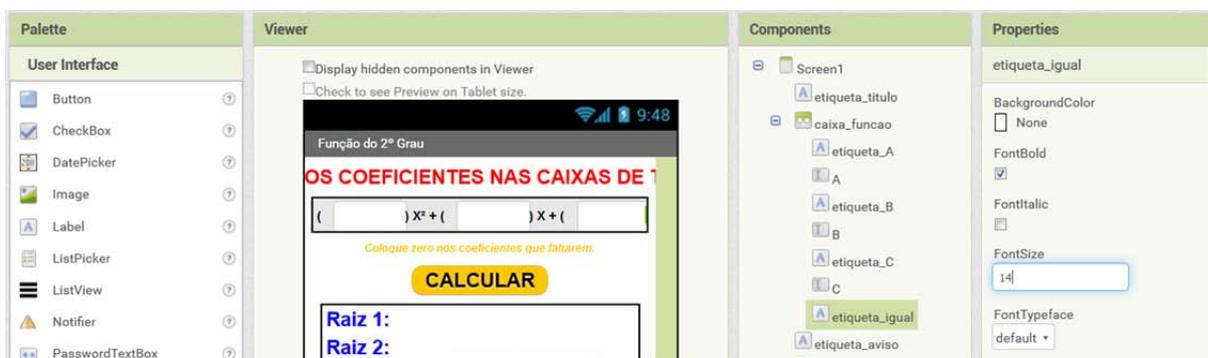


Figura 34 – Ilustração das últimas modificações realizadas nos componentes.

Escolha uma boa figura na internet e adicione através do botão *UploadFile*  no menu *Média* (Mídia). Depois selecione o componente “*Screen1*”, e vá e *BackgroundImage*

(Imagem de Fundo), selecionando o arquivo que foi inserido.

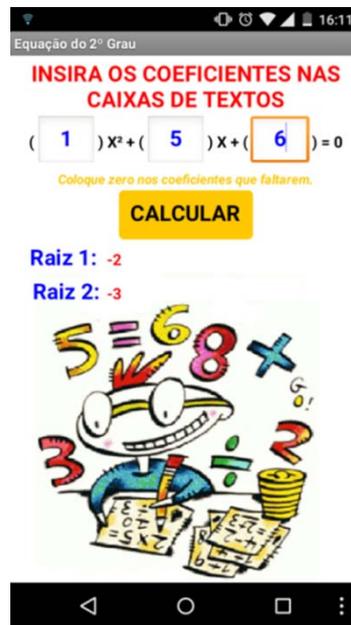


Figura 35 – Captura de tela do smartphone exibindo as modificações feitas no aplicativo.

Outro incremento interessante para ser aplicado é a inserção de um botão que limpe todas as caixas de textos e etiquetas que exibem respostas a cada vez que for incluir novos dados. Assim, adicione depois de “caixa_respostas” um novo botão chamado “botao_limpar” com o *Text* “LIMPAR”, aplicando as mesmas modificações feitas no outro botão. Em seguida, vá para o ambiente *Blocks* selecione a função  do “botao_limpar”, e também o bloco *set* de cada componente caixa de texto e etiqueta comentada anteriormente, agregando ao bloco  disponível no menu *Control*, conforme a composição ilustrada a seguir.

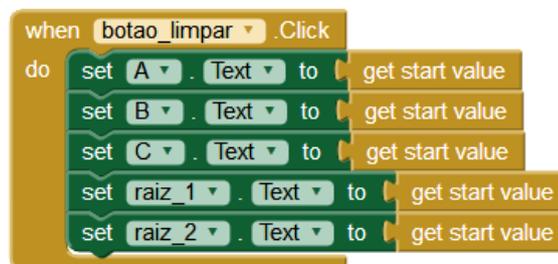


Figura 36 – Conjunto responsável por limpar todas as caixas de textos e etiquetas.

3.3.7 Desafio.

Para concluir esta sequência didática, proponha à turma uma tarefa em forma de desafio, que pode ser cumprida na própria escola ou como trabalho extra-classe, onde os alunos deverão usar os novos conhecimentos adquiridos.

Solicite a eles que pesquisem em livros e na internet métodos para determinar as raízes reais de equações do terceiro grau, e após a pesquisa comecem um novo projeto e construam o aplicativo que faça os referidos cálculos.

4 MATRIZES

Apesar do conteúdo de matriz ser intrínseco em algumas atividades do cotidiano, seu processo de ensino-aprendizagem nem sempre é agradável, conduzindo a altos níveis de rejeição em face do tema, prejudicando na assimilação do conhecimento e favorecendo um baixo rendimento na disciplina. Mesmo não sendo foco deste trabalho, algumas conjecturas podem ser feitas sobre isso, como a extensibilidade do conteúdo, a quantidade de definições e propriedades envolvendo as operações entre matrizes, aos cálculos monótonos para resolver problemas, além da abordagem pouco criativa do docente.

Campeonato Brasileiro de Futebol de 2015 - Série A									
Classificação									
#	Equipe	J	VIT	E	DER	GP	GC	SG	PTS
1	 Corinthians	38	24	9	5	71	31	40	81
2	 Atlético	38	21	6	11	65	47	18	69
3	 Grêmio	38	20	8	10	52	32	20	68
4	 São Paulo	38	18	8	12	53	47	6	62
5	 Internacional	38	17	9	12	39	38	1	60

Figura 37 – Exemplo de matriz aplicada no cotidiano. Fonte: Google.

Segue então, que este capítulo entrega uma sequência didática que usará a plataforma do MIT App Inventor 2 como ferramenta de fixação de algumas das principais definições relacionadas ao conteúdo de matrizes, através da construção de um aplicativo. Além utilizar ferramentas — tecnologia e dispositivos móveis — de grande interesse dos estudantes, garante uma drástica redução de tempo, poupando alunos e professores da resolução de cansativos cálculos, permitindo um processo de aprendizagem voltado para a aplicações do conteúdo.

4.1 Principais resultados.

Aqui serão listados os principais resultados sobre matrizes que nortearão esta sequência didática. As definições, teoremas, proposições e alguns exemplos desta seção foram extraídos dos livros BOLDRINI (1980), CALLIOLI (1990), HEFEZ (2012) e IEZZI (1977).

Definição 4.1: Sejam $m \geq 1$ e $n \geq 1$ dois números naturais e não nulos. Chama-se matriz real $m \times n$ toda tabela formada por números reais, distribuídos em m linhas e n colunas:

$$M = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

Cada elemento que compõe a matriz é chamado de termo, representado por a_{ij} , onde i representa a linha e j a coluna. É possível ainda representar a matriz como $M = (a_{ij})_{m \times n}$ ou simplesmente $M = (a_{ij})$.

O conjunto das matrizes ordem $m \times n$ encontradas em R será denotado por $M_{m \times n}(R)$.

Definição 4.2: Uma matriz A de ordem $n \times n$ denomina-se matriz quadrada. Neste caso, a matriz $A = (a_{ij})_{n \times n}$ diz-se que é uma matriz de ordem n , e denota-se por $A = (a_{ij})_n$. As matrizes quadradas de ordem n encontradas em R serão denotadas por $M_n(R)$.

Definição 4.3: A matriz quadrada $A = (a_{ij})_n$, onde $a_{ij} = 0$, para todo $i, j = 1, \dots, n$ com $i \neq j$, denomina-se matriz diagonal.

Definição 4.4: A matriz quadrada $I_n = (a_{ij})_n$, onde $a_{ij} = \begin{cases} 1, & \text{se } i = j; \\ 0, & \text{se } i \neq j; \end{cases}$ denomina-se matriz identidade.

Definição 4.5: A matriz de ordem $m \times n$, $O_{m \times n} = (a_{ij})_{m \times n}$, onde $a_{ij} = 0$, para todo $1 \leq i \leq m$ e para todo $1 \leq j \leq n$, denomina-se matriz nula.

Definição 4.6: Dada uma matriz $A = (a_{ij})_{m \times n}$, define-se a matriz oposta de A , denotada por $-A$, como a matriz $-A = (-a_{ij})_{m \times n}$.

Definição 4.7: Dada uma matriz $A = (a_{ij})_{m \times n}$, define-se a matriz transposta de A , denotada por A^t , como a matriz $A^t = (b_{ij})_{n \times m}$, onde $b_{ij} = a_{ji}$, para todo $1 \leq i \leq n$ e para todo $1 \leq j \leq m$.

Definição 4.8: Dadas duas matrizes $A = (a_{ij})_{m \times n}$ e $B = (b_{ij})_{m \times n}$, de mesma ordem, denota-se $A + B$ a matriz $C = (c_{ij})_{m \times n}$, tal que $c_{ij} = a_{ij} + b_{ij}$, para todo $1 \leq i \leq m$ e $1 \leq j \leq n$.

Proposição 4.1: Dadas três matrizes A , B e C pertencentes a $M_{m \times n}(R)$, a operação de adição de matrizes, satisfaz as seguintes propriedades²³:

- (1) $A + B = B + A$ (comutativa);
- (2) $(A+B) + C = A + (B+C)$ (associativa);
- (3) $A + O_{m \times n} = A$ (existência do elemento neutro aditivo, onde $O_{m \times n}$ denota a matriz nula);
- (4) $A + (-A) = O_{m \times n}$ (existência do elemento oposto, onde $O_{m \times n}$ denota a matriz nula).

Definição 4.9: Dada duas matrizes $A = (a_{ij})_{m \times n}$ e $B = (b_{jk})_{n \times p}$, define-se como o produto de A por B , denotado por AB , como a matriz $C = (c_{ik})_{m \times p}$ tal que:

$$C = (c_{ik}) = \sum_{j=1}^n a_{ij} b_{jk} = a_{i1} b_{1k} + \dots + a_{in} b_{nk}, \text{ para todo } 1 \leq i \leq m \text{ e } 1 \leq k \leq p.$$

Proposição 4.2: Sejam $A \in M_{m \times n}(R)$, $B \in M_{n \times p}(R)$, $C \in M_{p \times q}(R)$ e $D \in M_{n \times p}(R)$, a multiplicação de matrizes, satisfaz as seguintes propriedades²⁴:

- (1) $A(BC) = (AB)C$ (associatividade da multiplicação);
- (2) $AI_n = A$ e $I_m A = A$, onde I_n denota a matriz identidade de ordem n (elemento neutro);
- (3) $A(B+D) = AB + AD$ (distributividade à esquerda da multiplicação em relação à adição);
- (4) $(B+D)A = BA + DA$ (distributividade à direita da multiplicação em relação à adição);
- (5) $A_{m \times n} O_{n \times p} = O_{m \times p}$ e $O_{p \times m} A_{m \times n} = O_{p \times n}$, onde $O_{p \times n}$ denota a matriz nula.

²³ As demonstrações destas propriedades podem ser encontradas em (BOLDRINI, 1980, pg. 11).

²⁴ As demonstrações destas propriedades podem ser encontradas em (IEZZI, 1977), pg. 50.

Definição 4.10: Diz-se que uma matriz A , pertencente a $M_n(\mathbb{R})$, é inversível, se existe uma matriz B , pertencente a M_n , tal que $AB = BA = I_n$. A matriz B é chamada matriz inversa de A e denota-se por $B = A^{-1}$.

Definição 4.11: Determinante é uma função matricial que associa a cada matriz quadrada um escalar; ele transforma essa matriz em um número real. A notação para determinantes é: $\det(M) = k$, para $k \in \mathbb{R}$ e $M \in M_n$.

a) Dada uma matriz quadrada de primeira ordem, $A=[a_{11}]$, o seu determinante é o número real a_{11} , isto é, $\det(A) = a_{11}$.

b) Dada uma matriz quadrada de segunda ordem o seu determinante é o número real obtido pela subtração do produto da diagonal secundária do produto da diagonal principal.

c) Para calcular o determinante de matrizes de ordem 3, pode-se utilizar a regra de Sarrus, que consiste em repetir as duas primeiras colunas ao lado da terceira, em seguida encontrar a soma do produto dos elementos da diagonal principal com os dois produtos obtidos com os elementos das paralelas a essa diagonal, e subtrair da soma do produto dos elementos da diagonal secundária com os dois produtos obtidos com os elementos das paralelas a essa diagonal, conforme ilustra o desenho abaixo.

$$\det(B) = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{11} & b_{12} \\ b_{21} & b_{22} & b_{23} & b_{21} & b_{22} \\ b_{31} & b_{32} & b_{33} & b_{31} & b_{32} \end{bmatrix} = \begin{bmatrix} (b_{11} \times b_{22} \times b_{33}) \\ + \\ (b_{12} \times b_{23} \times b_{31}) \\ + \\ (b_{13} \times b_{21} \times b_{32}) \end{bmatrix} - \begin{bmatrix} (b_{13} \times b_{22} \times b_{31}) \\ + \\ (b_{11} \times b_{23} \times b_{32}) \\ + \\ (b_{12} \times b_{21} \times b_{33}) \end{bmatrix}$$

Figura 38 – Visualização da aplicação da Regra de Sarrus.

Este trabalho será limitado ao cálculo do determinante de matrizes até ordem 3. Para matrizes de ordem superior, as técnicas para o cálculo dos determinantes podem ser encontradas, por exemplo, em CALLIOLI (1990, pg. 203) e IEZZI (1977, pg. 67).

4.2 Exemplos.

Exemplo 4.1: A matriz $A = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$ é uma matriz diagonal.

Exemplo 4.2: A matriz $I_5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ é uma matriz identidade.

Exemplo 4.3: A matriz $O_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ é uma matriz nula.

Exemplo 4.4: A matriz transposta da matriz quadrada $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 1 & 6 \\ 0 & 2 & -3 \end{bmatrix}$ é matriz $A^t =$

$$\begin{bmatrix} 1 & 4 & 0 \\ 2 & 1 & 2 \\ 3 & 6 & -3 \end{bmatrix}.$$

Exemplo 4.5: A soma das matrizes $M = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 9 & 11 \\ 13 & 15 & 17 \end{bmatrix}$ e $N = \begin{bmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \\ 14 & 16 & 18 \end{bmatrix}$ é dada por

$$M + N = \begin{bmatrix} 1+2 & 3+4 & 5+6 \\ 7+8 & 9+10 & 11+12 \\ 13+14 & 15+16 & 17+18 \end{bmatrix} = \begin{bmatrix} 3 & 7 & 11 \\ 15 & 19 & 23 \\ 27 & 31 & 35 \end{bmatrix}.$$

Exemplo 4.6: O produto das as matrizes $A = \begin{bmatrix} 1 & 3 & -2 \\ 4 & -5 & 7 \\ 6 & 8 & -9 \end{bmatrix}$ e $B = \begin{bmatrix} 4 & 7 & 3 \\ 0 & 9 & 6 \\ 5 & 1 & 8 \end{bmatrix}$, é $AB =$

$$\begin{bmatrix} (1 \times 4) + (3 \times 0) + (-2 \times 5) & (1 \times 7) + (3 \times 9) + (-2 \times 1) & (1 \times 3) + (3 \times 6) + (-2 \times 8) \\ (4 \times 4) + (-5 \times 0) + (7 \times 5) & (4 \times 7) + (-5 \times 9) + (7 \times 1) & (4 \times 3) + (-5 \times 6) + (7 \times 8) \\ (6 \times 4) + (8 \times 0) + (-9 \times 5) & (6 \times 7) + (8 \times 9) + (-9 \times 1) & (6 \times 3) + (8 \times 6) + (-9 \times 8) \end{bmatrix}$$

$$= \begin{bmatrix} -6 & 32 & 5 \\ 51 & -10 & 38 \\ -21 & 105 & 6 \end{bmatrix}.$$

Exemplo 4.7: A inversa da matriz $C = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix}$, é a matriz $D = \begin{bmatrix} 0 & 1 \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix}$.

Seja $D = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ a inversa de C , logo $\begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, e portanto fica:

$$\begin{bmatrix} 1a + 2c & 1b + 2d \\ 1a + 0c & 1b + 0d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \text{ onde } \begin{cases} a + 2c = 1 \\ a = 0 \end{cases} \text{ e } \begin{cases} b + 2d = 0 \\ b = 1 \end{cases}. \text{ Segue daí que } a = 0, b = 1, c = \frac{1}{2} \text{ e } d = -\frac{1}{2} \text{ e então, a matriz } D = \begin{bmatrix} 0 & 1 \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix}.$$

Exemplo 4.8: O determinante da matriz $M = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 0 \\ 4 & 2 & 1 \end{bmatrix}$, é igual a -3.

Repetindo a primeira e a segunda coluna ao lado da terceira, some os produtos das três diagonais principais e subtraia o valor da soma dos produtos das três diagonais secundárias:

$$\det(M) = \begin{bmatrix} 1 & 2 & 3 & 1 & 2 \\ 2 & 1 & 0 & 2 & 1 \\ 4 & 2 & 1 & 4 & 2 \end{bmatrix} = [(1 \times 1 \times 1) + (2 \times 0 \times 4) + (3 \times 2 \times 2)] - [(3 \times 1 \times 4) + (1 \times 0 \times 2) + (2 \times 2 \times 1)] = [1 + 0 + 12] - [12 + 0 + 4] = 13 - 16 = -3.$$

4.3 Sequência didática para construção do aplicativo.

São objetivos desta sequência didática:

- criar um programa que calcula soma, produto e determinante de matrizes quadradas de segunda e terceira ordem e fazê-lo funcionar em dispositivos móveis como smartphones e tablets;
- propiciar ao estudante meios de fixar as definições mais usadas nas operações matriciais (soma e multiplicação entre matrizes e determinantes de matrizes);
- conhecer aplicações dos conectivos *se, ou, então, senão, se e somente se, falso, verdadeiro*, entre outros.

Consideram-se competências (conhecimentos e habilidades) desta sequência didática:

- identificar a ordem das matrizes;

- dominar as propriedades matrizes;
- saber somar, subtrair e multiplicar matrizes;
- saber calcular determinantes de matrizes de ordem 2 e 3.

Os recursos necessários para a execução desta sequência didática são:

- laboratório de informática;
- acesso à internet;
- rede sem fio (wireless);
- smartphones ou tablets que utilizem o sistema operacional Android;
- cabo de conexão tipo USB;
- aplicativo *MIT AI2 Companion* instalado nos smartphones ou tablets;
- conta de usuário no Google;
- lápis, caderno, borracha para anotações.

O conteúdo de matrizes permite escolher inúmeras operações possíveis: somas, produtos, inversas, transpostas, determinantes, além do tamanho das matrizes, 2×2 , 3×3 , 7×5 , e assim por diante. Para a elaboração deste material foram escolhidas as operações de soma, multiplicação e o cálculo de determinantes de matrizes 2×2 e 3×3 , para que o estudante tenha um primeiro contato com a plataforma, bem como pelo fato destas operações serem as mais comuns. Entretanto, se o docente quiser construir o aplicativo com tamanhos e operações diferentes, o procedimento será similar.

4.3.1 Esboço da estrutura do aplicativo e escolha dos componentes de algumas telas.

Inicie a aula conversando com os alunos à respeito do modo que eles vêem o futuro aplicativo e as possíveis telas que ele poderá ter. Crie com o grupo um roteiro através de desenhos e esquemas que ajude a idealizar como o aplicativo funcionará.

O esboço auxiliará durante a montagem dos componentes de interação do usuário com o aplicativo bem como na criação dos blocos relacionados à parte lógica do programa. A ideia é ter um programa simples e eficiente, permitindo uma navegação fácil entre as calculadoras.

Veja na Figura 39 um modelo de arranjo das telas que compõem o aplicativo.

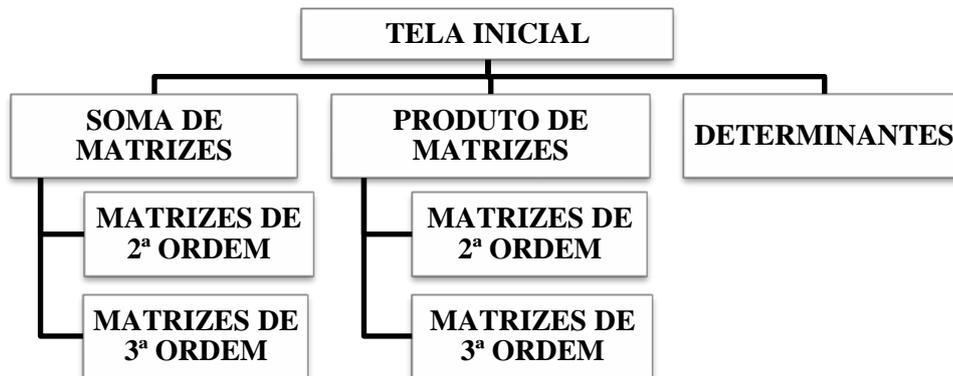


Figura 39 – Modelo de arranjo das múltiplas telas do aplicativo.

Prosseguindo, professor e alunos deverão discutir quais elementos serão necessários para compor as telas do aplicativo. Para ajudar no debate, desenhe no quadro um retângulo que simule a tela de um smartphone (Figura 40-a) e esboce a tela inicial do programa incluindo frases para destacar alguma informação e botões para cumprir a função de abrir novas telas.

Converse ainda como deve ser alguma tela que efetuará os cálculos envolvendo as matrizes, por exemplo, a soma de matrizes. Os alunos devem concluir que a tela deverá ter, pelo menos, os locais para receber os valores da primeira matriz, da segunda matriz, e também o local onde os resultados serão exibidos (Figura 40-b). Lembre-se ainda que “algo” deverá ordenar que os cálculos sejam iniciados. Esta ordem pode partir de um botão também. Vale ressaltar que este modelo pode ser usado em outras calculadoras do aplicativo.

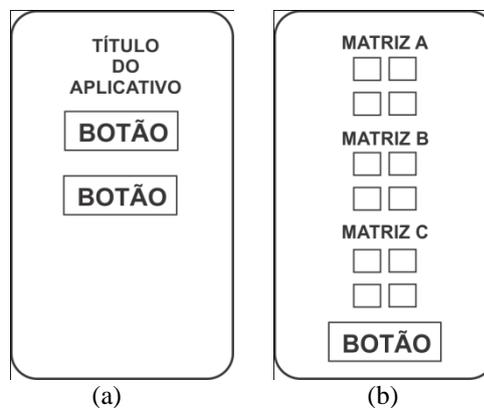
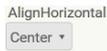


Figura 40 – (a) Esboço da tela inicial do programa. (b) Esboço de tela para operação de soma entre matrizes.

4.3.2 Início da construção do aplicativo com a inclusão das primeiras telas.

A partir do esboço, sugira aos estudantes que iniciem o projeto incluindo algumas telas, e logo após construam as calculadoras de somas, as calculadoras de produtos, e finalmente determinantes pela regra de Sarrus, sempre começando com matrizes de ordem 2 e depois de ordem 3. Assim, devem ser criadas quatro telas: a inicial, que exibirá as opções de calculadoras, a tela das opções de soma, a terceira com as opções de produto, e quarta para determinantes de matrizes.

Portanto, vá em *Projects*²⁵ (Projetos), e em seguida *Start new project* (Começar novo projeto) e escolha um nome para o trabalho. Após, selecione o objeto “*Screen1*” listado na coluna *Components* (Componentes), e edite no menu *Properties* (Propriedades) alterando a opção *Center* (Centro) em *AlignHorizontal* (Alinhamento Horizontal) . Em *AppName* (Nome do Aplicativo)  coloque o nome que preferir. Neste tutorial foi usado “Calculadora de Matrizes”. Esta frase é a que aparecerá abaixo do ícone do aplicativo quando estiver instalado no dispositivo. Marque a opção *Scrollable* (Rolar) , isto permite a tela rolar à medida que mais componentes são acrescentados. Em *Title* (Título)  coloque um título para esta tela do aplicativo. No exemplo foi colocado “Página Inicial”.

Agora, adicione uma etiqueta  *Label* e três botões  *Button*, renomeando²⁶ a etiqueta para “etiqueta_titulo” e os botões para “botao_soma”, “botao_produto” e “botao_determinantes”. Selecione o componente “etiqueta_titulo” e na propriedade *Text* (Texto)  coloque “ESCOLHA UMA CALCULADORA”. Com os componentes dos botões selecionados altere também os itens *Text* para “SOMA DE MATRIZES”, “PRODUTOS DE MATRIZES” e “DETERMINANTES DE MATRIZES” (Figura 41).

²⁵ Havendo dúvidas, consulte o **ANEXO B – Conhecendo a plataforma**, onde está detalhado o processo para iniciar um novo projeto.

²⁶ Para renomear um componente, selecione-o no menu *Components* e clique no botão *Rename* (Renomear) , que aparecerá uma caixa de diálogo solicitando a nova nomenclatura.

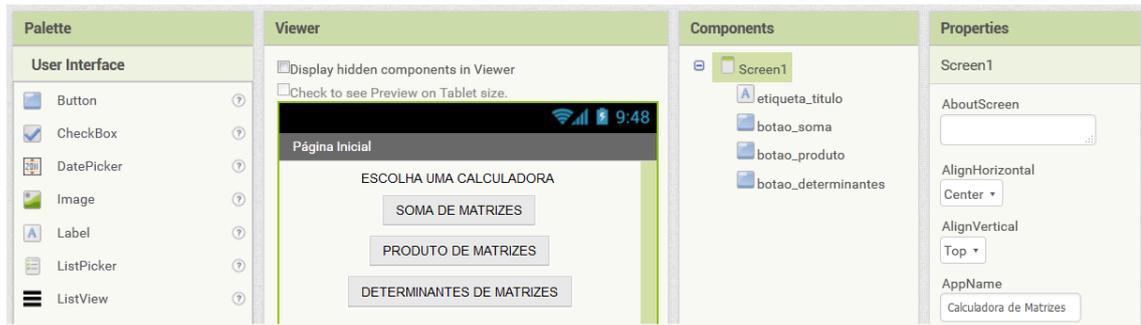


Figura 41 – Tela inicial do aplicativo “Calculadora de Matrizes”.

Para adicionar novas telas (Figura 42), clique no botão *AddScreen* (Adicionar Tela) na parte superior. Inclua três. A primeira chame de “tela_soma”, a segunda de “tela_produto” e a terceira de “tela_determinantes”, para combinar com os botões que foram criados anteriormente.

Assim que terminar, automaticamente o MAI deixará disponível a última tela criada. Para alternar entre as telas, clique no botão do lado esquerdo do *AddScreen* e escolha a tela que deseja trabalhar (Figura 43).



Figura 42 – Demonstração do procedimento para adicionar novas telas.

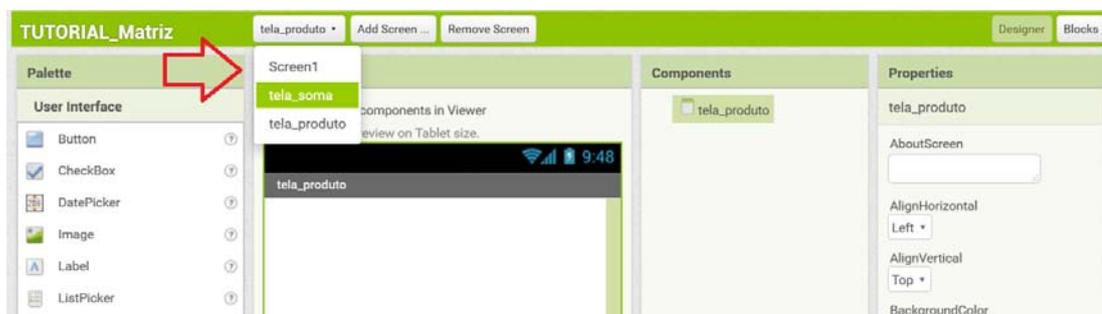


Figura 43 – Demonstração do procedimento para alternar entre as telas.

Com a “tela_soma” aberta, selecione o objeto “tela_soma” listado na coluna *Compo-*

ments, e edite no menu *Properties* alterando a opção *Center* em *AlignHorizontal*. Marque a opção *Scrollable*, e em *Text*, escreva “Soma de Matrizes”. Adicione uma etiqueta e três botões. À etiqueta dê o nome de “etiqueta_titulo”, o primeiro botão será “botao_soma2”, o segundo botão, “botao_soma3” e o terceiro, “botao_voltar”. Na propriedade *Text* de cada um deles substitua as frases nesta ordem por: “Soma de Matrizes”, “Matrizes 2x2”, “Matrizes 3x3” e “VOLTAR”.

Faça a mesma coisa em “tela_producto”, editando as propriedades o objeto “tela_producto”, colocando “Produto de Matrizes” em *Text*. Além disto, dois dos botões serão chamados de “botao_producto2” e “botao_producto3” e na propriedade *Text* fica: “Produto de Matrizes” na etiqueta, e “Matrizes 2x2”, “Matrizes 3x3” e “VOLTAR” para os botões.

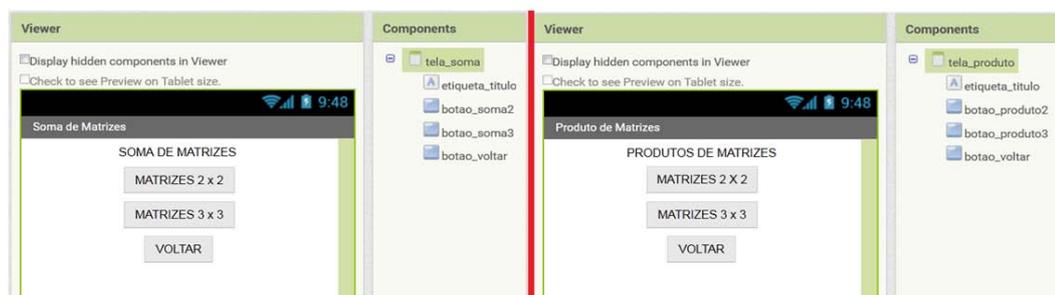


Figura 44 – Ilustração das telas “tela_soma” e “tela_producto” após a inclusão dos componentes.

4.3.3 Construção da calculadora de soma entre matrizes de ordem 2.

O próximo passo é criar a calculadora que efetuará a soma entre matrizes quadradas 2×2 . Inicie então criando uma tela chamada “tela_soma2”. Com esta tela ativada, vá na coluna *Properties*, com o componente “tela_soma2” selecionado, e escolha *Center* em *AlignHorizontal*, marque a opção *Scrollable* e inclua em *Title* a frase “Soma de Matrizes 2×2 ”.

Prossiga adicionando três *TableArrangement* (Arranjos em Tabela) , disponível em *Palette/Layout*, renomeando cada uma para “matriz_A”, “matriz_B” e “matriz_C”. Nas propriedades, deixe 2 em *Columns* (Colunas) e 2 em *Rows* (Linhas), edite o *Width* (Largura) digitando 90 na opção *percent* (por cento). Inclua ainda dois botões, alterando o nome para “botao_calcular” e “botao_voltar”, e na propriedade *Text* dos botões, escreva “CALCULAR” e “VOLTAR” (Figura 45).

Em seguida, acrescente os locais onde o usuário digitará os termos de cada matriz, e os locais que exibirão as respostas, que são quatro caixas de texto  em “matriz_A”, quatro caixas de textos em “matriz_B”, e quatro etiquetas em “matriz_C”. As caixas de textos incluídas em “matriz_A”, altere os nomes para “a11”, “a12”, “a21” e “a22”, fazendo a mesma coisa em “matriz_B”, alterando para “b11”, “b12”, “b21” e “b22”, e ainda em “matriz_C”, ficando as etiquetas nomeadas de “c11”, “c12”, “c21” e “c22”.

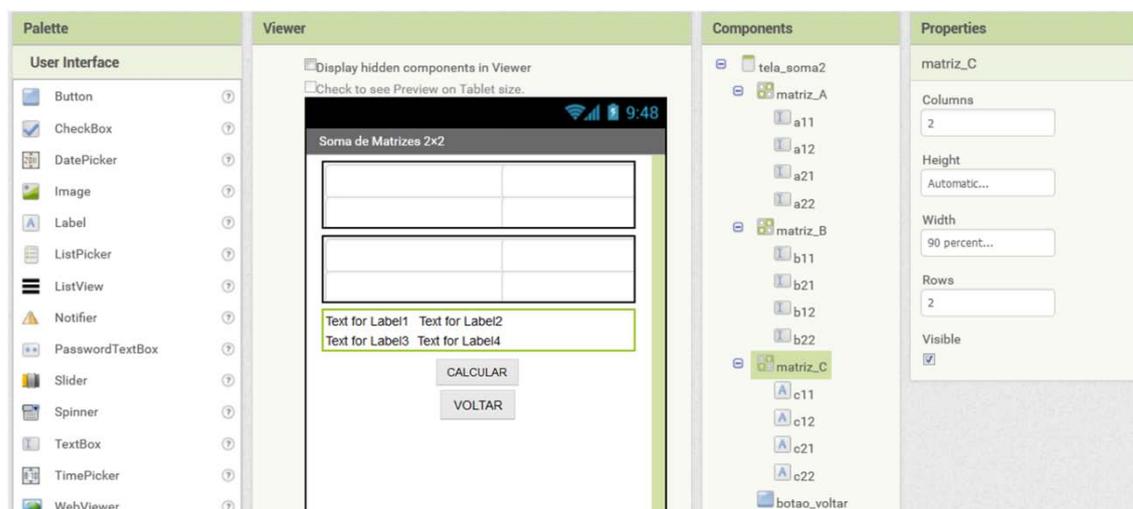


Figura 45 – Visão dos componentes inseridos na “tela_soma2”.

Adicione uma etiqueta antes dos componentes “matriz_A”, “matriz_B” e “matriz_C”, modificando seus nomes para “etiqueta_A”, “etiqueta_B” e “etiqueta_C”, e posteriormente, na propriedade *Text*, coloque para cada uma as seguintes frases: “Matriz A”, “Matriz B” e “Matriz C”.

Antes de prosseguir, faça alguns ajustes para facilitar o manuseio do aplicativo durante os testes. Selecione cada caixa de texto e, na coluna *Properties*, vá em *Width*²⁷, digite 45 em *percent*, retire a frase “*Hint for TextBox*” em *Hint* (Sugestão), pois ela é exibida dentro da caixa de texto quando o aplicativo é executado, atrapalhando a digitação, e marque a opção *NumbersOnly* (somente números) para dar agilidade na digitação dos valores nas caixas de textos.

²⁷ Digitando o valor em *percent* (porcentagem) permite que as caixas de textos se adequem a qualquer tamanho de tela. Por exemplo, na “tela_soma2”, as duas colunas de caixas de texto ocuparão 90% da largura, sendo 45% para cada uma, independente do tamanho da tela do smartphone.

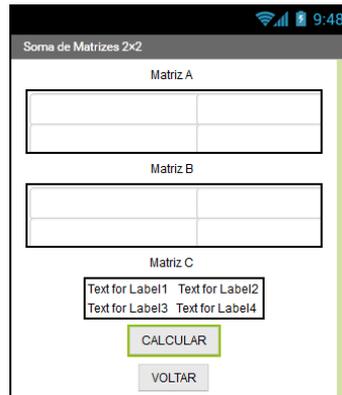


Figura 46 – Captura de tela do smartphone durante a montagem de “tela_soma2”.

A próxima tarefa é construir os blocos responsáveis pela soma dos termos da matriz A com os termos da matriz B e exibir os resultados nos locais corretos na matriz vazia C. Logo, clique no botão *Blocks* (Blocos) , no canto superior direito para ir ao ambiente de edição dos blocos.

Os valores dos termos da matriz C são dados por: $c_{11} = a_{11} + b_{11}$, $c_{12} = a_{12} + b_{12}$, $c_{21} = a_{21} + b_{21}$ e $c_{22} = a_{22} + b_{22}$. Comece então montando os blocos que calculam o valor de “c11”. Na coluna *Blocks*, do lado esquerdo, escolha o bloco azul de soma disponível no submenu *Math* (Matemática) e dois blocos verdes *text* para as caixas de textos “a11” e “b11” fazendo o seguinte agrupamento:

Para exibir a resposta da soma na etiqueta “c11”, procure o componente “c11” na coluna *Blocks* e, escolha o bloco verde *set* , faça o encaixe com o bloco azul de “a11 + b11”, ficando assim: . Para fazer os demais termos da matriz C, basta repetir o método, conforme Figura 47. É possível copiar e colar o primeiro conjunto fazendo a alteração dos nomes, facilitando o trabalho.

O processo de soma será iniciado quando o “botao_calcular” for clicado. Assim seleccione a função *Click* no submenu do componente “botao_calcular” e inclua o conjunto montado anteriormente dentro dele. Veja o resultado na imagem a seguir.

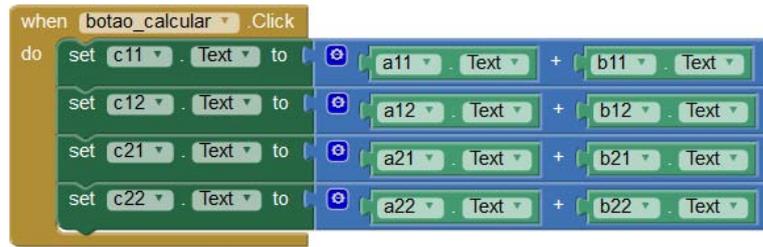
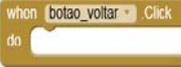


Figura 47 – Agrupamento responsável pela soma dos elementos das matrizes A e B com posterior exibição na matriz C.

Aproveite e faça a composição do “botao_voltar”. Adicione a função  daquele botão e depois no menu *Control* (Controle)  pegue , e no menu *Text* (Texto) , escolha o bloco lilás .

Dentro do bloco lilás, deverá ser escrito o nome da tela que o botão deve abrir: tela_soma. Conecte ele ao bloco “*open another screen*” (abrir outra tela), e associe ao primeiro bloco adicionado. Não pode haver diferença entre o nome dado à tela, com o que foi escrito no bloco lilás.



Figura 48 – Bloco responsável por abrir/retornar a “tela_soma”.

Peça aos alunos testarem²⁸ os aplicativos com os exemplos abaixo para averiguar se há algum problema, corrigindo o que for necessário. Normalmente os problemas são oriundos de equívocos durante as montagens dos blocos, como a troca dos componentes, ou a escolha da função errada do item. Por exemplo, ao invés de ter escolhido função *Text* do componente “a11”  pode ter sido escolhido a função  (Tamanho da Fonte). O professor pode delegar estudantes com mais habilidades para acompanhar os demais auxiliando na monitoria do projeto.

Exemplo 4.9:
$$\begin{bmatrix} -2 & 3 \\ 4 & -5 \end{bmatrix} + \begin{bmatrix} 1 & -4 \\ -3 & 9 \end{bmatrix} = \begin{bmatrix} -1 & -1 \\ 1 & 4 \end{bmatrix}.$$

²⁸ Para testar o aplicativo, consulte o ANEXO A – O que instalar, onde está descrito detalhadamente as três formas de testes.

Exemplo 4.10: $\begin{bmatrix} 0,5 & 0,3 \\ -0,95 & 2 \end{bmatrix} + \begin{bmatrix} 0,3333 & -0,7 \\ -0,05 & -2 \end{bmatrix} = \begin{bmatrix} 0,8333 & -0,4 \\ -1 & 0 \end{bmatrix}$.

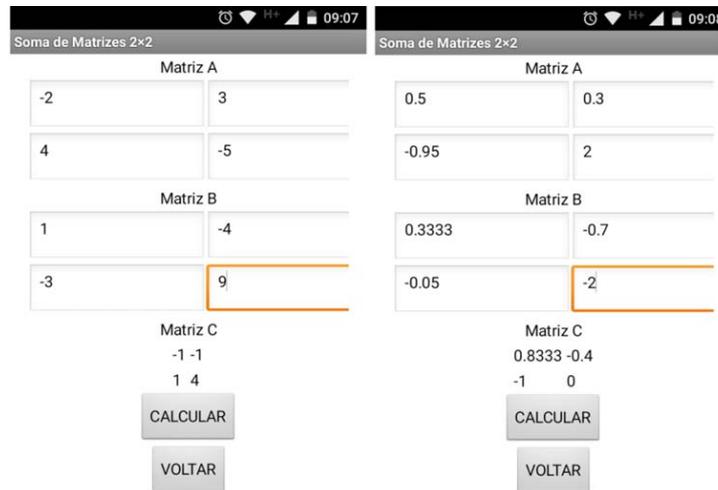


Figura 49 – Capturas de telas do smartphone exibindo o aplicativo solucionando os exemplos propostos.

4.3.4 Construção da calculadora de soma entre matrizes de ordem 3.

Para desenvolver a calculadora responsável pela da soma de matrizes 3×3 será usado o mesmo procedimento feito para soma de matrizes 2×2 . Adicione uma tela e a chame de “tela_soma3”. Vá na coluna *Properties*, com o componente “tela_soma3” selecionado, escolha *Center* em *AlignHorizontal*, marque a opção *Scrollable* e inclua em *Title* a frase “Soma de Matrizes 3×3 ”. Faça a inclusão dos mesmos componentes usando os mesmos nomes, bastando alterar a quantidade de caixas de textos para 9 em “matriz_A” e “matriz_B” além de 9 etiquetas em “matriz_C” (Figura 50).

Da mesma forma, antes de prosseguir, faça os ajustes em cada caixa de texto alterando a largura em *Width*²⁹ para 30 em *percent*, retirando a frase “*Hint for TextBox*” em *Hint* e marcando a opção *NumbersOnly*.

²⁹ Desta vez, a largura de 90% dos componentes “matriz_A” e “matriz_B” serão dividido em três colunas, ficando então 30% para cada.

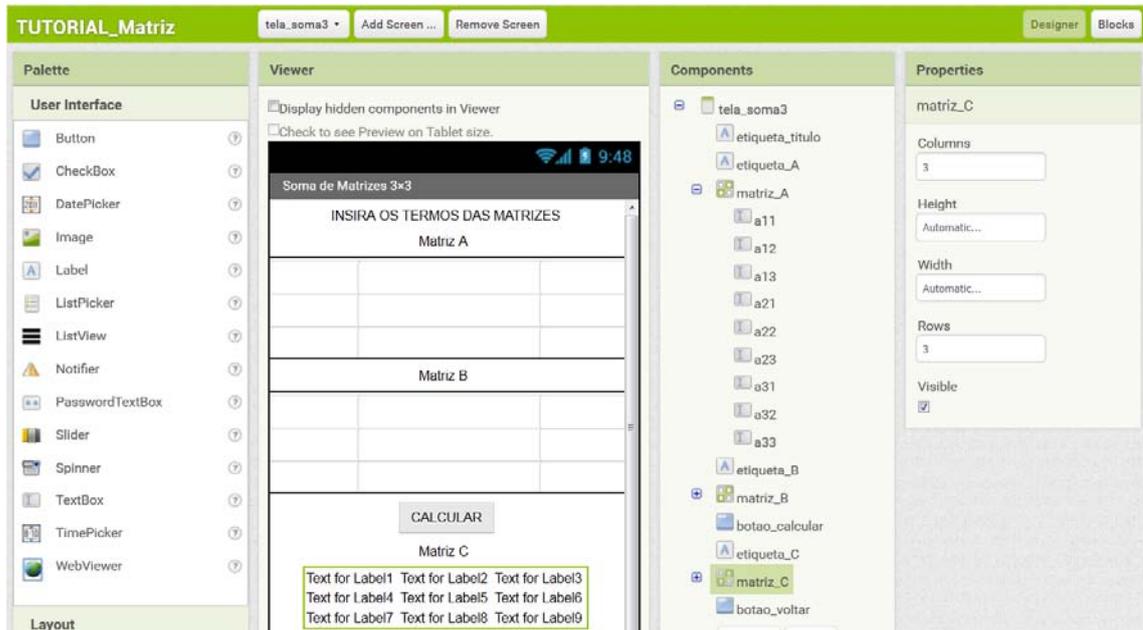


Figura 50 – Visão dos componentes inseridos na tela “tela_soma3”.

Já os blocos responsáveis pelos cálculos dos termos da matriz C e também para retornar à “tela_soma”, ficarão conforme Figura 51.

Ao concluir, solicite aos alunos que testem novamente o aplicativo usando o exemplo listado a seguir:

Exemplo 4.11:
$$\begin{bmatrix} -1 & 3 & -5 \\ 7 & -9 & 11 \\ -13 & 15 & -17 \end{bmatrix} + \begin{bmatrix} 2 & -4 & 6 \\ -8 & 10 & -12 \\ 14 & -16 & 18 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}.$$

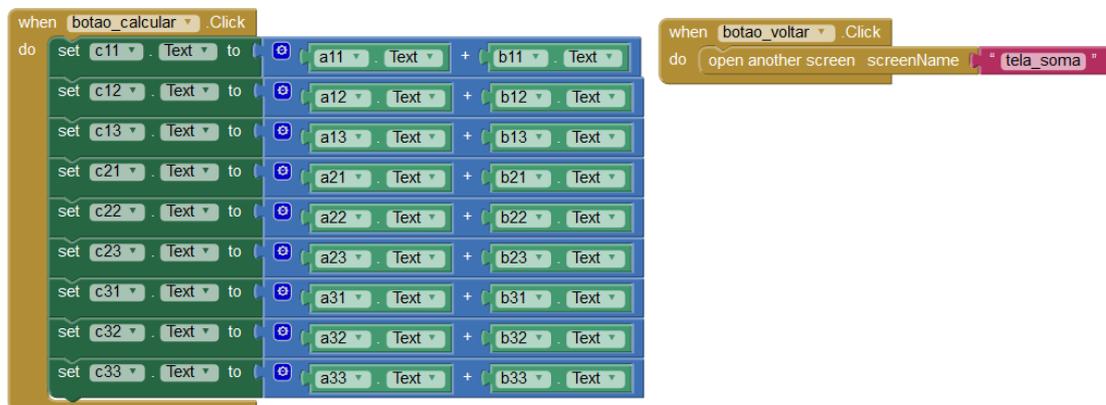


Figura 51 – Agrupamento responsável pela soma dos elementos das matrizes A e B com posterior exibição na matriz C e a função do “botao_voltar”.



Figura 52 – Captura de tela do smartphone exibindo o aplicativo funcionando.

4.3.5 Construção da calculadora de produto de matrizes de ordem 2.

Seguindo o projeto, agora será construído a calculadora de produto de matrizes 2×2 . Adicione outra tela e a chame de “tela_produto2”. Com a tela ativa, vá à coluna *Properties*, e com o componente “tela_produto2” selecionado, escolha *Center* em *AlignHorizontal*, marque a opção *Scrollable* e inclua em *Title* a frase “Produto de Matrizes 2×2 ”.

A distribuição dos componentes da tela é análoga àquela feita em soma de matrizes 2×2 , podendo repetir todo o procedimento, inclusive os nomes e as propriedades dos componentes, lembrando ainda das edições nas propriedades das caixas de texto para que se ajustem às telas dos smartphones.

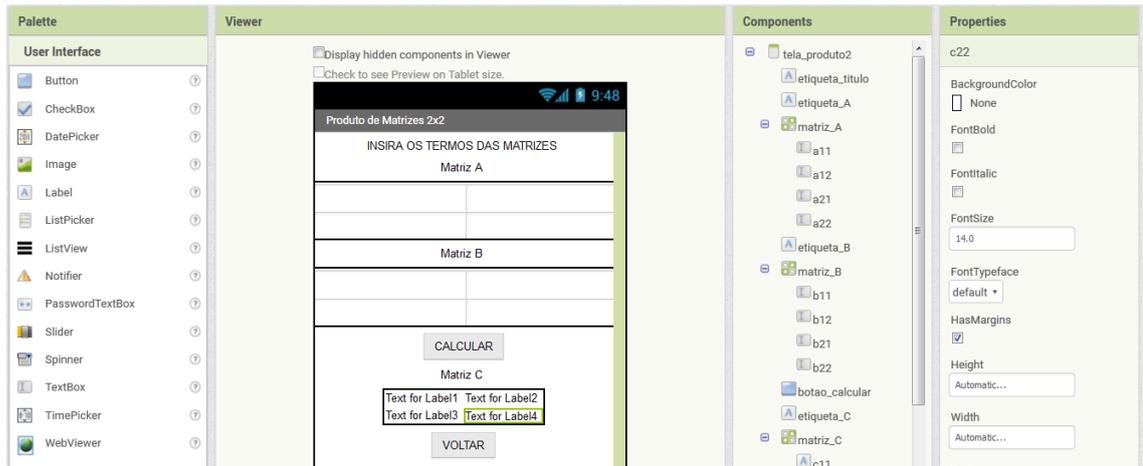


Figura 53 – Visão dos componentes inseridos na “tela_producto2”.

Antes de iniciar a construção dos blocos, volte à **Definição 4.9**, e relembre o processo de produto de matrizes. É interessante o professor esboçar os cálculos para orientar os estudantes na montagem dos blocos. Então, se a matriz C é o produto da matriz A com a matriz B , tem-se que

$$C = \begin{bmatrix} (a_{11} \times b_{11}) + (a_{12} \times b_{21}) & (a_{11} \times b_{12}) + (a_{12} \times b_{22}) \\ (a_{21} \times b_{11}) + (a_{22} \times b_{21}) & (a_{21} \times b_{12}) + (a_{22} \times b_{22}) \end{bmatrix}.$$

Portanto, observando a matriz C , fica fácil perceber que cada termo é composto por dois produtos e uma adição, bastando daí seguir esta ordem durante a montagem dos blocos. Por exemplo, o bloco que determina e exibe o valor do termo c_{11} é dado por:



Figura 54 – Bloco que calcula e exibe o valor de "c11".

Os demais termos da matriz C ficam assim:

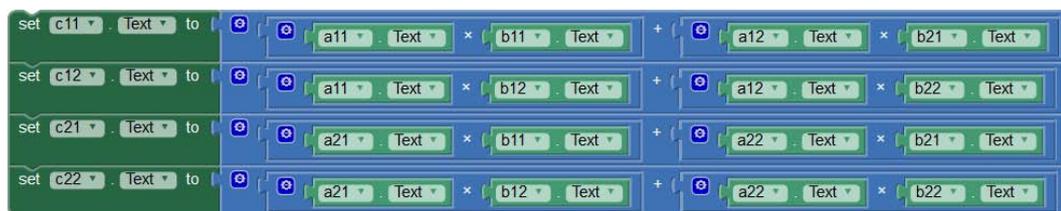
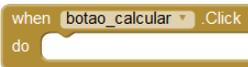


Figura 55 – Agrupamento responsável pelo cálculo de todos os elementos da matriz C .

Para finalizar, inclua o grupo ilustrado na Figura 55, no bloco  do componente “botao_calcular”.

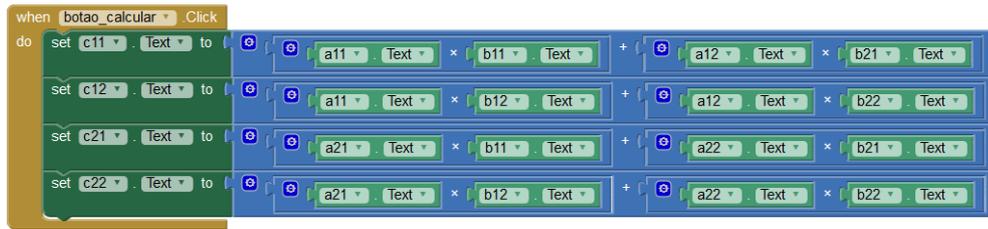


Figura 56 – Agrupamento responsável pelo cálculo da matriz C.

Monte também a função para o “botao_voltar”, que desta vez deverá abrir a “tela_produto”.



Figura 57 – Conjunto responsável por abrir a "tela_produto" através do "botao_voltar".

Concluindo os trabalhos, peça aos alunos testarem o aplicativo com os seguintes exemplos:

Exemplo 4.12: $\begin{bmatrix} -1 & 3 \\ 4 & 2 \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 8 & 10 \\ 10 & 16 \end{bmatrix}$.

Exemplo 4.13: $\begin{bmatrix} 15 & 1 \\ 25 & 0 \end{bmatrix} \times \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 31 & 47 \\ 50 & 75 \end{bmatrix}$.

Exemplo 4.14: $\begin{bmatrix} 51 & 81 \\ 9 & 14 \end{bmatrix} \times \begin{bmatrix} 2 & -3 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 21 & 9 \\ 4 & 1 \end{bmatrix}$.



Figura 58 – Capturas de tela do smartphone exibindo o aplicativo solucionando os exemplos dados.

4.3.6 Construção da calculadora de produto de matrizes de ordem 3.

Dando continuidade, agora será construída a calculadora de produto entre matrizes de ordem 3. Desta vez, a distribuição dos componentes da tela deverá ser idêntica àquela feita para soma entre matrizes 3×3 , podendo repetir todo o procedimento, inclusive os nomes e as propriedades dos componentes, lembrando ainda das edições nas propriedades das caixas de texto para que se ajustem às telas dos smartphones. Assim, crie uma nova tela de nome “tela_produto3”, e acrescente os respectivos componentes. Veja:

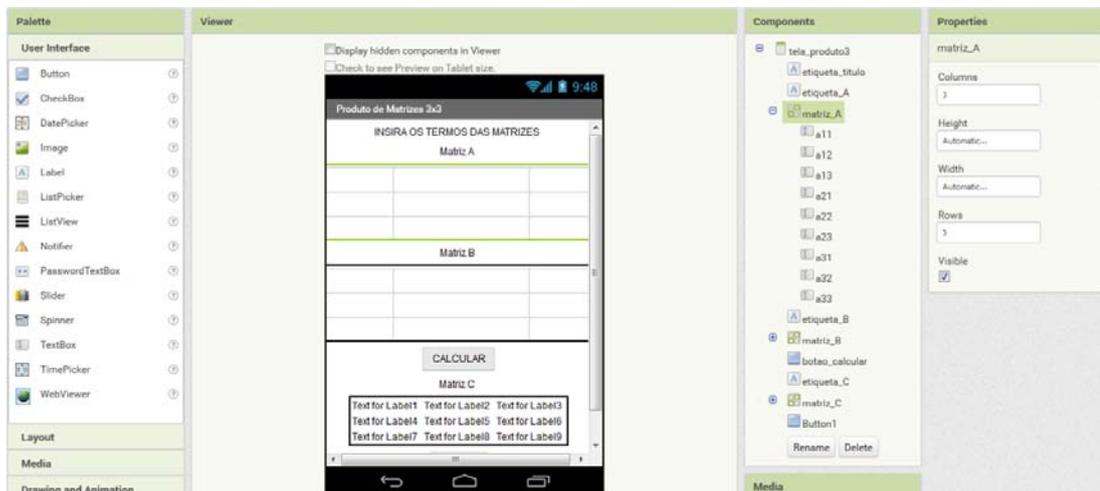


Figura 59 – Visão dos componentes inseridos na “tela_produto3”.

Faça também o esboço dos cálculos do produto entre matrizes quadradas de ordem 3, onde os termos da matriz C são dados por:

$$C = \begin{bmatrix} (a_{11} \times b_{11}) + (a_{12} \times b_{21}) + (a_{13} \times b_{31}) & (a_{11} \times b_{12}) + (a_{12} \times b_{22}) + (a_{13} \times b_{32}) & (a_{11} \times b_{13}) + (a_{12} \times b_{23}) + (a_{13} \times b_{33}) \\ (a_{21} \times b_{11}) + (a_{22} \times b_{21}) + (a_{23} \times b_{31}) & (a_{21} \times b_{12}) + (a_{22} \times b_{22}) + (a_{23} \times b_{32}) & (a_{21} \times b_{13}) + (a_{22} \times b_{23}) + (a_{23} \times b_{33}) \\ (a_{31} \times b_{11}) + (a_{32} \times b_{21}) + (a_{33} \times b_{31}) & (a_{31} \times b_{12}) + (a_{32} \times b_{22}) + (a_{33} \times b_{32}) & (a_{31} \times b_{13}) + (a_{32} \times b_{23}) + (a_{33} \times b_{33}) \end{bmatrix}$$

Deste modo, cada termo é composto por três produtos e uma soma, bastando seguir a ordem expressa. Novamente usando o elemento c_{11} como exemplo, o cálculo é dado por:



Figura 60 – Bloco que calcula o valor de "c11".

Para conseguir o efeito mostrado na Figura 60, primeiro deve aumentar a quantidade

de fatores no bloco azul da adição. No canto superior esquerdo deste bloco existe um ícone. Ao clicá-lo aparecerá a opção de incluir ou retirar fatores de adição. Para aumentar basta arrastar o bloco *number* (número) para o lado direito. A mesma coisa acontece com o bloco de multiplicação.

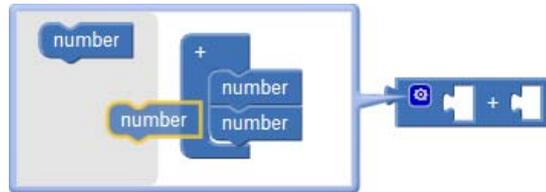


Figura 61 – Procedimento para aumentar fatores no bloco de adição.

Já para que os termos dos blocos de multiplicação fiquem um sobre o outro, clique com o botão direito do mouse na parte azul de cada conjunto de produto, e escolha a opção *External Inputs*. Para desfazer, clique em *Inline Inputs*.

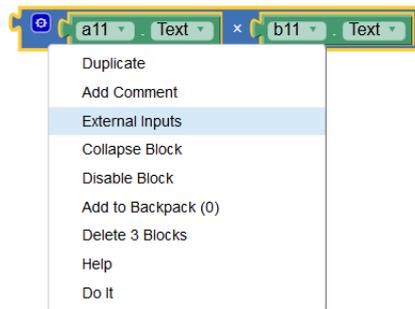


Figura 62 – Demonstração da função *External Inputs* que facilita o manuseio de blocos grandes.

Depois acople o bloco azul (Figura 60) ao bloco `set c11 . Text to`:



Figura 63 – Bloco que calcula o valor do elemento "c11".

Termine todos os conjuntos de blocos de cada termo da matriz C, junte um sob o outro e coloque dentro da função *Click* `when botao_calcular .Click` oriundo do componente “botao_calcular”. Faça ainda a composição para o “botao_voltar”.

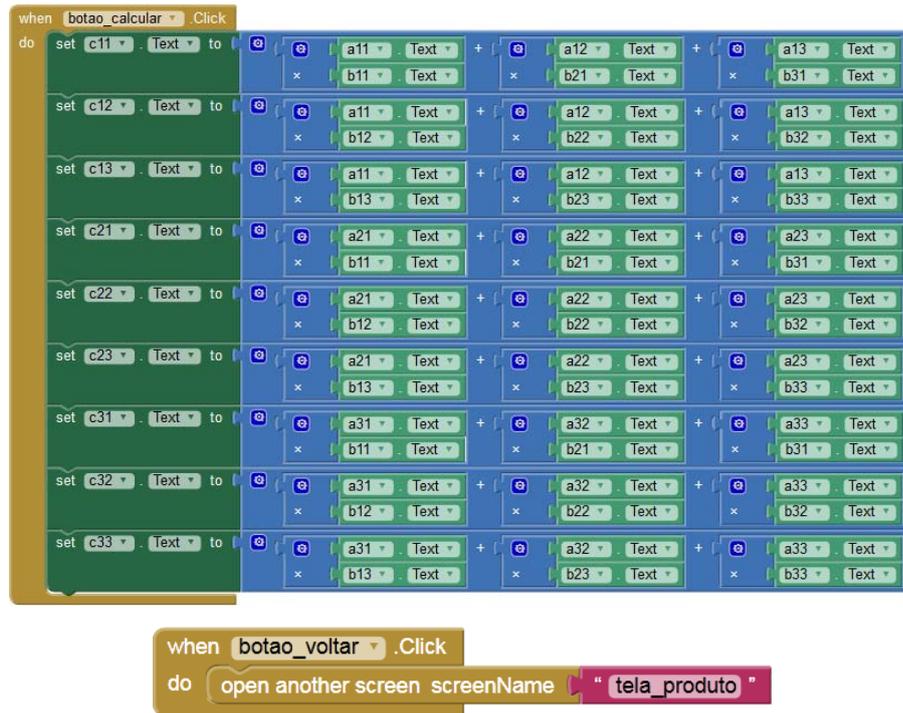


Figura 64 – Conjunto final dos blocos de todos termos da matriz C e a função do “botao_voltar”.

Para testar o programa, use o **Exemplo 4.15**.

Exemplo 4.15:
$$\begin{bmatrix} 3 & 0 & 2 \\ 9 & 1 & 7 \\ 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & -2 \\ -2 & 1 & -3 \\ -1 & 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

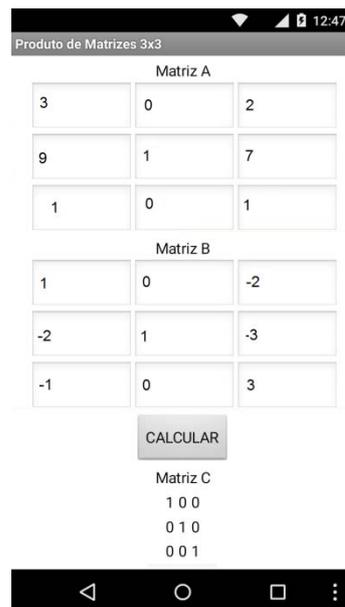


Figura 65 – Captura de tela com o aplicativo em funcionamento.

4.3.7 Construção da tela da calculadora de determinantes de matrizes.

Nesta etapa, será montada a tela da calculadora de determinantes de matrizes 2×2 e 3×3 . As próximas seções contém as etapas para montar os blocos de cada determinante. Abra a “tela_determinantes” e na aba *Properties* escolha *Center* em *AlignHorizontal*, marque a opção *Scrollable* e em *Title* coloque o título “Determinantes de Matrizes”. É possível fazer todas as calculadoras na mesma tela. Siga os passos abaixo comparando com a Figura 66.

- (1) Adicione uma etiqueta;
- (2) Adicione uma *TableArrangement* 2×2 e insira quatro caixas de texto dentro dela. Mude os *Width* das caixas de textos para *45 Percent*;
- (3) Adicione um botão;
- (4) Adicione um *HorizontalArrangement* e coloque duas etiquetas dentro dela;
- (5) Adicione uma etiqueta;
- (6) Adicione uma *TableArrangement* 3×3 e ponha nove caixas de texto nela. Mude os *Width* das caixas de textos para *30 Percent*;
- (7) Adicione um botão;
- (8) Adicione um *HorizontalArrangement* e insira duas etiquetas dentro;
- (9) Adicione um botão.

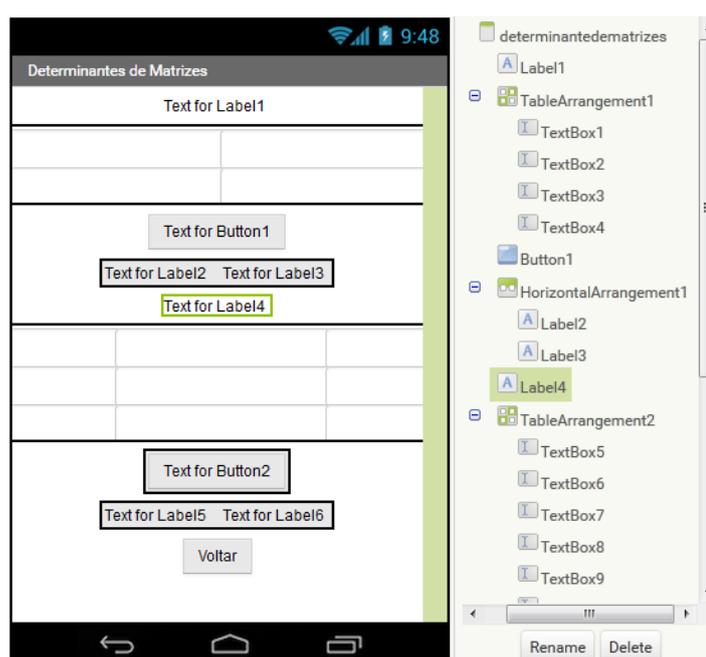


Figura 66 – Captura de tela exibindo os componentes inseridos nos passos anteriores.

Agora organize os nomes e algumas propriedades antes de ir para a construção dos blocos:

(1) Nas etiquetas, faça:

- *Label1*: altere o nome para “etiqueta_A”, modifique o *Text* para: “Matriz 2x2”;
- *Label2*: altere o nome para “etiqueta_detA”, modifique o *Text* para: “det(A) =”;
- *Label3*: altere o nome para “etiqueta_respostaA”, modifique o *Text* deixando vazio;
- *Label4*: altere o nome para “etiqueta_B”, modifique o *Text* para: “Matriz 3x3”;
- *Label5*: altere o nome para “etiqueta_detB”, modifique o *Text* para: “det(B) =”;
- *Label6*: altere o nome para “etiqueta_respostaB”, modifique o *Text* deixando vazio.

(2) Nos botões, faça:

- *Button1*: altere o nome para “botao_calcularA”, modifique o *Text* para: “CALCULAR”;
- *Button2*: altere o nome para “botao_calcularB”, modifique o *Text* para: “CALCULAR”;
- *Button3*: altere o nome para “botao_voltar”, modifique o *Text* para: “VOLTAR”.

(3) Modifique os arranjos:

- *TableArrangement1*: altere o nome para “matriz_A”, e mude *Width* para 90 percent;
- *HorizontalArrangement1*: altere o nome para “det_matriz_A”, e mude *AlignHorizontal* para *Center* e *Width* para 90 percent;
- *TableArrangement2*: altere o nome para “matriz_B” e mude *Width* para 90 percent;
- *HorizontalArrangement3*: altere o nome para “det_matriz_B”, mude *AlignHorizontal* para *Center* e *Width* para 90 percent.

(4) Nas caixas de texto, altere:

- retire a frase “*Hint for TextBox*”;
- marque a opção *NumbersOnly*;
- renomeie as caixas de texto contidas em “matriz_A” para “a11”, “a12”, “a21” e “a22” e as caixas de texto contidas em “matriz_B” para “b11”, “b12”, “b13”, “b21”, “b22”, “b23”, “b31”, “b32” e “b33”.



Figura 67 – Captura de tela do celular exibindo as modificações realizadas.

4.3.8 Montagem dos blocos da calculadora de determinantes de matrizes 2×2.

Agora serão montadas as composições que efetuarão os cálculos dos determinantes de matrizes 2×2. De acordo com a **Definição 4.11**, o determinante de uma matriz quadrada de ordem 2 é igual a $[(a_{11} \times a_{22}) - (a_{12} \times a_{21})]$. Então, neste sentido, os blocos ficam:



Figura 68 – Parte do conjunto que calcula determinantes de matrizes 2×2.

A resposta deverá aparecer no componente “etiqueta_respostaA” representada pelo bloco verde `set etiqueta_respostaA . Text to`, e o responsável pela ordem é a função *Click*

do componente “botao_calcularA”. Portanto a composição fica assim:

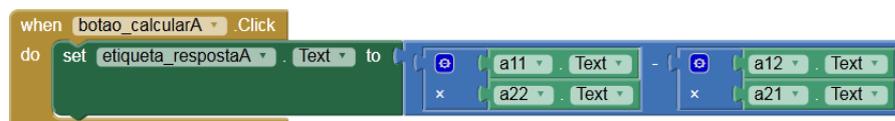


Figura 69 – Conjunto que calcula determinantes de matrizes 2×2.

Para testar o aplicativo, use os exemplos a seguir.

Exemplo 4.16: $A = \begin{bmatrix} 1 & 3 \\ 0 & 2 \end{bmatrix}$, $\det(A) = (1 \times 2) - (3 \times 0) = 2$.

Exemplo 4.17: $M = \begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix}$, $\det(M) = (2 \times 1) - (4 \times 3) = -10$.



Figura 70 – Capturas de telas do smartphone exibindo o aplicativo solucionando os exemplos propostos.

4.3.9 Montagem dos blocos da calculadora de determinantes de matrizes 3×3 .

O objetivo agora é construir os blocos que calculam determinantes de matrizes quadradas de ordem 3, através do método de Sarrus. Novamente, para auxiliar na montagem dos

blocos, faça o esboço do cálculo usando a **Definição 4.11** na matriz $B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$.

Logo $\det(B) = [(b_{11} \times b_{22} \times b_{33}) + (b_{12} \times b_{23} \times b_{31}) + (b_{13} \times b_{21} \times b_{32})] - [(b_{13} \times b_{22} \times b_{31}) + (b_{11} \times b_{23} \times b_{32}) + (b_{12} \times b_{21} \times b_{33})]$.

Repetindo a expressão utilizando os blocos, têm-se:



Figura 71 – Conjunto de blocos dos produtos de cada diagonal para determinantes de matrizes 3×3 .

Fazendo a soma dos produtos das três diagonais principais e das três diagonais secundárias, fica:



Figura 72 – Conjunto de soma dos produtos de cada diagonal para determinantes de matrizes 3x3.

A resposta deverá aparecer no componente “etiqueta_respostaB” representada pelo bloco verde `set etiqueta_respostaB . Text to`, e o responsável pela ordem é a função *Click* `when botao_calcularB . Click` do componente “botao_calcularB”. Junte os dois blocos da Figura 72 num bloco de subtração, e faça a composição padrão, resultando na seguinte imagem:

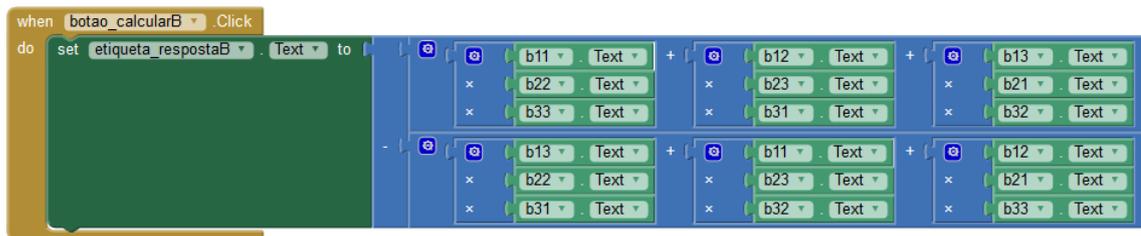


Figura 73 – Bloco que calcula determinantes de matrizes 3x3.

Além disso, há a função do “botao_voltar”.



Figura 74 – Bloco responsável por abrir a tela inicial “Screen1”.

Use o **Exemplo 4.8** e exemplo abaixo para testar o aplicativo.

$$\mathbf{Exemplo 4.18: } B = \begin{bmatrix} 2 & 3 & -1 \\ 5 & 2 & 0 \\ 1 & 4 & -3 \end{bmatrix} \Rightarrow \det(B) = [(2 \times 2 \times (-3)) + (3 \times 0 \times 1) +$$

$$((-1) \times 5 \times 4)] - [((-1) \times 2 \times 1) + (2 \times 0 \times 4) + (3 \times 5 \times (-3))] = [-12 + 0 - 20] - [-2 + 0 - 45] = -32 + 47 = 15.$$



Figura 75 – Captura de telas do smartphone exibindo o aplicativo solucionando os exemplos propostos.

4.3.10 Finalização do projeto com modificações nas propriedades dos componentes.

Veja que o projeto desenvolvido já consegue atingir os objetivos propostos, que são somar, multiplicar e encontrar determinantes de matrizes 2×2 e 3×3 . Porém é necessário retornar em cada tela e efetuar ajustes nas propriedades de alguns componentes para que a aparência do aplicativo fique mais atraente. Além disso é preciso criar os blocos dos botões responsáveis por abrir as telas a partir de “Screen1”.

Então comece pelos botões. Vá à tela “Screen1” para atribuir funções ao “botao_soma”, “botao_produto” e “botao_determinantes” de modo que ao serem clicados, as respectivas telas sejam abertas. Do mesmo jeito que fora feito antes, os conjuntos ficarão da seguinte forma:

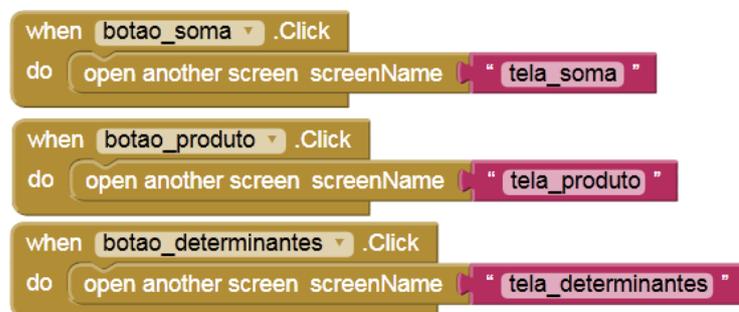


Figura 76 – Blocos responsáveis por abrir as telas “tela_soma”, “tela_produto” e “tela_determinantes”.

Agora, acesse as telas “tela_soma” e “tela_produto” para construir os blocos dos bo-

tões que chamam “tela_soma2”, “tela_soma3”, “tela_produto2” e “tela_produto3”, e também retornar à tela inicial “Screen1”.



Figura 77 – Captura de tela do smartphone exibindo a “tela_soma” e os blocos responsáveis por abrirem as telas.



Figura 78 – Captura de tela do smartphone exibindo a “tela_produto” e os blocos responsáveis por abrirem as telas desejadas.

Abaixo segue uma lista de edições usadas no aplicativo deste tutorial, que podem inspirar a deixar o seu projeto personalizado. Assim, selecionando cada objeto e indo no menu *Properties* faça:

(1) Em todas caixas de texto:

- Coloque 25 para *FontSize* (tamanho da fonte);
- Verifique se em *Width*, foi digitado 45 em *percent* quando for matriz quadrada de ordem 2 e 30 em *percent* para matriz quadrada de ordem 3;
- Verifique se foi retirado todas as frases “*Hint for TextBox*” em *Hint*;
- Confira se foi marcado a opção *NumbersOnly*;
- Selecione *Center* em *TextAlignment*;
- Altere os *TextColor* (cor do texto) para vermelho quando a caixa for da matriz A e azul para as caixas da matriz B.

(2) Nas etiquetas “etiqueta_A”, “etiqueta_B” e “etiqueta_C”:

- Marque *FontBold* (*Negrito*);
- Coloque 25 para *FontSize*;
- Selecione *Center* em *TextAlignment*;
- Altere o *TextColor* da “etiqueta_A” para vermelho e “etiqueta_B” para azul.

(3) Nas etiquetas de “matriz_C”:

- Marque *FontBold*;
- Coloque 25 para *FontSize*;
- Retire a frase “*Text for Label*” em *Text*;
- Selecione *Center* em *TextAlignment*.

(4) Insira uma etiqueta chamada “etiqueta_titulo” nas telas de calculadoras antes de todos os componentes e edite:

- Marque *FontBold*;
- *FontSize* para 20;
- Mude *Text* para “INSIRA OS TERMOS DAS MATRIZES”;
- Selecione *Center* em *TextAlignment*;
- Faça a mesma coisa com a “etiqueta_titulo” da tela “Screen1”.

(5) Troque de lugar o “botao_calcular” para ficar antes do componente “etiqueta_C”.

(6) Nos botões:

- Escolha uma cor em *BackgroundColor* (Cor de Fundo);
- Marque *FontBold*;
- Coloque 20 para *FontSize*;
- Em *Shape* (Forma), escolha *rounded*;
- Selecione *Center* em *TextAlignment*.

(7) Escolha uma boa figura na internet e adicione através do botão *UploadFile*  no menu *Média* (Mídia). Depois selecione o componente “Screen1”, e vá e *BackgroundImage* (Imagem de Fundo), selecionando o arquivo que foi inserido.

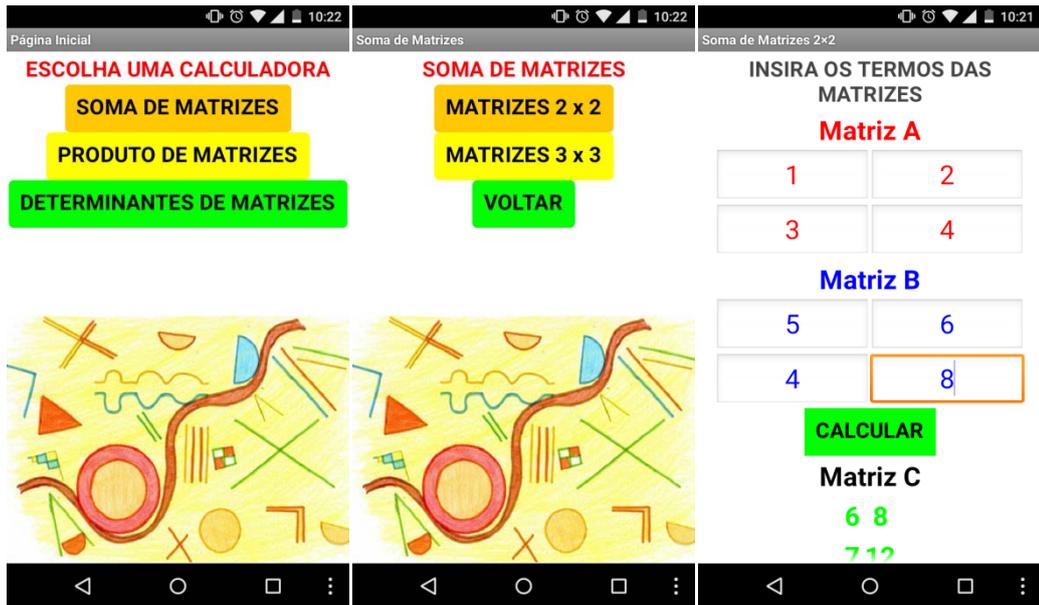


Figura 79 – Captura de algumas telas do aplicativo exibindo as modificações aplicadas.

Outro incremento interessante para ser aplicado é a inserção de um botão que limpe todas as caixas de texto e etiquetas de respostas a cada vez que for inserir novos dados. Para isso, adicione (em todas as calculadoras) antes do “botao_voltar” um novo botão chamado “botao_limpar” com o *Text* “LIMPAR” (Figura 80), e no ambiente *Blocks* selecione a função

when **botao_limpar** .Click
do  do “botao_limpar”, e também o bloco set de cada componente caixa de texto das matrizes, além das etiquetas que exibem respostas, agregando ao bloco  disponível no menu *Control*, conforme a composição ilustrada na Figura 81.

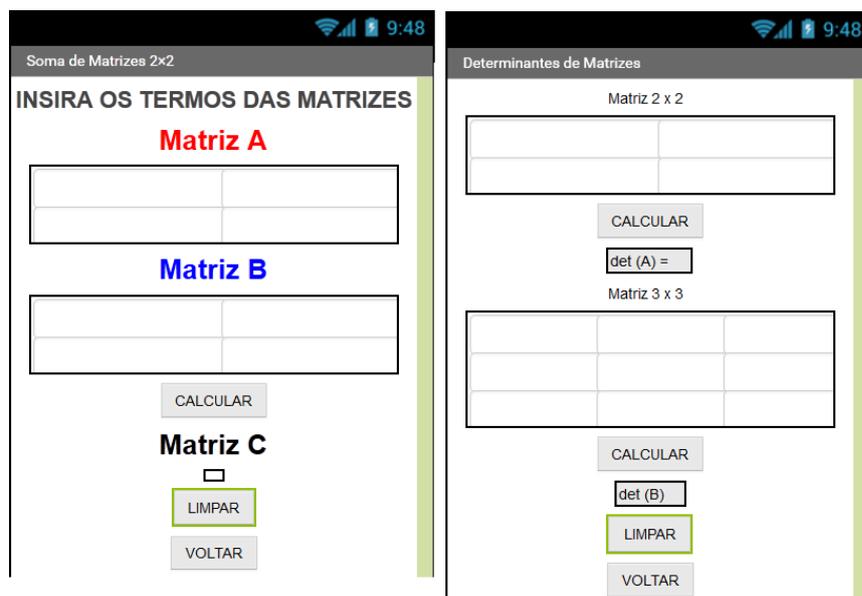


Figura 80 – Ilustração da inserção do “botao_limpar” nas telas “tela_soma2” e “tela_determinantes”.

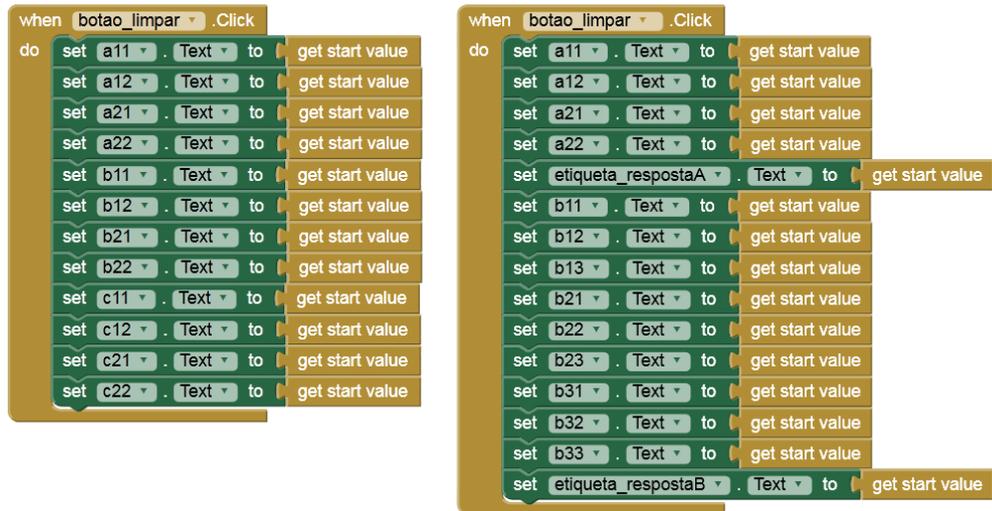


Figura 81 – Exemplos de blocos responsáveis por limpar caixas de textos e etiquetas.

4.3.11 Desafio.

Para concluir esta sequência didática, proponha à turma uma tarefa em forma de desafio, que pode ser cumprida na própria escola ou como trabalho extra-classe, onde os alunos deverão usar os novos conhecimentos adquiridos.

Solicite a eles que aumentem a quantidade de funções do aplicativo que acabaram de criar da seguinte forma: ou incluam uma calculadora para achar a inversa de matrizes de ordem 2 e 3, ou, que adicionem uma calculadora para achar a soma, o produto e o determinante de matrizes de ordem 4. Em ambas escolhas, oriente os estudantes a pesquisarem em livros e na internet sobre o conteúdo, e a partir dos estudos montem os métodos de resolução nos blocos do aplicativo.

5 SISTEMAS DE EQUAÇÕES LINEARES

Neste capítulo, será apresentada a terceira e última sequência didática deste trabalho, na qual ensinará a construir um aplicativo que soluciona sistemas de equações lineares de duas e três incógnitas usando o Teorema de Cramer.

Sistemas lineares é outro conteúdo matemático recorrente na vida escolar do aluno, não só na Matemática, como na Física e na Química, alcançando também os cursos superiores das áreas de Exatas. E da mesma forma, quando não é bem apresentado pelo docente deixa grandes prejuízos no processo de ensino-aprendizagem, em especial no que tange às definições e métodos práticos para solucionar exercícios.

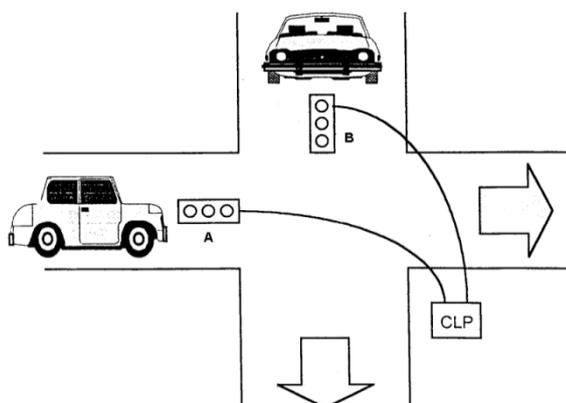


Figura 82 – Demonstração prática da aplicação de sistemas lineares no cotidiano.

5.1 Principais resultados.

Aqui serão listados os principais resultados sobre sistemas de equações lineares que nortearão esta sequência didática. As definições, teoremas, proposições e alguns exemplos

c) *Sistema Impossível* (SI): ocorre quando o sistema de equações lineares não admite solução alguma.

Em resumo:

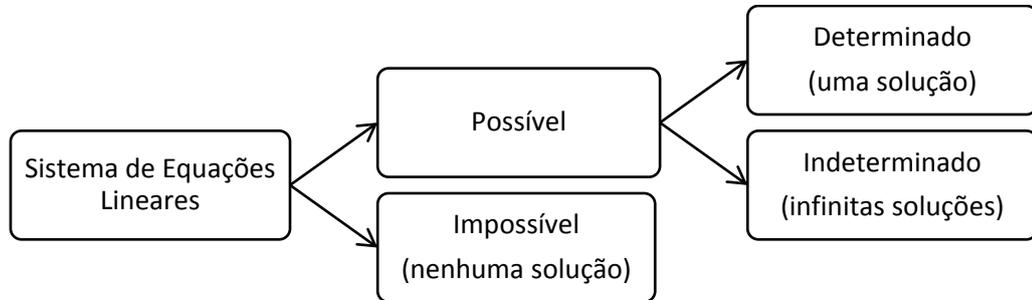


Figura 83 – Classificação dos sistemas de equações lineares de acordo com suas soluções.

Teorema 5.1 - Teorema de Cramer: Considere um sistema linear com n equações e n incógnitas. Se o determinante da matriz dos coeficientes, $\det(S)$, for diferente de zero, então o sistema será possível e terá solução única $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \dots, \alpha_n)$, tal que $\alpha_i = \frac{\det(S_i)}{\det(S)}$, $\forall i \in \{1, 2, 3, \dots, n\}$, onde $\det(S_i)$ é o determinante da matriz obtida de S , substituindo-se a i 'ésima coluna pela coluna de termos independentes das equações do sistema.

5.2 Exemplos.

Exemplo 5.1: Dado o sistema de equações lineares $K: \begin{cases} x + 3y = 5 \\ 2x + 6y = 9 \end{cases}$, determine os valores de x e y .

Solução:

A forma matricial dos coeficientes do sistema K é a matriz $K = \begin{bmatrix} 1 & 3 \\ 2 & 6 \end{bmatrix}$. Verifica-se que $\det(K)$ é igual a 0, e portanto indica que o sistema é possível e indeterminado ou impossível.

Exemplo 5.2: Dado o sistema de equações lineares $S: \begin{cases} 2x - 3y + 7z = 1 \\ x + 3z = 5 \\ 2y - z = 0 \end{cases}$, determine os valores de x , y e z .

Solução:

A forma matricial dos coeficientes do sistema S é a matriz $S = \begin{bmatrix} 2 & -3 & 7 \\ 1 & 0 & 3 \\ 0 & 2 & -1 \end{bmatrix}$. Verifica-se

que $\det(S)$ é igual a -1 , e portanto diferente de 0 , o que indica que o sistema tem solução única. Em seguida, substituindo cada coluna dos coeficientes de x , y e z pela coluna dos termos independentes $\begin{bmatrix} 1 \\ 5 \\ 0 \end{bmatrix}$ para achar os determinantes das novas matrizes S_1 , S_2 e S_3 respectivamente, onde os resultados são: $\det(S_1) = 49$, $\det(S_2) = -9$ e $\det(S_3) = -18$.

Tem-se portanto: $x = \frac{\det(S_1)}{\det(S)} = \frac{49}{-1} = -49$, $y = \frac{\det(S_2)}{\det(S)} = \frac{-9}{-1} = 9$ e $z = \frac{\det(S_3)}{\det(S)} = \frac{-18}{-1} = 18$.

5.3 Sequência didática para construção do aplicativo.

São objetivos desta sequência didática:

- criar um programa que soluciona sistemas de equações lineares onde o número de equações é igual ao número de incógnitas, com coeficientes reais e ordem no máximo 3 e fazê-lo funcionar em dispositivos móveis como smartphones e tablets;
- propiciar ao estudante meios de fixar as definições mais usadas nas operações envolvendo sistemas de equações lineares;
- conhecer aplicações dos conectivos *se*, *ou*, *então*, *senão*, *se e somente se*, *falso*, *verdadeiro*, entre outros;

Consideram-se competências (conhecimentos e habilidades) desta sequência didática:

- identificar sistemas de equações lineares;
- saber classificar sistemas de equações lineares de acordo com a solução apresentada;
- saber aplicar o teorema de Cramer;
- dominar as propriedades de matrizes;
- saber somar, subtrair e multiplicar matrizes;
- saber calcular determinantes de matrizes 2×2 e de matrizes 3×3 pelo método de Sarrus.

Os recursos necessários para a execução desta sequência didática são:

- laboratório de informática;
- acesso à internet;
- rede sem fio (wireless);
- smartphones ou tablets que utilizem o sistema operacional Android;
- cabo de conexão tipo USB;
- aplicativo *MIT AI2 Companion* instalado nos smartphones ou tablets;
- conta de usuário no Google;
- lápis, caderno, borracha para anotações.

5.3.1 Esboço da estrutura do aplicativo e escolha dos componentes de algumas telas.

Inicie a aula conversando com os alunos à respeito do modo que eles vêem o futuro aplicativo e as possíveis telas que ele poderá ter. Crie com o grupo um roteiro através de desenhos e esquemas que ajude a idealizar como o aplicativo funcionará.

O esboço auxiliará durante a montagem dos componentes de interação do usuário com o aplicativo bem como na criação dos blocos relacionados à parte lógica do programa. A ideia é ter um programa simples e eficiente, permitindo uma navegação fácil entre as calculadoras. Veja na Figura 84 um modelo de arranjo das telas que compõe o aplicativo.

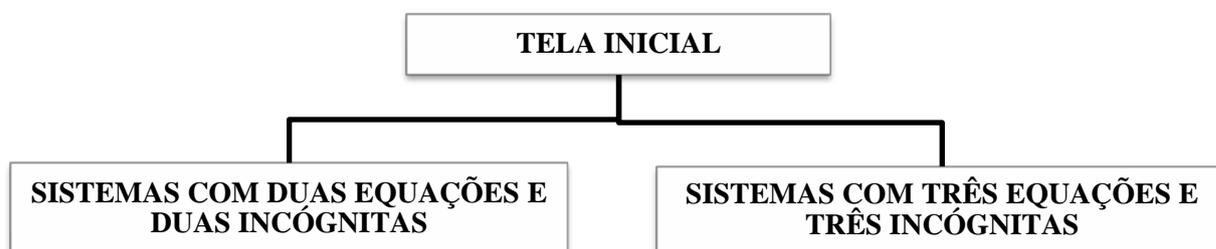


Figura 84 – Esquema das múltiplas telas do aplicativo.

Prosseguindo, professor e alunos deverão discutir quais elementos serão necessários para compor as telas do aplicativo. No quadro, desenhe um retângulo que simule a tela de um smartphone (Figura 85), e esboce a tela inicial do programa incluindo frases para destacar al-

guma informação e botões para cumprir a função de abrir novas telas.

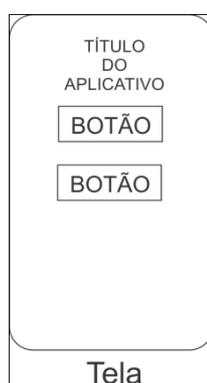


Figura 85 – Esboço da tela inicial do aplicativo.

Converse ainda como deve ser alguma tela que efetuará os cálculos envolvendo os sistemas lineares, por exemplo, sistemas lineares de duas incógnitas. Os alunos devem concluir que a tela deverá ter, pelo menos, os locais para receber os valores dos coeficientes, e também o local onde os resultados aparecerão. Lembre-se ainda que um objeto deverá ordenar que os cálculos sejam iniciados. Esta ordem pode partir de um botão também. Vale ressaltar que este modelo poderá ser usado na tela da outra calculadora do aplicativo.

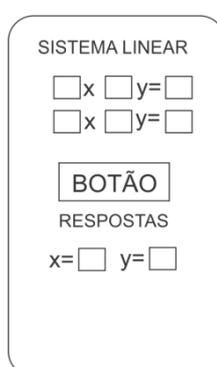


Figura 86 – Esboço das telas que conterão as calculadoras.

5.3.2 Início da construção do aplicativo com a inclusão das telas.

A partir do esboço, sugira aos estudantes que iniciem o projeto incluindo as três telas (Figura 84), após coloquem os componentes necessários nelas, finalizando com a montagem

dos blocos responsáveis pelos cálculos, fazendo primeiro para sistemas de duas incógnitas e depois para três. Portanto, vá em *Projects*³⁰ (Projetos), e em seguida *Start new project* (Começar novo projeto) e escolha um nome para o seu trabalho.

Em seguida selecione o componente “*Screen1*”, e no menu *Properties* (Propriedades), altere a opção *Center* (Centro) em *AlignHorizontal* (Alinhamento Horizontal). Em *AppName* (Nome do Aplicativo) coloque o nome que preferir. Neste tutorial foi usado “Sistemas de Equações Lineares”. Esta frase é a que aparecerá abaixo do ícone do aplicativo quando estiver instalado no dispositivo. Marque a opção *Scrollable* (Rolar) , isto permite a tela rolar à medida que mais componentes são acrescentados. Em *Title* (Título) coloque um título para esta tela do aplicativo. No exemplo foi colocado “Sistemas de Equações Lineares” também.

Adicione uma etiqueta *Label* e dois botões *Button*, renomeando³¹ a etiqueta para “etiqueta_titulo” e os botões para “botao_2” e “botao_3” (de duas incógnitas e três incógnitas). Prosseguindo, selecione o componente “etiqueta_titulo” e em *Text* coloque “ESCOLHA O NÚMERO DE INCÓGNITAS DO SISTEMA”. Com os componentes dos botões selecionados altere também os itens *Text* para “SISTEMA DE 2 INCÓGNITAS” e “SISTEMA DE 3 INCÓGNITAS”.

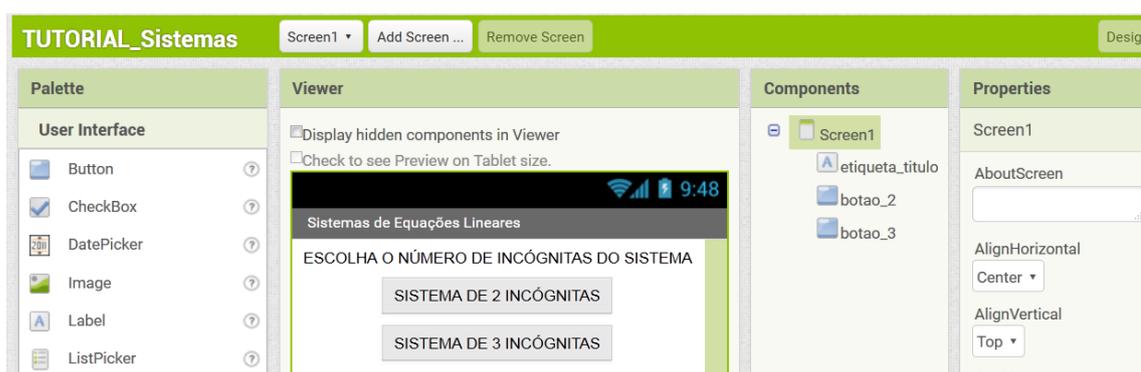


Figura 87 – Tela inicial do aplicativo com os componentes inseridos.

Para adicionar novas telas, clique no botão *AddScreen* (Adicionar Tela) *Add Screen ...* na

³⁰ Havendo dúvidas, consulte o ANEXO B – **Conhecendo a plataforma**, onde está detalhado o processo para iniciar um novo projeto.

³¹ Para renomear um componente, selecione-o no menu *Components* e clique no botão *Rename* (Renomear) *Rename*, que aparecerá uma caixa de diálogo solicitando a nova nomenclatura.

parte superior. Inclua duas. A primeira chame de “tela_2” e a segunda de “tela_3”, para combinar com os botões que foram criados anteriormente.

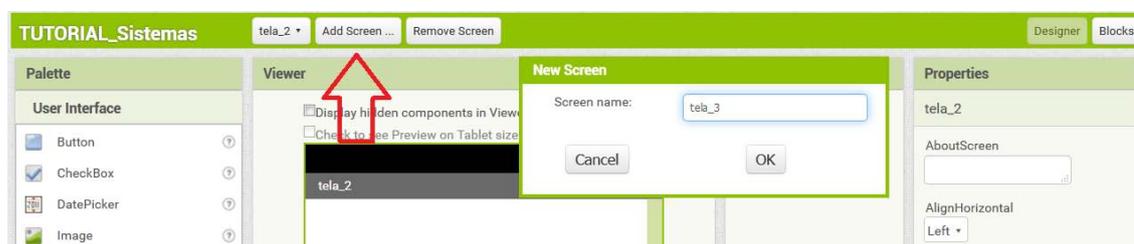


Figura 88 – Demonstração do procedimento para adicionar novas telas.

Assim que terminar, automaticamente o MAI deixará disponível a última tela criada. Para alternar entre as telas, clique no botão do lado esquerdo do *AddScreen* e escolha a tela que deseja trabalhar.

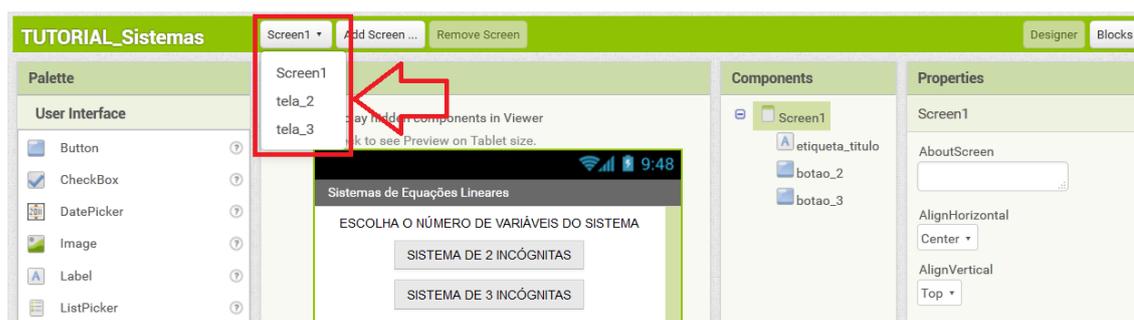


Figura 89 – Demonstração do procedimento para alternar entre as telas.

5.3.3 Construção da tela da calculadora de que soluciona sistemas lineares de duas incógnitas.

A primeira calculadora a ser construída será aquela que soluciona sistemas lineares de duas equações e duas incógnitas que será montada na “tela_2”. Comece selecionando o componente “tela_2”, e no menu *Properties*, altere a opção *Center* (Centro) em *AlignHorizontal*, marque a opção *Scrollable*. Em *Title* coloque um título para esta tela do aplicativo, por exemplo “Sistemas de 2 Incógnitas”.

Agora monte na tela uma estrutura que simula um sistema linear onde o usuário deverá inserir os coeficientes e também onde o aplicativo exibirá as respostas. Abaixo segue a lista dos objetos que deverão ser adicionados na tela, os respectivos nomes e algumas edições nas

propriedades. Use a Figura 90 para nortear a tarefa.

- (1) Uma etiqueta, com o nome “etique_tatitulo” e a fase em *Text*: “INSIRA OS COEFICIENTES”;
- (2) Uma *TableArrangement*  *TableArrangement* (Arranjos em Tabela), disponível em *Palette/Layout*, renomeando para “caixa_sistema”. Deixe 5 em *Columns* (Colunas), 2 em *Rows* (Linhas), escreva 90 em *percent* na opção *Width* (Largura);
- (3) Um botão, de nome “botao_calcular”, e em *Text*, coloque “CALCULAR”;
- (4) Uma etiqueta, de nome “etiqueta_solucão”, apagando a frase em *Text*.
- (5) Outra *TableArrangement*, de nome “caixa_respostas”. Deixe 2 em *Columns*, 2 em *Rows*, e escreva 90 em *percent* na opção *Width*;
- (6) Outro botão com o nome “botao_voltar” com o *Text* “VOLTAR”.

Agora, nos componentes “caixa_sistema” e “caixa_respostas” (Figura 91), serão adicionados caixas de textos que receberão os coeficientes e as etiquetas que completarão os elementos de um sistema de equações lineares. Proceda:

- (1) Em “caixa_sistema”, na primeira linha adicione:
 - Uma caixa de texto com o nome “a”;
 - Uma etiqueta com o nome “etiqueta_1”, e no *Text* escreva “X + (“;
 - Um caixa de texto com o nome “b”;
 - Uma etiqueta com o nome “etiqueta_2”, e no *Text* coloque “) Y = ”;
 - Uma caixa de texto com o nome “c”.
- (2) Na segunda linha de “caixa_sistema” coloque:
 - Uma caixa de texto com o nome “d”;
 - Uma etiqueta com o nome “etiqueta_3”, com *Text* igual a “X + (“;
 - Um caixa de texto com o nome “e”;
 - Uma etiqueta com o nome “etiqueta_4”, escrevendo em *Text* a frase “) Y = ”;
 - Uma caixa de texto com o nome “f”.

No componente “caixa_respostas”, adicione na primeira linha duas etiquetas: “etiqueta_valorX”, com *Text* “Valor de X:” e “etiqueta_X”, sem nada escrito em *Text*. Na segunda linha, fica uma “etiqueta_valorY”, com *Text* “Valor de Y:” e “etiqueta_Y”, sem nada escrito em *Text*.

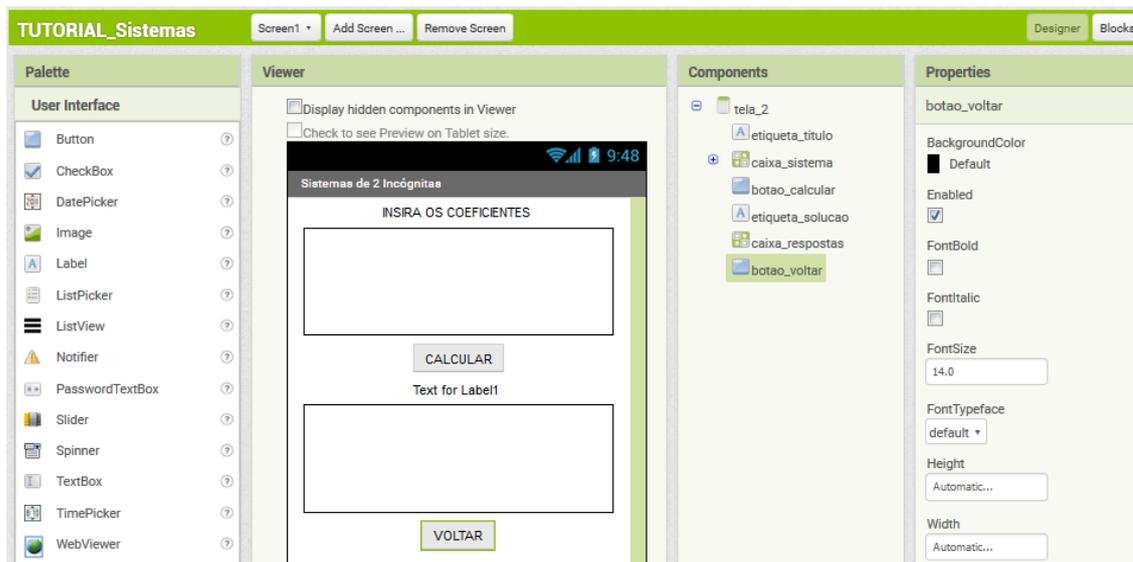


Figura 90 – Componentes inseridos na “tela_2”.

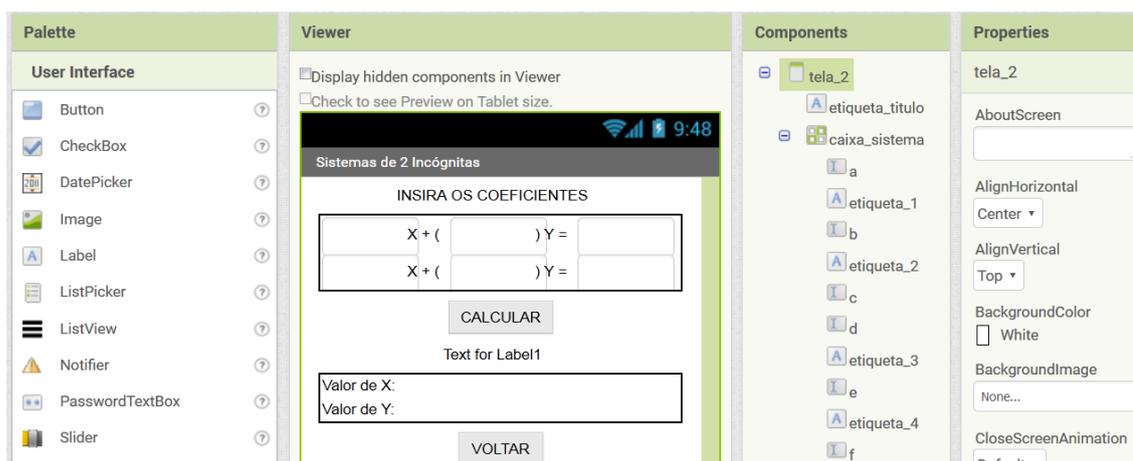


Figura 91 – Captura da “tela_2” com os novos componentes inseridos.

Antes de prosseguir, faça alguns ajustes para facilitar o manuseio do aplicativo durante os testes. Selecione cada caixa de texto e, na coluna *Properties*, vá em *Width*³², digite 20 em *percent*, retire a frase “*Hint for TextBox*” em *Hint* (Sugestão), pois ela é exibida dentro da caixa de texto quando o aplicativo é executado, atrapalhando a digitação, e marque a opção *NumbersOnly* (somente números) para dar agilidade na digitação dos valores nas caixas de textos.

³² Digitando o valor em *percent* (porcentagem) permite que as caixas de textos se adaptem a qualquer tamanho de tela. Por exemplo, na “tela_2”, o componente “caixa_sistema” ocupará 90% da largura, deixando um espaço nas margens da tela. As caixas de textos ocuparão 20% cada uma, e o restante dividido entre as etiquetas.

5.3.4 Esboço do cálculo para determinar as incógnitas e montagem dos blocos para sistemas lineares de duas incógnitas.

Agora serão montados os blocos que efetuarão os cálculos para determinarem os valores das duas incógnitas. De acordo com o Teorema de Cramer (**Teorema 5.1**), se o determinante da matriz formada pelos coeficientes for diferente de zero, então o sistema será possível e terá solução única. Caso o determinante seja igual a zero então o sistema pode ter infinitas soluções ou ser indeterminado. Portanto, a primeira tarefa que o aplicativo deverá fazer é verificar o valor do determinante.

É interessante o professor esboçar os cálculos para orientar os estudantes na montagem dos blocos. Assim, conforme a tela montada na seção anterior, o sistema de equações lineares é dado por $S: \begin{cases} ax + by = c \\ dx + ey = f \end{cases}$. Pelo teorema, o primeiro passo é calcular o valor do determinante, $\det(S)$, da matriz $S = \begin{bmatrix} a & b \\ d & e \end{bmatrix}$ formada pelos coeficientes do sistema linear S . Assim, $\det(S) = [(a \times e) - (b \times d)]$.

Suponha que $\det(S)$ seja igual a 0, então, o sistema linear S possui infinitas soluções ou é indeterminado. Logo não há mais nada a ser feito. Agora, sendo $\det(S)$ diferente de 0, o aplicativo deverá calcular dois novos determinantes. O segundo determinante é o da matriz $S_1 = \begin{bmatrix} c & b \\ f & e \end{bmatrix}$, formada pela substituição da primeira coluna (a coluna dos coeficientes da incógnita x) pela coluna de termos independentes, onde $\det(S_1) = [(c \times e) - (b \times f)]$. O terceiro determinante será da matriz $S_2 = \begin{bmatrix} a & c \\ d & f \end{bmatrix}$, onde foi substituído a segunda coluna (dos coeficientes da incógnita y) pela coluna de termos independentes, sendo $\det(S_2) = [(a \times f) - (c \times d)]$. Concluindo, $x = \frac{\det(S_1)}{\det(S)}$ e $y = \frac{\det(S_2)}{\det(S)}$.

Para montar os blocos de acordo com o raciocínio desenvolvido, clique no botão *Blocks* (Blocos)  no canto superior direito e será aberto o ambiente de edição dos blocos. A tarefa será iniciada com a construção dos blocos que calculam os valores de $\det(S) = [(a \times e) - (b \times d)]$, $\det(S_1) = [(c \times e) - (b \times f)]$ e $\det(S_2) = [(a \times f) - (c \times d)]$.

Usando $\det(S)$ como modelo, vá no submenu *Math*  **Math** (Matemática), escolha dois blocos de multiplicação  e um de subtração . Ainda no menu lateral, procure cada caixa de texto “a”, “b”, “d” e “e”, selecione o bloco *Text*  e faça a montagem ilustrada a seguir.



Figura 92 – Composição do cálculo de $\det(S)$.

Da mesma forma, construa os blocos para $\det(S_1)$ e $\det(S_2)$. Para facilitar os trabalhos, basta copiar e colar o bloco da Figura 92, substituindo os objetos das caixas verdes. Veja:

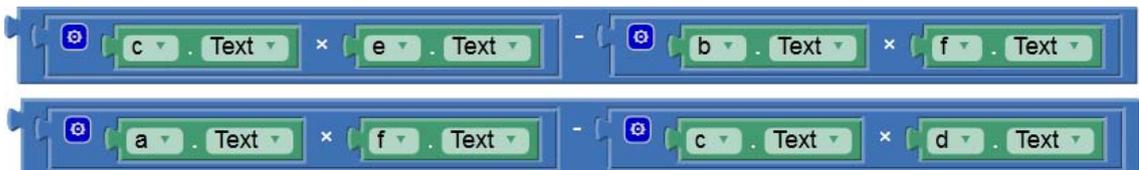
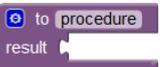


Figura 93 – Composição do cálculo de $\det(S_1)$ e $\det(S_2)$.

Uma estratégia para auxiliar no manuseio dos conjuntos, é usar o bloco³³  disponível no submenu *Procedures* (Procedimentos)  **Procedures**. Adicione três dele, e no lugar da palavra *procedure*, escreva “detS”, “detS1” e “detS2”, e acople neles os blocos da Figura 92 e Figura 93, da seguinte forma:

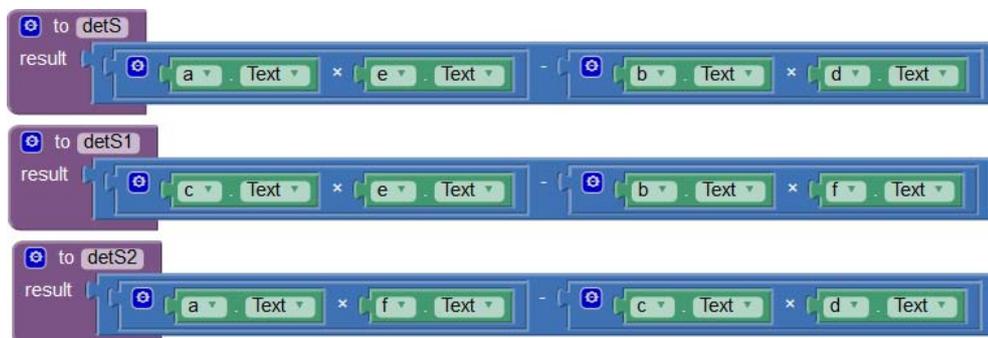


Figura 94 – Demonstração dos determinantes $\det(S)$, $\det(S_1)$ e $\det(S_2)$ transformados em procedimentos.

³³ Um procedimento é um conjunto de instruções que executam uma tarefa ou tarefas específicas. Ele também pode ser chamado de uma função. No MIT App Inventor é usado para substituir blocos que são acessados repetidamente e ocupam menos espaço do que todos os blocos utilizados no processo inicial.

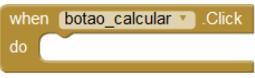
Prosseguindo, a primeira tarefa do aplicativo é verificar se o valor de $\det(S)$, representado agora por  (disponível também em *Procedure*), é igual ou diferente de zero. Esta verificação pode ser feita usando o bloco  e o bloco  ambos do submenu *Math*, chegando na seguinte composição: .

Se isso ocorrer, então o sistema de equações lineares não possui soluções ou possui infinitas soluções, finalizando a operação. O usuário precisa saber desta notícia, que deve ser exibido na tela do aplicativo. Isto pode ser feito utilizando o bloco  no submenu **Text**. Dentro dele escreva: “O sistema de equações lineares é indeterminado ou impossível.”, conforme **Teorema 5.1 e Definição 5.4**.

 “O sistema de equações lineares é indeterminado ou impossível.”

Figura 95 – Bloco que guarda a informação a ser exibida quando o $\det(S)$ é igual a zero.

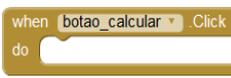
Agora, serão unidos todos os conjuntos construídos anteriormente para que a calculadora funcione plenamente. Todas estas operações devem ser iniciadas pelo “botao_calcular”.

Assim selecione a função *Click*  no submenu daquele componente.

A análise do $\det(S)$ é feita através do bloco  disponível no menu *Control* (Controle) **Control**, que precisará de mais um item *else* (senão). A inclusão de mais termos *else if* ou apenas *else* se dá clicando no botão azul no canto superior, e arrastando o item para dentro do bloco, conforme demonstrado na figura a seguir.



Figura 96 – Procedimento para adicionar itens *else*.

Feito isso acople o bloco dentro de  ficando assim:

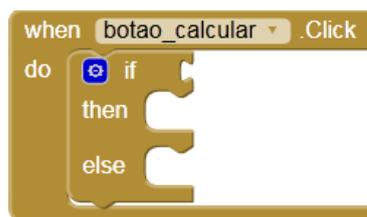
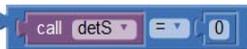


Figura 97 – Agrupamento do bloco *if/then* com o bloco *click* do “botao_calcular”.

Prosseguindo, pegue o conjunto  e coloque na abertura *if* (se). Fi-

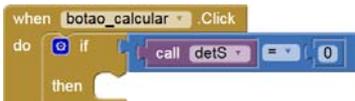
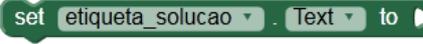
cando assim: . A leitura deste bloco pode ser feita da seguinte forma: “*when botao_calcular.clik do: if call detS = 0, then...*”, numa tradução literal seria: “quando o botao_calcular for clicado faça (no caso, uma verificação): se *detS* igual a zero, então...”. Tem-se aí a primeira hipótese que é $\det(S) = 0$. Sendo verdadeira, o sistema de equações lineares digitado pelo usuário é impossível ou possui infinitas soluções. Portanto, o aplicativo exibirá esta mensagem, ora guardada no bloco da Figura 95. Para que isso ocorra, escolha o componente “*etiqueta_solucao*” no menu lateral e procure pelo bloco . A seguir, encaixe o bloco lilás (Figura 95) na abertura *to* e coloque o bloco *set* de “*etiqueta_solucao*” na abertura *then* (então). Veja:



Figura 98 – Parte inicial da composição que soluciona os sistemas lineares.

A Figura 98 ilustra a composição da primeira hipótese da verificação do $\det(S)$ que é sendo este igual a zero. Prosseguindo, será construído os blocos para o caso de $\det(S)$ ser diferente de zero. Se isto ocorrer, então $x = \frac{\det(S_1)}{\det(S)}$ e $y = \frac{\det(S_2)}{\det(S)}$ sendo representados pelos

blocos  e  respectivamente. Os resultados destas divisões deverão aparecer em “*etiqueta_X*” e “*etiqueta_Y*”, da seguinte forma:



Agora junte estes dois últimos blocos e coloque na abertura *else* (senão), ficando assim:

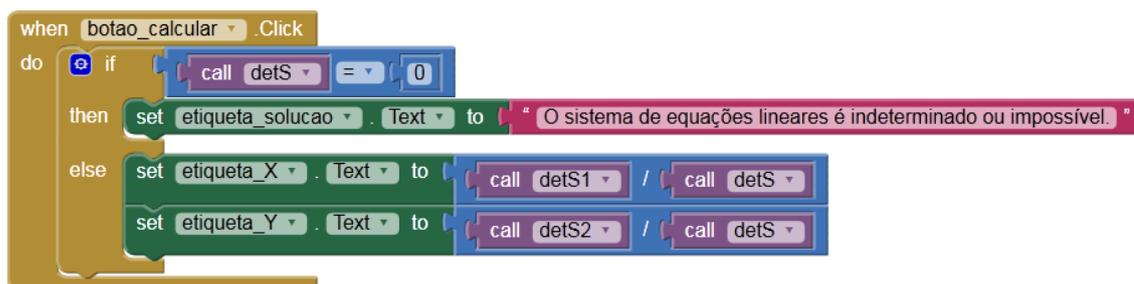
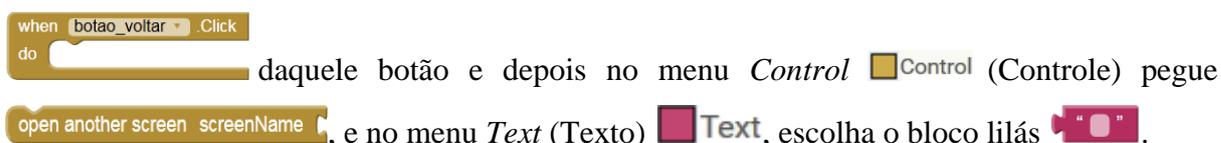


Figura 99 – Composição final dos cálculos para solução dos sistemas lineares de duas incógnitas.

Aproveite e faça a composição do “botao_voltar”. Adicione a função



Dentro do bloco lilás, deverá ser escrito o nome da tela que o botão deve abrir: *Screen1*. Conecte ele ao bloco “open another screen” (abrir outra tela), e associe ao primeiro bloco adicionado. Não pode haver diferença entre o nome dado à tela com o que foi escrito no bloco lilás.



Figura 100 – Bloco responsável por abrir a tela inicial “Screen1”.

Peça para os alunos testarem³⁴ os aplicativos com os exemplos abaixo para averiguar se há algum problema, corrigindo o que for necessário. Normalmente os problemas são oriundos de equívocos durante as montagens dos blocos, como a troca dos componentes, ou a escolha da função errada do item. O professor pode delegar estudantes com mais habilidades para acompanhar os demais auxiliando na monitoria do projeto.

Exemplo 5.3: $S: \begin{cases} 2x + 3y = 4 \\ x - y = 2 \end{cases}$. Solução: $x = 2$ e $y = 0$.

Exemplo 5.4: $S: \begin{cases} 2x + y = 2 \\ 4x + 2y = 4 \end{cases}$. Solução: sistema possível e indeterminado.

³⁴ Para testar o aplicativo, consulte o ANEXO A – O que instalar, onde está descrito detalhadamente as três formas de testes.

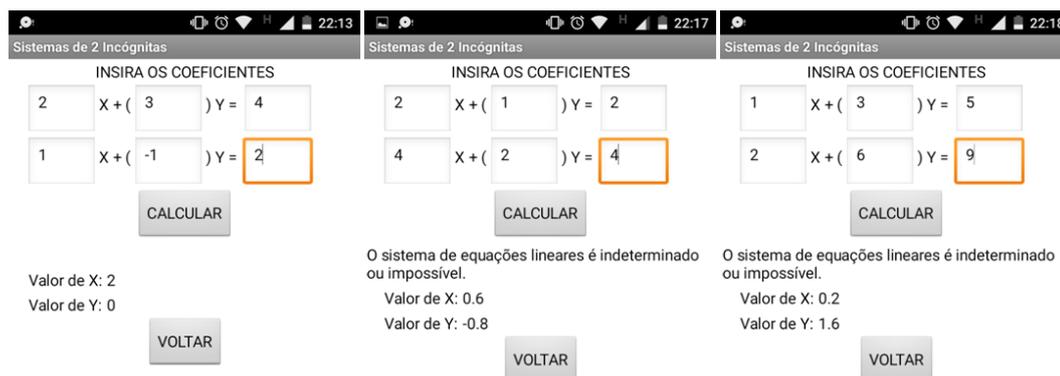


Figura 101 – Captura de tela do celular exibindo as soluções encontradas pelo aplicativo.

Veja que o aplicativo continua exibindo resultados encontrados em outros cálculos. Isso será resolvido mais à frente.

5.3.5 Esboço do cálculo para determinar as incógnitas para sistemas lineares de três incógnitas.

Uma vez construída a calculadora para sistemas lineares de duas equações e duas incógnitas, será iniciada a construção da calculadora para sistemas lineares de três incógnitas e três equações. O processo é análogo, e tal como foi feito anteriormente, é interessante que o grupo esboce os cálculos para orientar na montagem dos blocos do aplicativo.

Sendo o sistema de equações lineares $S: \begin{cases} ax + by + cy = d \\ ex + fy + gz = h \\ ix + jy + kz = l \end{cases}$, a forma matricial dos coeficientes será $S = \begin{bmatrix} a & b & c \\ e & f & g \\ i & j & k \end{bmatrix}$. Seguindo o mesmo roteiro, o primeiro passo é verificar se o

$\det(S)$ é igual ou diferente de zero. Aplicando a regra de Sarrus (**Definição 4.12**), o $\det(S) = [(a \times f \times k) + (b \times g \times i) + (c \times e \times j)] - [(c \times f \times i) + (a \times g \times j) + (b \times e \times k)]$.

O $\det(S_1)$, oriundo da matriz $S_1 = \begin{bmatrix} d & b & c \\ h & f & g \\ l & j & k \end{bmatrix}$ será $[(d \times f \times k) + (b \times g \times l) + (c \times h \times j)] -$

$[(c \times f \times l) + (d \times g \times j) + (b \times h \times k)]$. Seguindo, o $\det(S_2)$ da matriz $S_2 = \begin{bmatrix} a & d & c \\ e & h & g \\ i & l & k \end{bmatrix}$, é dado por

$[(a \times h \times k) + (d \times g \times i) + (c \times e \times l)] - [(c \times h \times i) + (a \times g \times l) + (d \times e \times k)]$. Finalmente, o $\det(S_3)$, segundo a matriz $S_3 = \begin{bmatrix} a & b & d \\ e & f & h \\ i & j & l \end{bmatrix}$ fica $[(a \times f \times l) + (b \times h \times i) + (d \times e \times j)] - [(d \times f \times i) + (a \times h \times j) + (b \times e \times l)]$.

Os valores das incógnitas são: $x = \frac{\det(S_1)}{\det(S)}$, $y = \frac{\det(S_2)}{\det(S)}$ e $z = \frac{\det(S_3)}{\det(S)}$.

5.3.6 Construção da tela da calculadora de que soluciona sistemas lineares de três incógnitas.

Dando prosseguimento, faça a montagem da “tela_3”, nos mesmos moldes feitos par a “tela_2”. Selecione o componente “tela_3”, e no menu *Properties*, altere a opção *Center* em *AlignHorizontal*, marque a opção *Scrollable*. Em *Title* coloque um título para esta tela do aplicativo, por exemplo “Sistemas de 3 Incógnitas”.

Agora monte na tela uma estrutura que simula um sistema linear onde o usuário deverá inserir os coeficientes e também onde o aplicativo exibirá as respostas. Abaixo segue a lista dos objetos que deverão ser incorporados na tela, os respectivos nomes e algumas edições nas propriedades:

- (1) Uma etiqueta, com o nome “etiqueta_titulo” e a fase em *Text*: “INSIRA OS COEFICIENTES”;
- (2) Uma *TableArrangement*, renomeando para “caixa_sistema”. Deixe 7 em *Columns*, 3 em *Rows*, escreva 90 em *percent* na opção *Width*;
- (3) Um botão, de nome “botao_calcular”, e em *Text*, coloque “CALCULAR”;
- (4) Uma etiqueta, de nome “etiqueta_soluciao”, apagando a frase em *Text*.
- (5) Outra *TableArrangement*, de nome “caixa_respostas”. Deixe 2 em *Columns*, 3 em *Rows*, e escreva 90 em *percent* na opção *Width*;
- (6) Outro botão com o nome “botao_voltar” com *Text* “VOLTAR”.

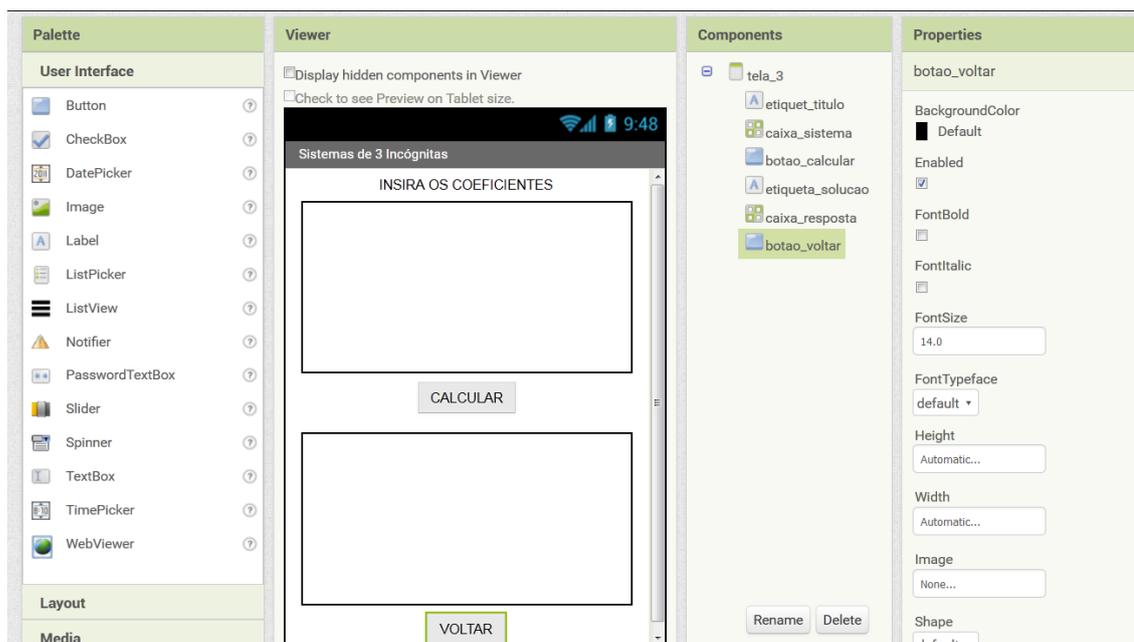


Figura 102 – Componentes inseridos na “tela_3”.

Agora, nos componentes “caixa_sistema” e “caixa_respostas”, serão adicionados caixas de textos que receberão os coeficientes e as etiquetas que completarão os elementos de um sistema de equações lineares. Vale ressaltar que inserindo três caixas de textos em cada *TableArrangement* a terceira não ficará visível tanto na coluna *Viewer* bem como na tela do smartphone. Por isso, como feito anteriormente, a cada caixa de texto adicionada, edite imediatamente a propriedade *Width*, colocando 13 em *percent*.

Assim faça:

(1) Em “caixa_sistema”, na primeira linha adicione:

- Uma caixa de texto com o nome “a”;
- Uma etiqueta com o nome “etiqueta_1”, e *Text* “X + (“;
- Um caixa de texto com o nome “b”;
- Uma etiqueta com o nome “etiqueta_2” , e *Text* “) Y + (“;
- Uma caixa de texto com o nome “c”;
- Uma etiqueta com o nome “etiqueta_3”, e *Text* “) Z = ”;
- Uma caixa de texto com o nome “d”.

(2) Na segunda linha de “caixa_sistema”:

- Uma caixa de texto com o nome “e”;

- Uma etiqueta com o nome “etiqueta_4”, e *Text* “X + (“;
- Um caixa de texto com o nome “f”;
- Uma etiqueta com o nome “etiqueta_5” , e *Text* “) Y + (“;
- Uma caixa de texto com o nome “g”;
- Uma etiqueta com o nome “etiqueta_6”, e *Text* “) Z = ”;
- Uma caixa de texto com o nome “h”.

(3) Na terceira linha de “caixa_sistema”:

- Uma caixa de texto com o nome “i”;
- Uma etiqueta com o nome “etiqueta_7”, e *Text* “X + (“;
- Um caixa de texto com o nome “j”;
- Uma etiqueta com o nome “etiqueta_8” , e *Text* “) Y + (“;
- Uma caixa de texto com o nome “k”;
- Uma etiqueta com o nome “etiqueta_9”, e *Text* “) Z = ”;
- Uma caixa de texto com o nome “l”.

Antes de prosseguir, selecione cada caixa de texto e, na coluna *Properties*, retire a frase “*Hint for TextBox*” em *Hint* e marque a opção *NumbersOnly*.

No componente “caixa_respostas”, adicione na primeira linha duas etiquetas: “etiqueta_valorX”, com *Text* “Valor de X:” e “etiqueta_X”, sem nada escrito em *Text*. Na segunda linha, coloque uma “etiqueta_valorY”, com *Text* “Valor de Y:” e “etiqueta_Y”, sem nada escrito em *Text*. E na terceira linha, fica uma “etiqueta_valorZ”, com *Text* “Valor de Z:” e “etiqueta_Z”, sem nada escrito em *Text*.

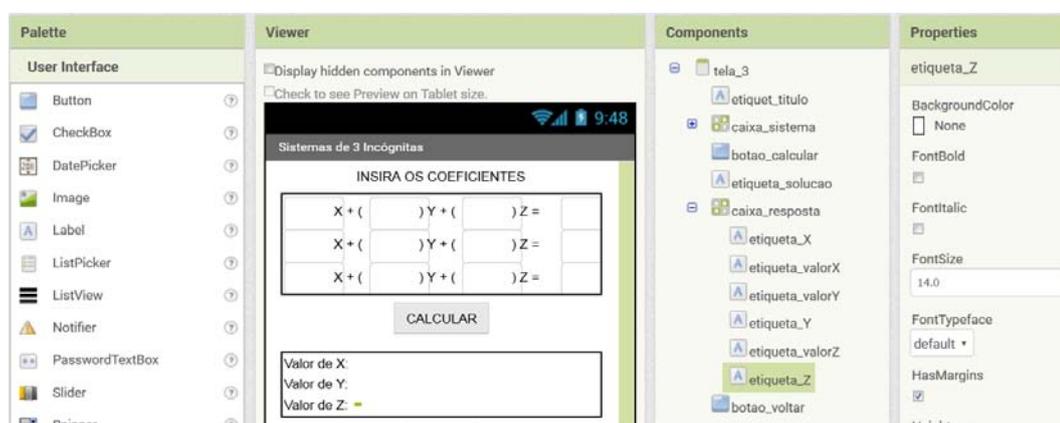
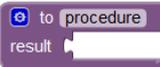


Figura 103 – Captura da “tela_3” com os novos componentes inseridos.

5.3.7 Montagem dos blocos da calculadora de que soluciona sistemas lineares de três incógnitas.

Como o processo é idêntico ao efetuado na tela anterior, você já pode começar criando

os blocos  para os determinantes de S , S_1 , S_2 e S_3 , associando os respectivos grupos que efetuarão os cálculos, conforme esboço feito. Veja:

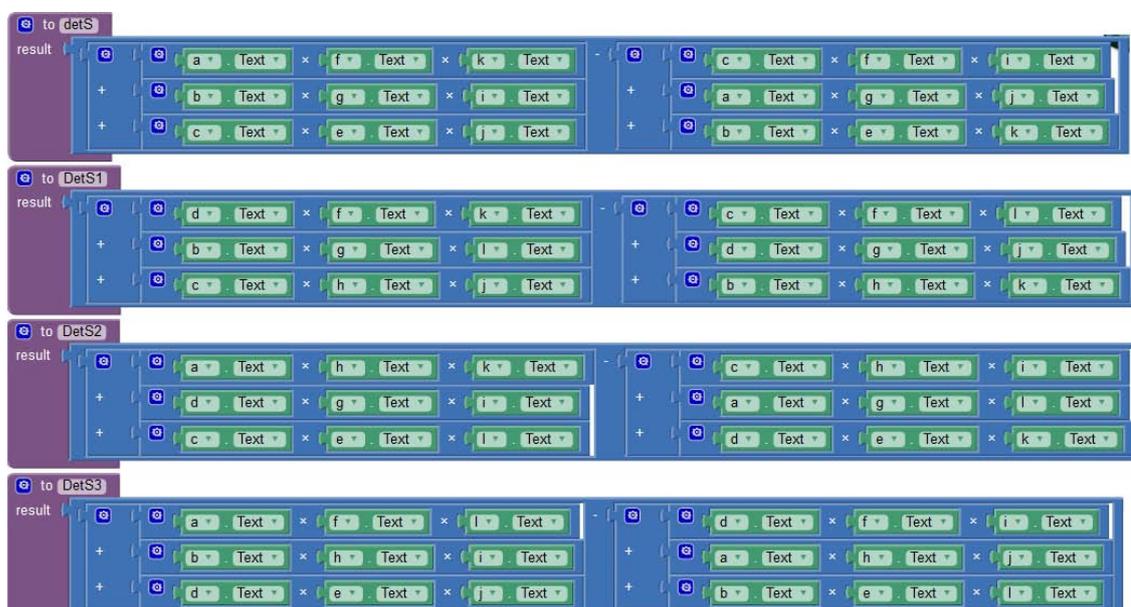


Figura 104 – Demonstração dos determinantes $\det(S)$, $\det(S_1)$, $\det(S_2)$ e $\det(S_3)$ transformados em procedimentos.

Para conseguir que os blocos das multiplicações fiquem empilhados, clique com o botão direito do mouse na parte azul de cada conjunto de produto, e escolha a opção *External Inputs*. Para desfazer, clique em *Inline Inputs*.

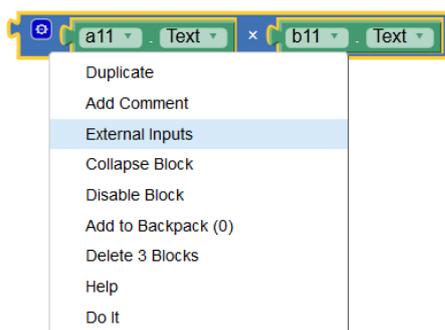


Figura 105 – Demonstração da função *External Inputs* que facilita o manuseio de blocos grandes.

Para finalizar, monte o bloco associado ao “botao_calcular”, lembrando de adicionar o cálculo para o valor da incógnita z , e também a função do “botao_voltar” (Figura 106) e teste o aplicativo com os exemplos abaixo.

Exemplo 5.4: $S: \begin{cases} x + 2y - z = 1 \\ 2x - y + 3z = 2 \\ x - 3y + 4z = 0 \end{cases}$. Solução: O sistema não possui solução, ou possui infinitas soluções, pois $\det(S) = 0$.

Exemplo 5.5: $S: \begin{cases} x + y + z = 6 \\ x - y - z = -4 \\ 2x - y + z = 1 \end{cases}$. Solução: $x = 1, y = 3$ e $z = 2$.

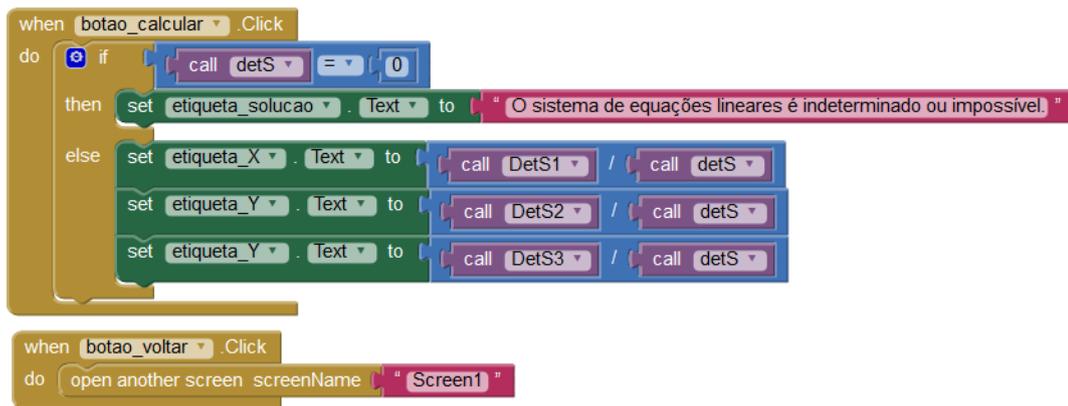


Figura 106 – Composição final dos cálculos para solução dos sistemas lineares de três incógnitas e a função do “botao_voltar”.

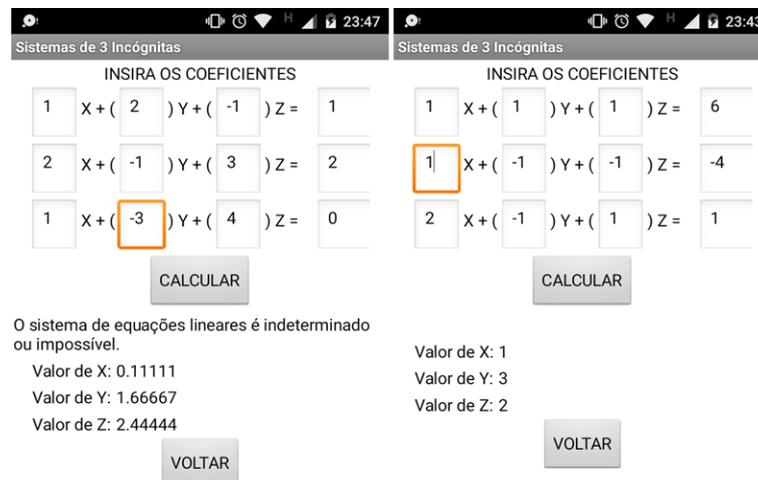


Figura 107 – Captura de tela do celular exibindo as soluções encontradas pelo aplicativo.

5.3.8 Finalização do projeto com modificações das propriedades dos componentes.

Veja que o projeto desenvolvido já consegue atingir os objetivos propostos, que são solucionar sistemas de equações lineares de duas e três incógnitas. Porém é necessário retornar em cada tela e efetuar ajustes nas propriedades de alguns componentes para que a aparência do aplicativo fique atraente. Além disso é preciso resolver o problema da permanência de informações de cálculos anteriores que continuam exibidos na calculadora, conforme Figura 101 e Figura 107.

Isto pode ser solucionado usando um botão para limpar todas as caixas de texto e todas as etiquetas que exibem os resultados a cada vez que for inserir novos dados. Para isso, adicione antes ou depois do “botao_voltar” um novo botão chamado “botao_limpar” com o *Text* “LIMPAR” na “tela_2” e “tela_3” (Figura 108), e no ambiente *Blocks* selecione a função

do “botao_limpar”, e também o bloco set de cada componente caixa de texto e etiqueta comentada anteriormente, agregando ao bloco `get plain start text` disponível no menu *Control*, conforme a composição ilustrada na Figura 109.

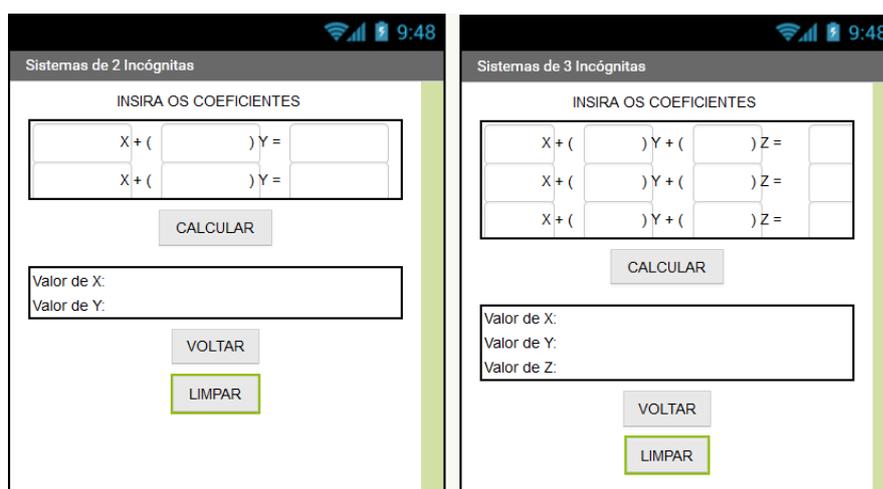


Figura 108 – Ilustração da inserção do “botao_limpar”.

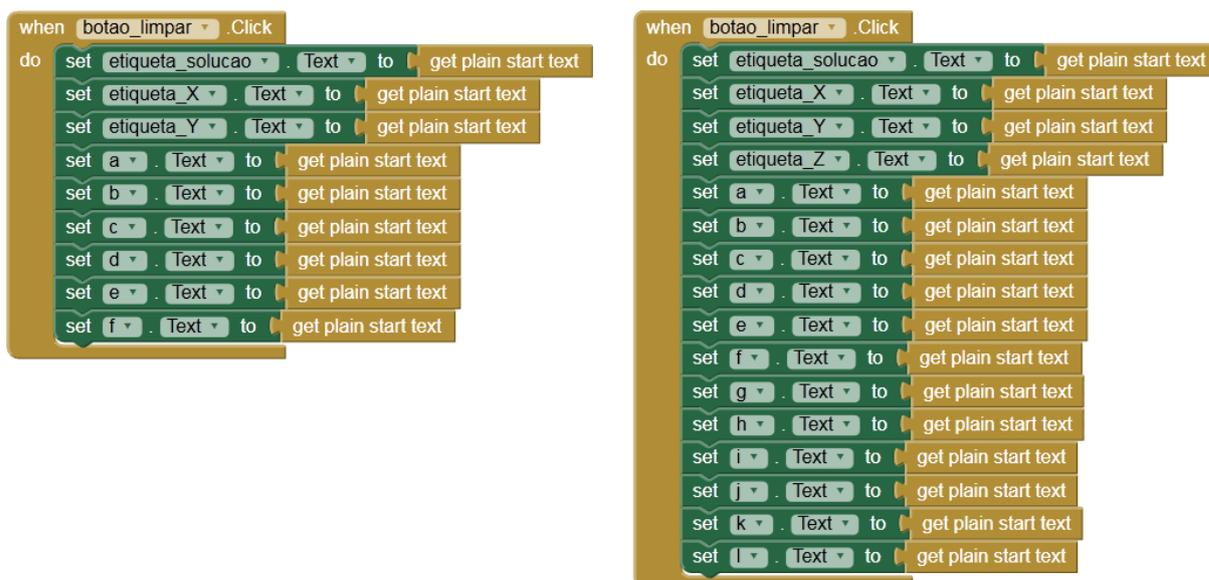


Figura 109 – Exemplos de blocos responsáveis por limpar as caixas de textos e etiquetas.

Feito isso, volte à tela “Screen1” e monte os blocos para atribuir a função correta ao “botao_2” e “botao_3” de modo que abram as respectivas telas. Conforme montagem similar feita anteriormente, a composição fica:



Figura 110 – Composições do “botao_2” e “botao_3”.

Com a finalização da construção dos blocos, faça as modificações que achar melhor para deixar a aparência personalizada. Como sugestão, as modificações usadas neste tutorial foram:

- (1) Nas “etiqueta_titulo” de cada tela, faça:
 - Marque *FontBold* (Negrito);
 - Coloque 20 para *FontSize* (Tamanho da Fonte);
 - Selecione *Center* em *TextAlignment*;
 - Altere o *TextColor* (Cor do Texto) para vermelho.

- (2) Em todas caixas de texto:

- Marque *FontBold*;
- Verifique se em *Width*, está digitado 20 em *percent*, na “tela_2” e 13 em *percent* na “tela_3”.
- Verifique se foram retiradas todas as frases “*Hint for TextBox*” em *Hint*;
- Confira se foi marcado a opção *NumbersOnly*;
- Selecione *Center* em *TextAlignment*.

(3) Nos botões:

- Escolha uma cor em *BackgroundColor* (Cor de Fundo);
- Marque *FontBold*;
- Coloque 20 para *FontSize*;
- Em *Shape* (Forma), escolha *rounded*;
- Selecione *Center* em *TextAlignment*.

(4) Nas etiquetas da “etiqueta_X” e “etiqueta_Y”:

- Marque *FontBold*;
- *FontSize* para 20;
- Selecione *Center* em *TextAlignment*;
- Mude o *TextColor* para vermelho.

(4) Nas etiquetas da “etiqueta_valorx” e “etiqueta_valory” e “etiqueta_valorz”:

- Marque *FontBold*;
- *FontSize* para 20;
- Selecione *Center* em *TextAlignment*.

(4) Nas etiquetas da “etiqueta_1”, ..., e “etiqueta_9”, marque *FontBold*.

(5) Escolha uma figura bonita na internet e adicione através do botão *UploadFile*  na aba *Média*. Depois selecione o componente “Screen1”, e vá e *BackgroundImage* (Imagem de Fundo), selecionando o arquivo que foi inserido.



Figura 111 – Captura das telas do aplicativo exibindo as modificações aplicadas.

5.3.9 Desafio.

Para concluir esta sequência didática, proponha à turma uma tarefa em forma de desafio, que pode ser feita na própria escola ou como trabalho extra-classe, onde os alunos deverão usar os novos conhecimentos adquiridos em sistemas de equações lineares bem como na plataforma do MAI.

Note que o aplicativo só determina as soluções dos sistemas lineares quando o determinante da matriz formada pelos coeficientes for diferente de zero. Todavia, para sistemas lineares de duas incógnitas e duas equações, é possível descobrir se, mesmo não tendo solução, ele é possível e indeterminado ou se ele é impossível.

A tarefa consiste no seguinte objetivo: os alunos deverão voltar ao aplicativo desenvolvido anteriormente, e na “tela_2” incluir um método que permite o aplicativo informar ao usuário se o sistema de duas incógnitas, é do tipo *SPD*, *SPI* ou *SI* (conforme **Definição 5.4**).

Uma dica importante para ser dada aos alunos é que eles precisam manipular a equação $[(a \times e) - (b \times d)] \neq 0$ que calcula o *det* (*S*).

6 CONSIDERAÇÕES FINAIS

Ao usar a plataforma do MIT App Inventor 2 como estratégia para aprimoramento da prática docente nas aulas de Matemática, revelou-se um novo mundo com inúmeras opções para trabalhar os conteúdos matemáticos, superando as expectativas planejadas inicialmente. Isso se deve ao fato do MAI oferecer um ambiente fértil para sustentar a criatividade dos alunos, tão quão as dos professores, através de uma ferramenta compatível com as mudanças tecnológicas e sociais ocorridas nos últimos anos, que é o acesso à rede de informações e de aparelhos eletrônicos de alta performance.

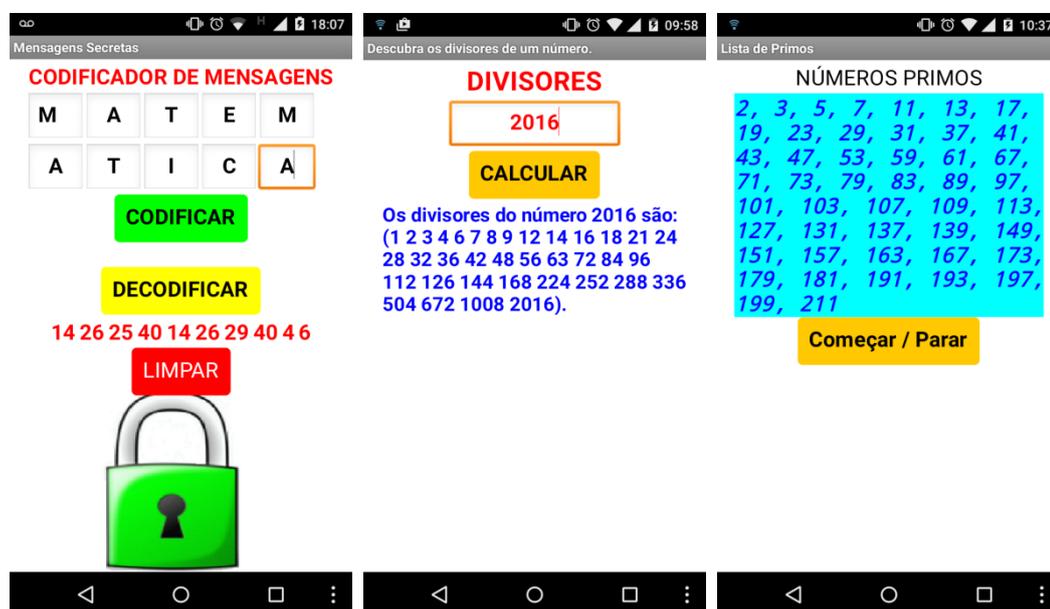


Figura 112 – Exemplos de aplicativos desenvolvidos paralelamente ao projeto.

Desta percepção, dois pontos podem ser destacados. O primeiro está atrelado à disciplina de Matemática e ao processo de ensino-aprendizagem, ora objetivos principais deste trabalho. Na Figura 112, estão ilustrados outros aplicativos que foram desenvolvidos paralelamente no decorrer desta dissertação, demonstrando assim, que à medida que as atividades vão

se desdobrando, oportunidades surgem, ampliando assim o leque de opções e materiais que o professor acumula durante as atividades em sala de aula. Na verdade, não apenas a coleção de aplicativos (considera-se os aplicativos não como ápice da aula desenvolvida pelo professor, mas sim como caminho/estratégia para colaborar com sua prática docente) aumentam, como também as chances de abranger o maior número possível de alunos. Por exemplo, alguns alunos ficam mais a vontade com matrizes, outros alunos se dedicam aos estudos de estatística, e outro grupo demonstram interesse com a Geometria.

Assim, com um portfólio maior de projetos, o professor incentiva os alunos a estimularem suas habilidades, produzindo conhecimento por meio das práticas e do pensamento crítico e criativo. No caso em tela, fixando-se no ambiente do MAI, projetos realizados em grupos permitem envolver aptidões diferentes de cada integrante. Veja um exemplo de cadeia de ações:

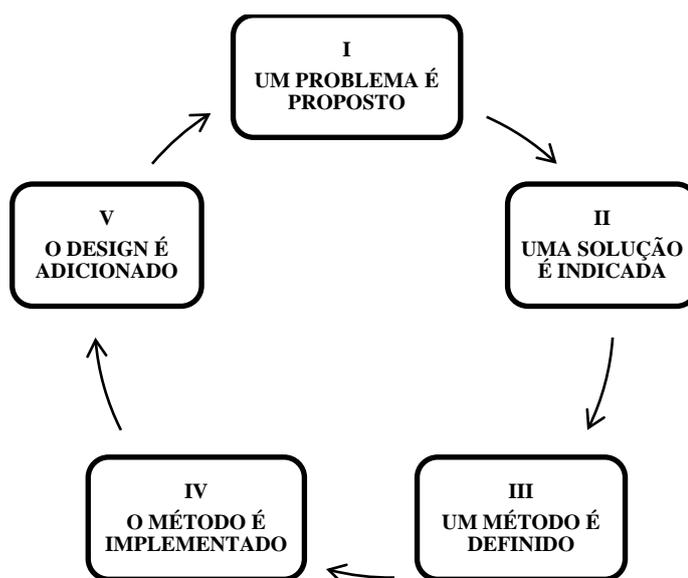


Figura 113 – Cadeia de tarefas para serem realizadas em grupo durante o trabalho com o MAI.

No momento (I), através da observação do mundo que o cerca, questionando determinados padrões através das perguntas “e quando isto acontecer, o que pode ser feito?” ou ainda “existe outra forma de fazer?”, gera o problema a ser resolvido. Para que (II) aconteça, várias soluções são levantadas, e ao escolher a mais viável, ocorre o (III) onde o método para resolver é construído. A fase (IV) é o momento em que o procedimento sugerido é transformado na linguagem de programação, e finalmente em (V) é aplicado o acabamento do design do aplicativo. Portanto momentos diferentes permitem que habilidades diferentes colaborem para a execução do projeto, respeitando a habilidade de cada indivíduo no grupo, abrindo espaço pa-

ra expressar as opiniões enriquecendo a discussão.

O segundo ponto, e talvez mais nobre que o anterior, é perceber que o MAI não está vinculado somente ao ensino de Informática ou Matemática. Pelo contrário, é possível desenvolver atividades multidisciplinares apoiados na plataforma do MIT App Inventor, atendendo a uma necessidade específica e momentânea para os diversos projetos na unidade de ensino, permitindo assim uma aproximação com a educação ideal com o fluxo de informações em todas as áreas do conhecimento.



Figura 114 – Exemplos de aplicativos que podem ser trabalhados na escola.

Na figura acima está ilustrada a tela de três aplicativos desenvolvidos por este autor objetivando comprovar estas ricas relações entre a Matemática, a plataforma MIT App Inventor e as demais disciplinas. Na primeira imagem, está um programa que acompanha o Índice de Massa Corpórea que pode ser usado nas aulas de Educação Física e Ciências. Neste mesmo sentido, a segunda imagem contém um aplicativo que monitora os alimentos servidos na escola de acordo com a quantidade diária de calorias por criança, usando para isto os conhecimentos de sistemas lineares. Na última imagem, uma versão eletrônica da competição de soletração para ser usado nas aulas de Língua Portuguesa.

Portanto, fica o convite ao professor que fez uso deste material em divulgar e fomentar o uso da plataforma MIT App Inventor 2 como ferramenta de ensino e democratização do acesso à tecnologia em sua escola.

7 REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, Maria Elizabeth B. **Informática e formação de professores**. Coleção Informática Aplicada na Educação. São Paulo: MEC/SEED/PROInfo, 1999. Disponível em: <http://www.dominiopublico.gov.br/pesquisa/DetalheObraForm.do?select_action=&co_obra=40244>. Acesso em: 09 de setembro de 2015.

ALTOÉ, Anair; FUGIMOTO, Sonia Maria A. **Computador na educação e os desafios educacionais**. Artigo. IX Congresso Nacional de Educação. III Encontro Sul Brasileiro de Psicopedagogia. Curitiba-PR. 2009. Disponível em: <www.pucpr.br/eventos/educere/educere2009/anais/pdf/1919_1044.pdf>. Acesso em 09 de setembro de 2015.

ANDROID. In: WIKIPÉDIA: a enciclopédia livre. Disponível em: <<https://pt.wikipedia.org/wiki/Android>>. Acesso em: 24 de Novembro de 2015.

BITTENCOURT, Maria Aparecida L. et al. **Normas técnicas para elaboração de trabalhos acadêmicos**. Ilhéus: Editus, 2010, 91 p. Disponível em: <<http://nead.uesc.br/arquivos/Fisica/tcc1/manual-de-normas.UESC.pdf>>. Acessado em: 01 de Setembro de 2015.

BOLDRINI, J. L.; COSTA, S.I.R.; RIBEIRO, V. L.; WETZLER, H.G. **Álgebra Linear**. 5 ed. São Paulo: Harper-Row, 1980.

BRASIL. Secretaria de Educação Fundamental. **Parâmetros curriculares nacionais: terceiro e quarto ciclos do ensino fundamental. Matemática**. Brasília: MEC/SEF, 1998. 148 p. Disponível em: <<http://portal.mec.gov.br/seb/arquivos/pdf/matematica.pdf>>. Acesso em: 24 de Novembro de 2015.

CALLIOLI, Carlos A.; DOMINGUES, Hygino H, COSTA; Roberto C. F. **Álgebra linear e aplicações**. 6 ed. reform. São Paulo: Atual, 1990.

Código aberto. In: WIKIPÉDIA: a enciclopédia livre. Disponível em: <https://pt.wikipedia.org/wiki/C%C3%B3digo_aberto>. Acesso em: 21 de Agosto de 2015.

Feature Phone. In: WIKIPÉDIA: a enciclopédia livre. Disponível em: <https://en.wikipedia.org/wiki/Feature_phone>. Acesso em: 21 de Agosto de 2015.

FILHO, Benigno B.; SILVA, Claudio X. **Matemática.** Vol. Único. São Paulo: FTD. 2004.

FREIRE, Fernanda M. P. **A diversidade de descrições de um mesmo problema: relações entre a linguagem Logo e a língua do usuário.** Artigo. Campinas, SP:UNICAMP/NIED, 1999. Disponível em <http://www.nied.unicamp.br/oea/mat/Logo_linguagem_nanda.pdf> Acesso em 01 de setembro de 2015.

FUNARO, Vânia M. B. O. et al. **Diretrizes para apresentação de dissertações e teses da USP: documento eletrônico e impresso Parte I (ABNT).** 2 ed. rev. ampl. São Paulo: Sistema Integrado de Bibliotecas da USP, 2009, 102 p. Disponível em: <http://www.theses.usp.br/index.php?option=com_content&view=article&id=52&Itemid=67>. Acesso em 01 de setembro de 2015.

FUNDAÇÃO ITAÚ SOCIAL. **O Impacto da Educação nos salários dos jovens brasileiros.** Disponível em: <<http://www.redeitausocialdeavaliacao.org.br/blog/o-impacto-da-educacao-nos-salarios-dos-jovens-brasileiros/>>. Acesso em: 24 de Novembro de 2015.

HEFEZ, Abramo; FERNANDEZ, Cecília de Souza. **Introdução à Álgebra Linear.** Rio de Janeiro: SBM, 2012.

IEZZI, Gelson; et al. **Fundamentos de matemática elementar:** sequências matrizes determinantes sistemas, volume 4, 2 ed. São Paulo: Atual, 1977.

IEZZI, Gelson. et al. **Matemática.** Vol. Único. São Paulo: Atual. 2002.

INSTITUTO TIM. O Círculo da Matemática do Brasil. **Como é a matemática dos adultos?** Disponível em <<http://institutotim.org.br/2015/11/06/como-e-a-matematica-dos-adultos/>>. Acessado em 24 de Novembro de 2015.

International Data Corporation Pesquisa de Mercado e Consultoria Ltda. **Estudo da IDC Brasil aponta que, em 2014, brasileiros compraram cerca de 104 smartphones por minuto.** São Paulo. Disponível em: <<http://br.idclatin.com/releases/news.aspx?id=1801>>. Acesso em: 14 de maio de 2015.

_____. **Estudo da IDC Brasil registra alta de 13% no mercado brasileiro de tablets, em 2014.** São Paulo. Disponível em: <<http://br.idclatin.com/releases/news.aspx?id=1785>>. Aces-

so em: 21 de Agosto de 2015.

IPEA. Aumento da escolaridade amplia renda do trabalho. Disponível em: <http://www.ipea.gov.br/portal/index.php?option=com_content&view=article&id=20067>. Acessado em 24 de Novembro de 2015.

LIMA, Elon L. et al. **A Matemática do Ensino Médio. Vol. I.** 10 ed. redig. Revis. Rio de Janeiro: SBM, 2012.

Mensagens Secretas com Matrizes. Recursos educacionais multimídia para a matemática do ensino médio. Matemática Multimídia. Unicamp. Disponível em: <<http://m3.ime.unicamp.br/recursos/1020>>Acessado em 22 de Novembro de 2015.

Mindstorms: Children, Computers and Powerful Ideas. In: WIKIPÉDIA: a enciclopédia livre. Disponível em: <https://pt.wikipedia.org/wiki/Mindstorms:_Children,_Computers_and_Powerful_Ideas>. Acesso em: 31 de Agosto de 2015.

MIRANDA, Mylene R. et al. **Matriz e Criptografia: há alguma relação?** Portal do Professor. Disponível em: <<http://portaldoprofessor.mec.gov.br/fichaTecnicaAula.html?aula=18475>>. Acessado em 22 de Novembro de 2015.

MIT App Inventor. Disponível em: <<http://appinventor.mit.edu/explore/index-2.html>>. Acessado em 03 de Janeiro de 2016.

MULTIMÉDIA. In: WIKIPÉDIA: a enciclopédia livre. Disponível em: <<https://pt.wikipedia.org/wiki/Multim%C3%A9dia>>. Acesso em: 24 de Novembro de 2015.

Oficina 2 de App Inventor. IF-SC São José. In: WIKIPÉDIA: a enciclopédia livre. Disponível em: <http://wiki.sj.ifsc.edu.br/wiki/index.php/Oficina_2_de_App_Inventor>. Acesso em: 16 de Novembro de 2015.

PAPERT, Seymour. **Mindstorms: children, computers, and powerful ideas.** Massachusetts: Basic Books, 1980.

PAIVA, Manoel. **Matemática.** Vol. II. São Paulo: Moderna, 1995.

TECHNOVATION CHALLENGE. **Introdução ao MIT App Inventor.** Disponível em: <http://www.technovationchallenge.org/wp-content/uploads/2015/01/tutorial-construcao_app-FaleComigo-pt.pdf>. Acessado em 02 de Dezembro de 2015.

TODOS PELA EDUCAÇÃO. **Indicadores da Educação**. Disponível em: <http://www.todospelaeducacao.org.br/indicadores-da-educacao/5-metas?task=indicador_educacao&id_indicador=15#filtros>. Acessado em 24 de Novembro de 2015.

VALENTE, José. A. **O Computador na Sociedade do Conhecimento**. In: Valente (org). Campinas, SP: UNICAMP/NIED, 1999. Disponível em: <<http://www.nied.unicamp.br/oea/pub/livro1/>>. Acesso em 26 de Agosto de 2015.

VALENTE, José. A.; ALMEIDA, Fernando. J. **Visão analítica da informática na educação no Brasil: a questão da formação do professor**. Disponível em: <<http://www.professores.uff.br/hjbortol/car/library/valente.html>>. Acesso em 13 de maio de 2015.

8 ANEXOS

ANEXO A – O QUE INSTALAR.

Para usufruir o MIT App Inventor 2, o seu computador/notebook e o dispositivo móvel, smartphone/tablet, deverão ter um dos seguintes requisitos básicos³⁵:

- Computador: Macintosh (com processador Intel) tipo Mac OS X 10.5 ou superior³⁶;
- Sistema Operacional: Windows (versões Windows XP, Windows Vista, Windows 7) ou GNU / Linux³⁷ (versões Ubuntu 8 ou superior, Debian 5 ou superior);
- Navegadores³⁸: Mozilla Firefox³⁹ 3.6, Apple Safari 5.0 e Google Chrome 4.0 (ou as versões superiores);
- Dispositivos móveis: sistema operacional Android 2.3 ("Gingerbread") ou superior.

O próximo passo é configurar uma (ou também as três) forma de testar o projeto que você está desenvolvendo. Acesse o endereço⁴⁰ <<http://appinventor.mit.edu/explore/ai2/setup.html>> e lá serão exibidos as opções conforme figura a seguir.

³⁵ Para saber mais, acesse: < <http://appinventor.mit.edu/explore/content/system-requirements.html>>. Acessado em 09 de Dezembro de 2015.

³⁶ A plataforma sugere o uso de computadores Macintosh, mas foi constatado que computadores que possuem 512 Mb de memória RAM, processador de 1,0 Gb, 80 Gb de armazenamento e conexão USB é suficiente para desenvolver os projetos.

³⁷ GNU / Linux desenvolvimento ao vivo é suportado apenas para conexões Wi-Fi entre o computador e o dispositivo Android.

³⁸ Vale lembrar que o Microsoft Internet Explorer não é suportado.

³⁹ Se você estiver usando o Firefox com a extensão NoScript, você precisa deixar a extensão off.

⁴⁰ Acessado em 09 de Dezembro de 2015.

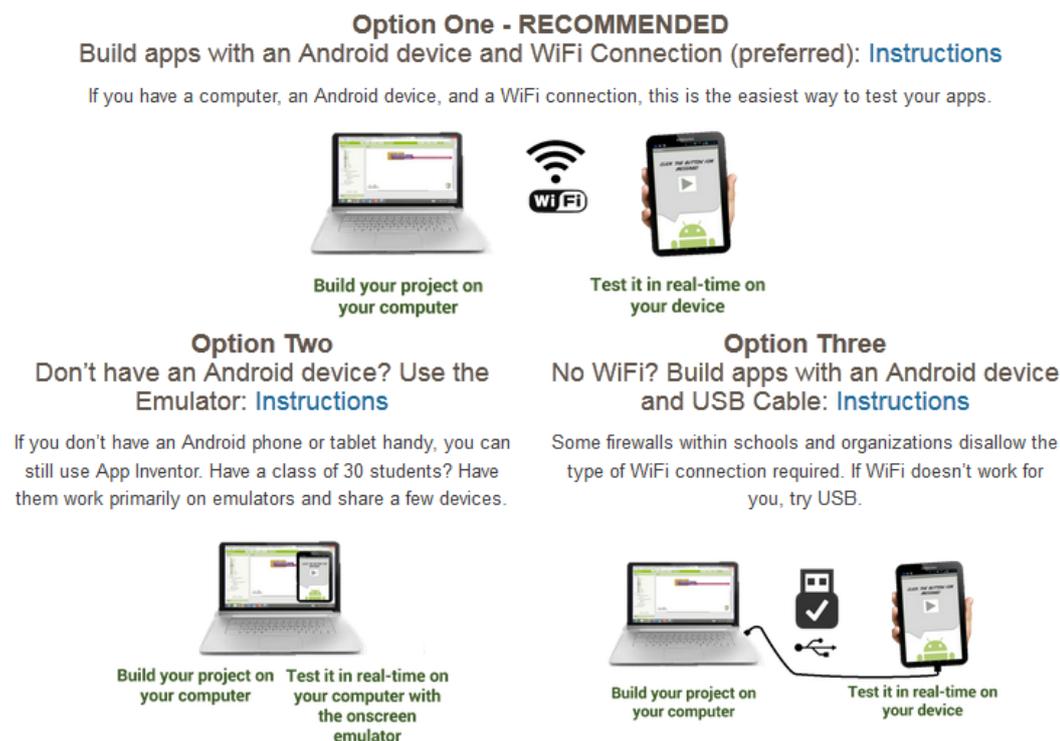


Figura 115 – Relação das três opções para testar os projetos em desenvolvimento.

A primeira opção para testar os aplicativos, que é a recomendada pela plataforma, e também a mais fácil de operar é aquela que se conecta através do wi-fi. Porém esta alternativa fica à mercê da oferta de redes sem fio na unidade escolar. Se isso não for problema, basta apenas instalar o aplicativo desenvolvido pelo MAI para testar projetos diretamente no dispositivo móvel, conforme indicado na Figura 116.

Acesse o endereço⁴¹ <<http://appinventor.mit.edu/explore/ai2/setup-device-wifi.html>> e lá haverá os links para que o aplicativo possa ser baixado em seu aparelho. Selecionando o link do Play Store, você será conduzido à loja de aplicativos do Android. Lembre-se de que precisa de conta no Google para fazer a instalação no seu aparelho móvel.

⁴¹ Acessado em 09 de Dezembro de 2015.

Connect your Phone or Tablet over WiFi

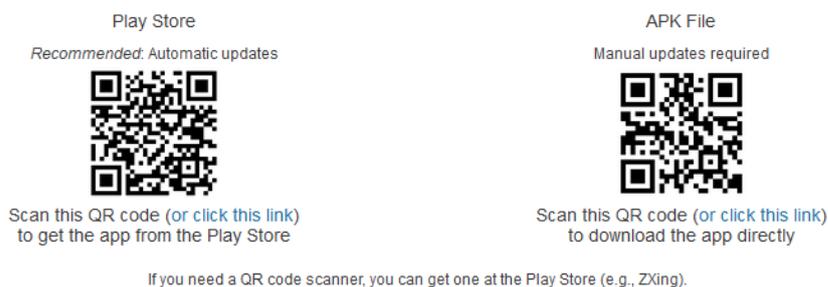
You can use App Inventor without downloading anything to your computer! You'll develop apps on our website: ai2.appinventor.mit.edu. To do live testing on your Android device just install the MIT App Inventor Companion app on your Android phone or tablet. Once the Companion is installed, you can open projects in [App Inventor on the web](#), open the companion on your device, and you can test your apps as you build them:



The following steps will walk you through the process.

Step 1: Download and install the MIT AI2 Companion App on your phone.

Open your device's QR code scanner and scan the QR code on the left below to download the Companion App from the Play Store. If you can't use the Play Store, use the QR code on the right to download the Companion App directly to your phone.



If you need a QR code scanner, you can get one at the Play Store (e.g., ZXing).

Figura 116 – Página que disponibiliza os links para a instalação do aplicativo do MAI que executa o projeto em desenvolvimento no smartphone/tablet.

Usando este método, quando estiver na plataforma de desenvolvimento, deverá escolher no menu *Connect* (Conectar), a opção *All Companion* (Figura 117), que gerará um código de barras do tipo QRCode, e este será lido pelo seu aparelho através do aplicativo instalado (Figura 118), conforme ilustrado na Figura 119.

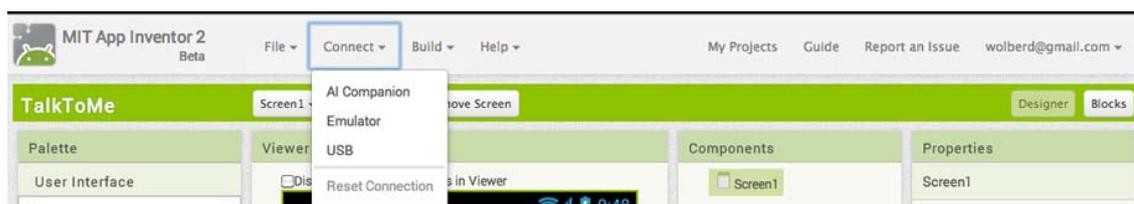


Figura 117 – Demonstração da escolha *All Companion* para testar o projeto através do wi-fi.

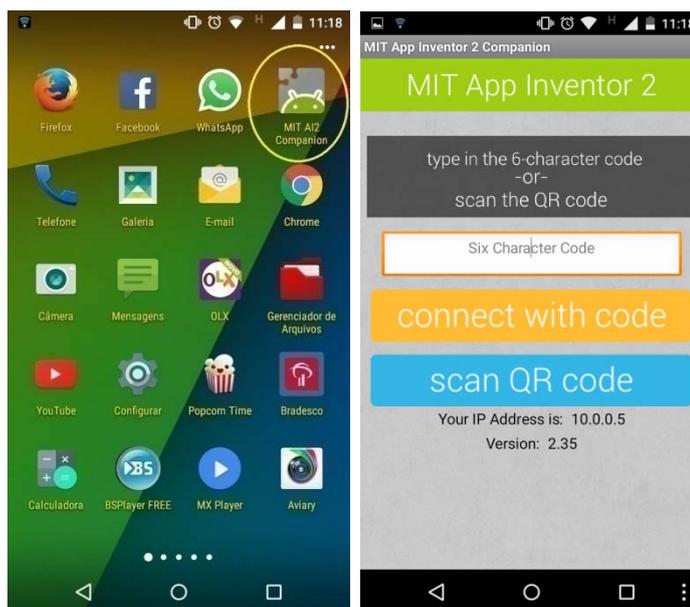


Figura 118 – Capturas de telas do smartphone demonstrando o programa instalado e em execução.

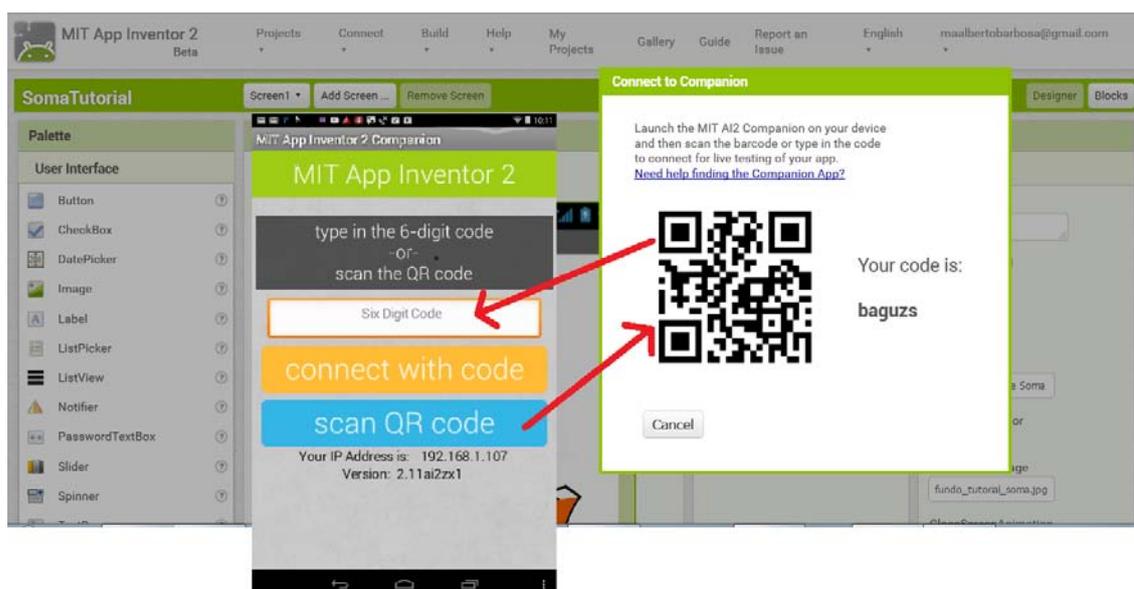


Figura 119 – Código gerado pela plataforma para que o dispositivo baixe e execute o aplicativo em desenvolvimento.

Se o laboratório de informática de sua escola não possuir acesso amplo à rede *wi-fi*, é possível testar os projetos com um emulador (*option two*) ou diretamente no smartphone/tablet através de um cabo USB (*option three*) que cada aluno possui, eis que é um acessório que acompanha os aparelhos.

Escolhendo a opção do emulador, clique na figura correspondente (Figura 115), e na

página ⁴² seguinte <<http://appinventor.mit.edu/explore/ai2/setup-emulator.html>> aparecerá uma lista com os três sistemas operacionais suportados. Escolha o que sirva a sua realidade, faça o download do arquivo e siga as instruções de instalação. Apesar de estar em inglês é intuitivo e não foge do padrão de qualquer instalação típica.

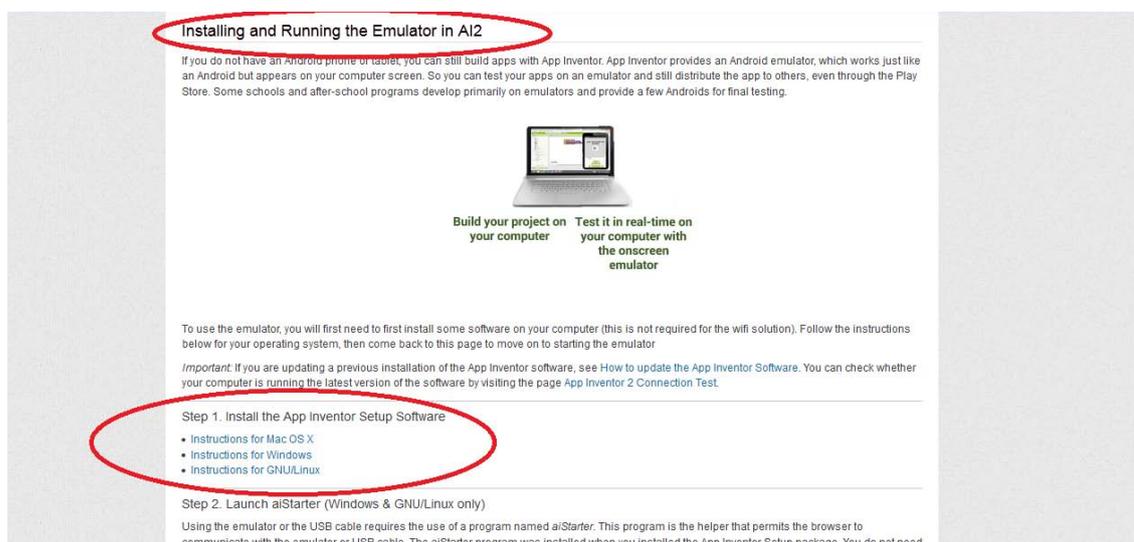


Figura 120 – Página que disponibiliza os links para a instalação do emulador no computador.

Para usar o emulador, deverá escolher no menu *Connect*, a opção *Emulator* (Figura 121), que por sua vez executará o programa instalado no seu computador simulando a tela de um celular (Figura 122 e Figura 123).

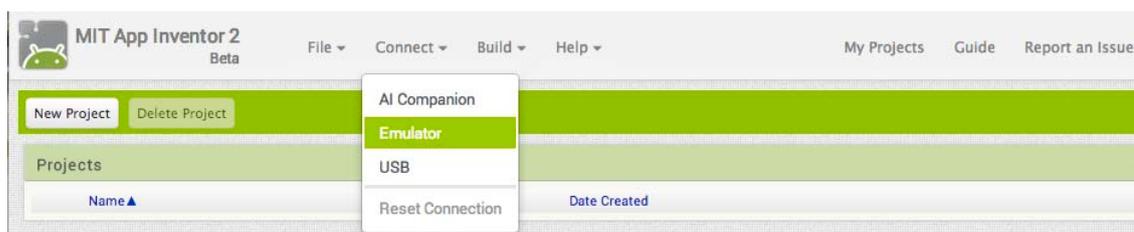


Figura 121 – Demonstração da escolha *Emulator* para testar o projeto.

⁴² Acesso em 09 de Dezembro de 2015.

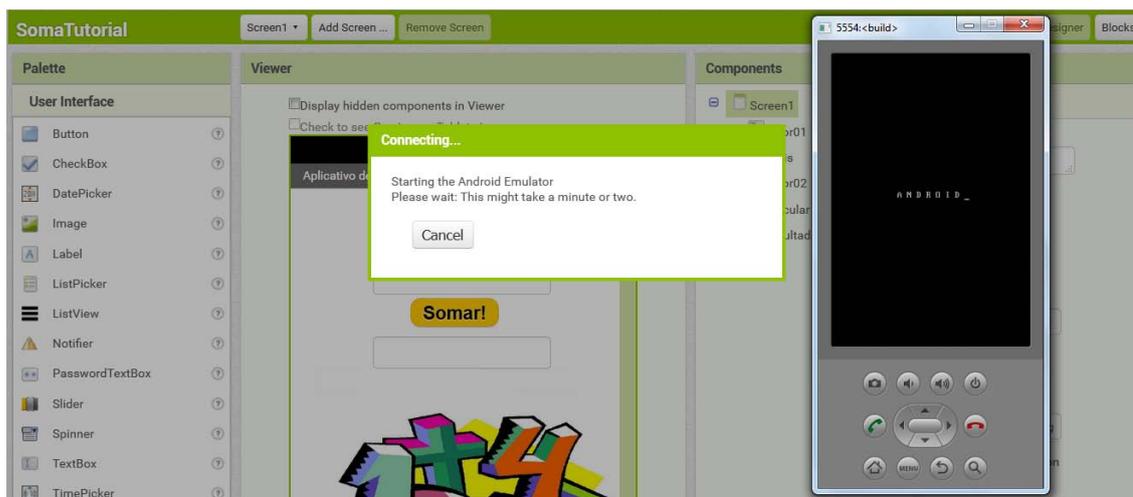


Figura 122 – Ilustração da plataforma iniciando a execução do emulador no computador.

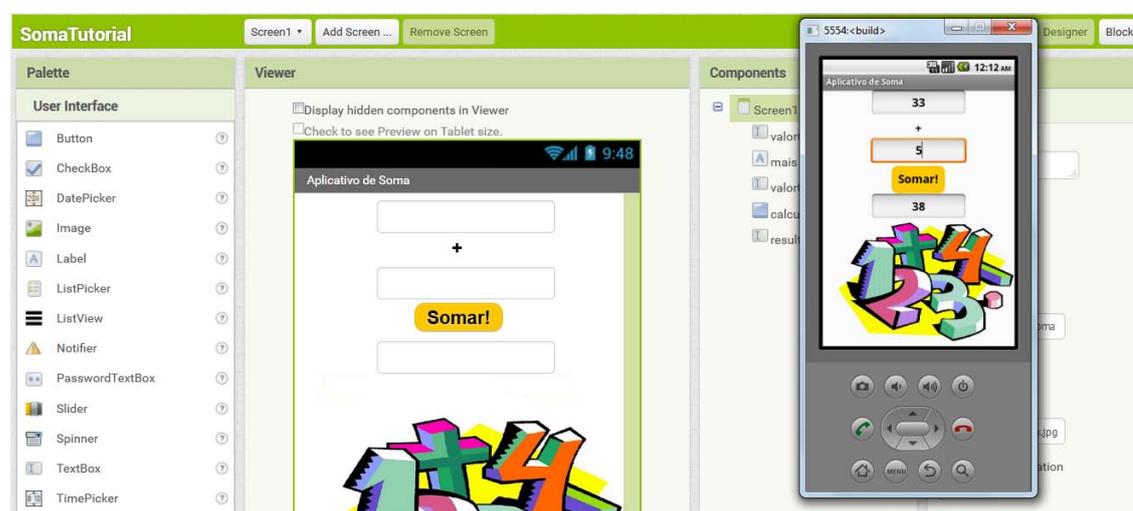


Figura 123 – Ilustração da plataforma executando o emulador no computador.

Se optar por testar o aplicativo no dispositivo móvel via conexão USB, escolha a *option three* (Figura 115) ou diretamente neste endereço⁴³ <<http://appinventor.mit.edu/explore/ai2/setup-device-usb.html>> aparecerá duas etapas para instalação. A primeira é um programa que será instalado no computador. Escolha o sistema operacional desejado e siga os passos da instalação. Feito isto, a outra etapa é instalar um aplicativo no dispositivo móvel, que é o mesmo da conexão via wi-fi. Basta seguir os links mostrados na tela.

⁴³ Acessado em 09 de Dezembro de 2015.

Step 1: Install the App Inventor Setup Software

To connect with USB, you need to first install the App Inventor setup software on your computer. (This is not required for the wifi method.) Follow the instructions below for your operating system, then **come back to this page** to move on to step 2

Important: If you are updating a previous installation of the App Inventor software, see [How to update the App Inventor Software](#).

- [Instructions for Mac OS X](#)
- [Instructions for Windows](#)
- [Instructions for GNU/Linux](#)

You can check whether your computer is running the latest version of the software by visiting the [Connection Test Page](#).

Step 2: Download and install the MIT AI2 Companion App on your phone.

Open your device's QR code scanner and scan the QR code on the left below to download the Companion App from the Play Store. If you can't use the Play Store, use the QR code on the right to download the Companion App directly to your phone.

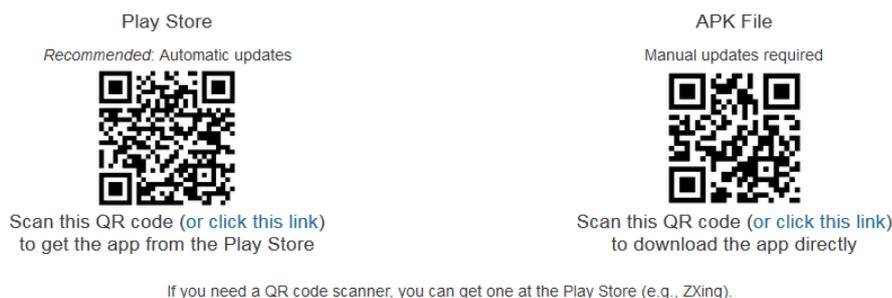


Figura 124 – Links de downloads dos programas para serem instalados no computador e no dispositivo móvel.

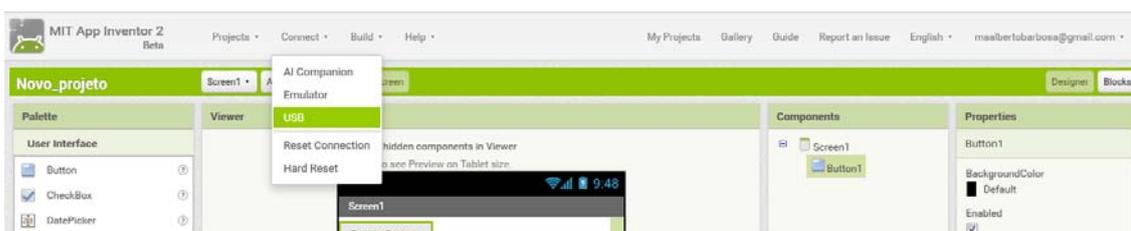


Figura 125 – Demonstração da escolha *USB* para testar o projeto.

Para usar a conexão via USB, talvez seja necessário ativar a função⁴⁴ **DESENVOLVEDOR** no dispositivo móvel. Basta ir em **CONFIGURAÇÕES**, depois em **PROGRAMADOR**, ativar a função e permitir a **DEPURAÇÃO USB** (Figura 126).

É possível ainda obter o aplicativo desenvolvido clicando no item *Build* (Construir) no menu. Em *App (provide QR code for .apk)* será disponibilizado um link para baixar a instalação definitiva do projeto em seu dispositivo pelo QRcode. Optando por *App (save .apk to my computer)* a instalação do aplicativo será baixado no computador para ser transferido para o smartphone/tablet e também compartilhado para outros usuários (Figura 127).

⁴⁴ Vale ressaltar que além das diferentes versões do Android, fabricantes de smartphones e tablets costumam personalizar o sistema operacional. Portanto, se não conseguir ativar a função desenvolvedor, consulte o manual de instruções do aparelho.



Figura 126 - Ilustração da ativação da função DESENVOLVEDOR em diferentes versões do Android.

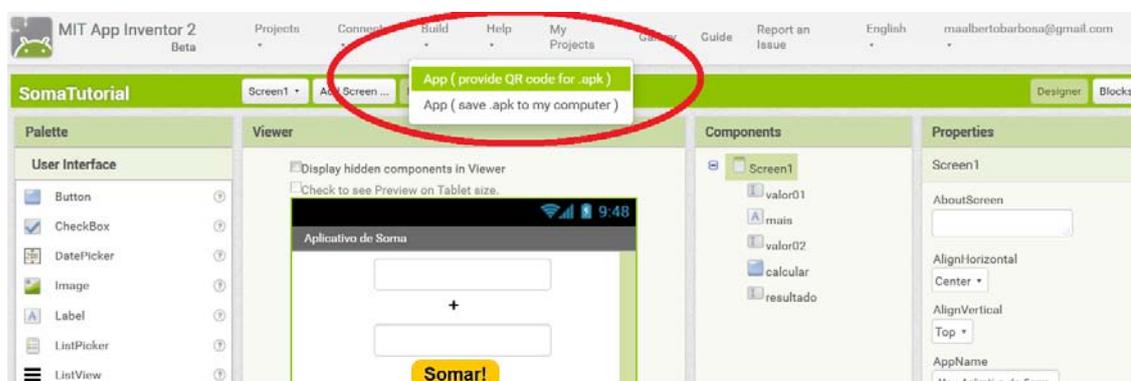


Figura 127 – Destaque para a opção que exporta a instalação do aplicativo.

ANEXO B – CONHECENDO A PLATAFORMA.

Comece acessando a página do MIT App Inventor <appinventor.mit.edu/explore/index-2.html>. Em seguida, clique no botão **Create apps!** (Criar aplicativos!) no canto superior direito para fazer o acesso através da sua conta do Google (Gmail), conforme Figura 129.

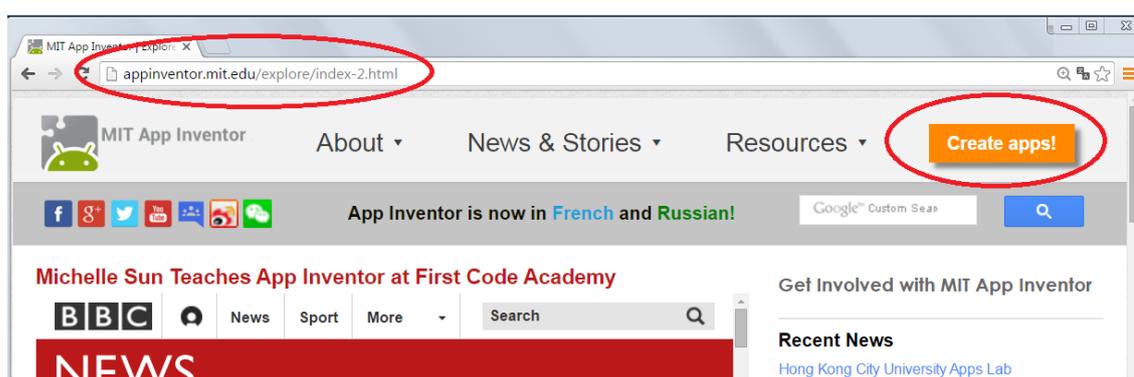


Figura 128 – Página inicial do MIT App Inventor.

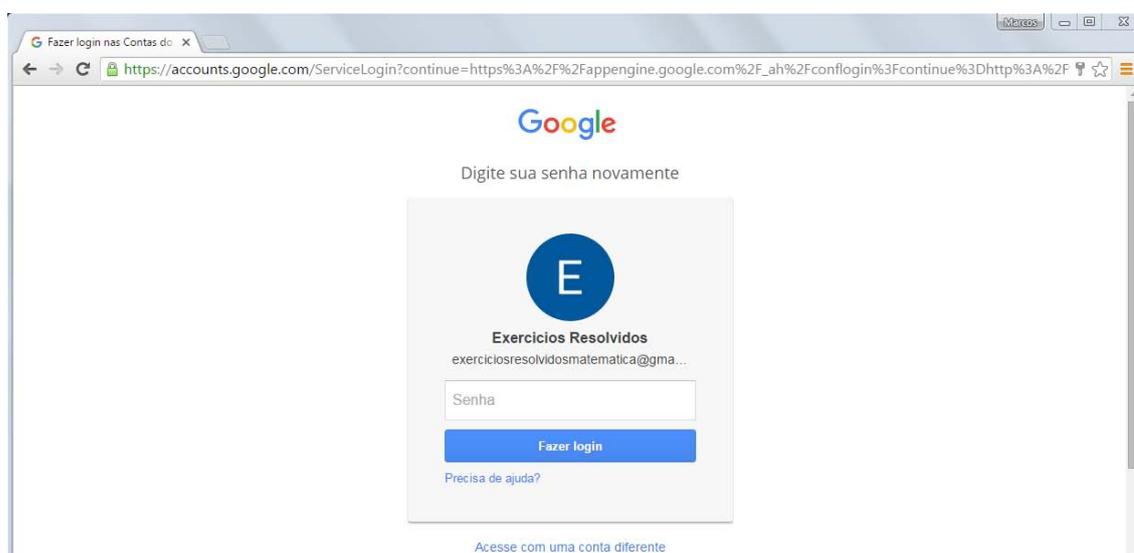


Figura 129 – Página de acesso à conta do Google.

Feito o login, você será direcionado para a plataforma de desenvolvimento, onde dali poderá começar os seus projetos (Figura 130). Nesta tela é exibido o menu com as opções relacionadas à criar, salvar, exportar, editar, construir e executar os aplicativos, além de disponibilizar os arquivos de ajuda (em inglês) e também aplicativos desenvolvidos por outros usuários através da opção *Gallery* (Galeria).

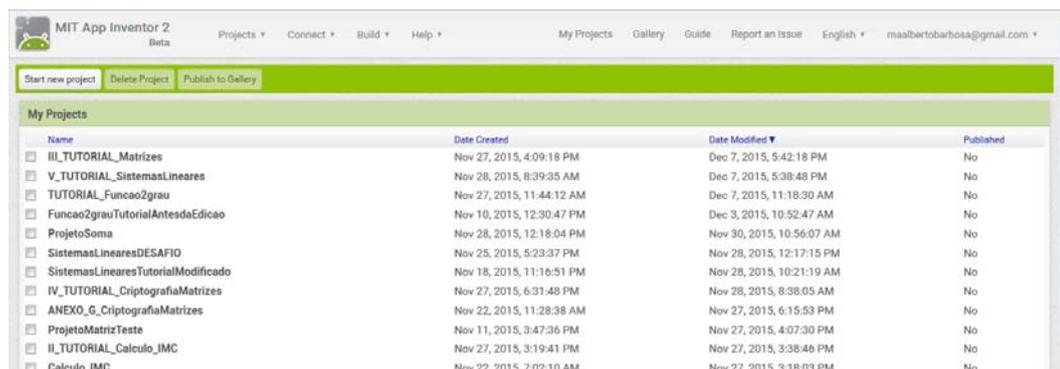


Figura 130 – Página inicial da plataforma exibindo o menu de opções e os projetos desenvolvidos.

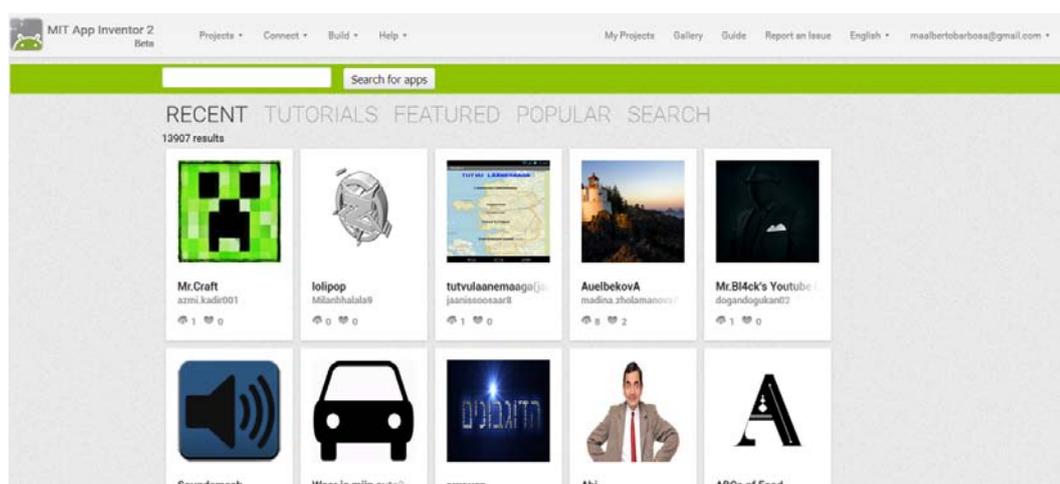


Figura 131 – Página da *Gallery* (Galeria) do MIT App Inventor 2 disponibilizando aplicativos desenvolvidos por outros usuários.

Para começar a desenvolver você deve ir no menu *Projects* (Projetos), e depois em *Start new project* (Começar novo projeto) ou também clicar no botão (Figura 132). Logo após, será aberta uma caixa de diálogo onde deverá escrever o nome do seu novo trabalho (Figura 133). Não pode iniciar com números, não deve conter espaços ou caracteres como ponto, vírgula, exclamação ou interrogação, ou mesmo barra e asterisco.

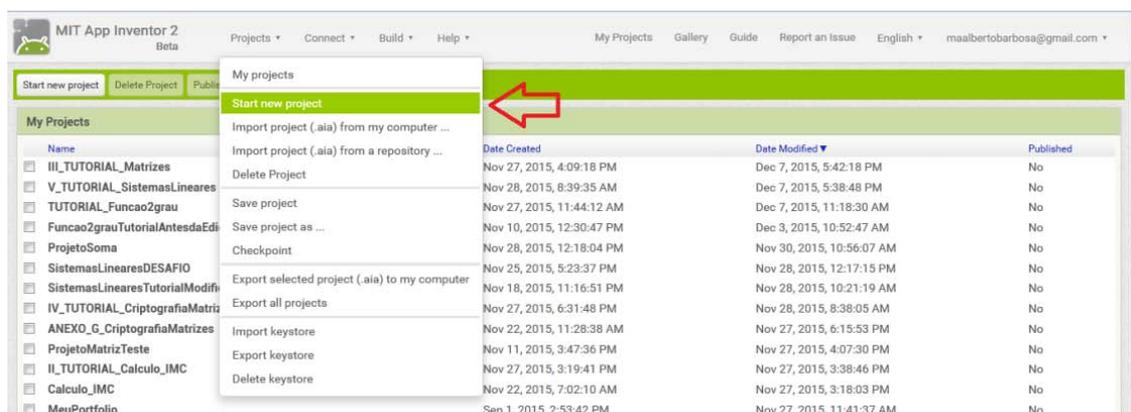


Figura 132 – Demonstração do procedimento de criar um novo projeto.

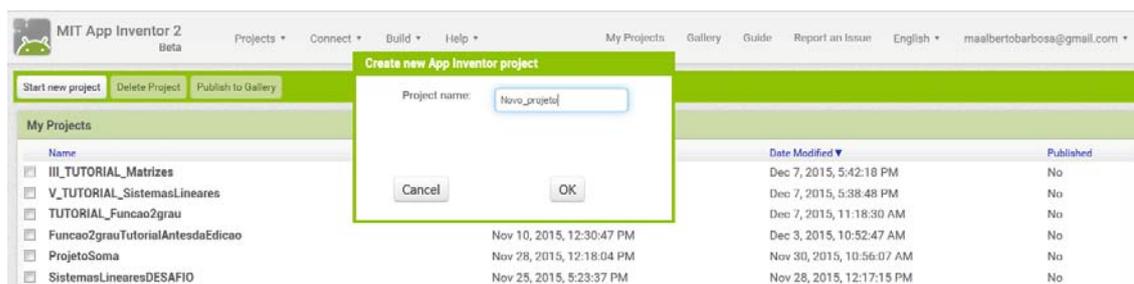


Figura 133 – Caixa de diálogo para entrada do nome escolhido do novo projeto.

Com a página do MAI aberta, perceberá que há quatro colunas: *Palette* (Paleta), *Viewer* (Visualizador), *Components* (Componentes) e *Properties* (Propriedades). Em *Palette* há os elementos básicos de interface de usuários: botões, caixas de textos, listas, botões de marcar, etc. No bloco *Viewer* é simulado a tela inicial do smartphone/tablet mostrando como ficaria os elementos inseridos pelo desenvolvedor. Já no bloco *Components* é listado o inventário de todos os elementos inseridos na tela *Viewer*. E finalmente, na caixa *Properties* ficam disponíveis as opções para editar cada elemento selecionado em *Viewer* ou *Components*, tais como cor do objeto, tamanho da fonte, legendas e outros atributos.

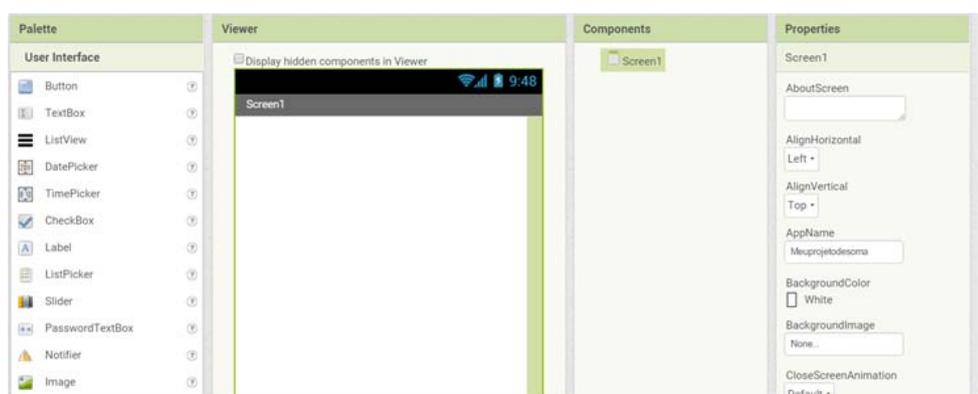


Figura 134 – Ambiente de trabalho do MAI.

Para adicionar componentes na tela *Viewer*, basta arrastar e soltar no local desejado.

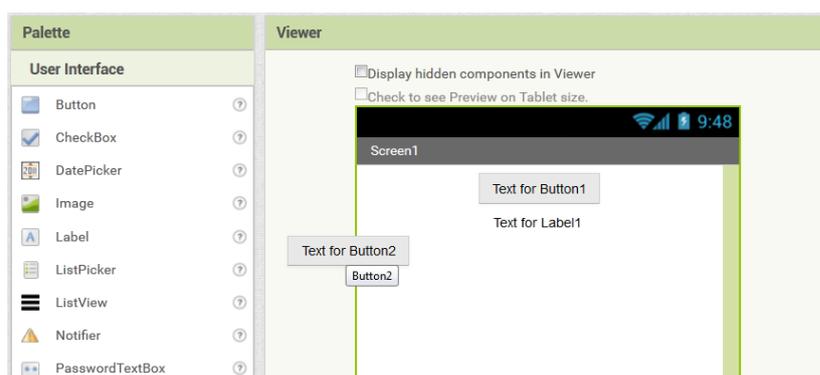


Figura 135 – Procedimento de arrastar e soltar componentes na tela *Viewer*.

Na figura a seguir é mostrado a modificação de algumas propriedades do componente *Button* adicionado anteriormente. Foi alterado a cor do botão em *BackgroundColor* (Cor de fundo), escolhido a opção *rectangular* (retangular) em *Shape* (Forma), e alterado o texto original que aparece no botão para “BOTAO CALCULAR” na propriedade *Text*, finalizando com a mudança do nome do objeto clicando no botão *Rename* (Renomear).

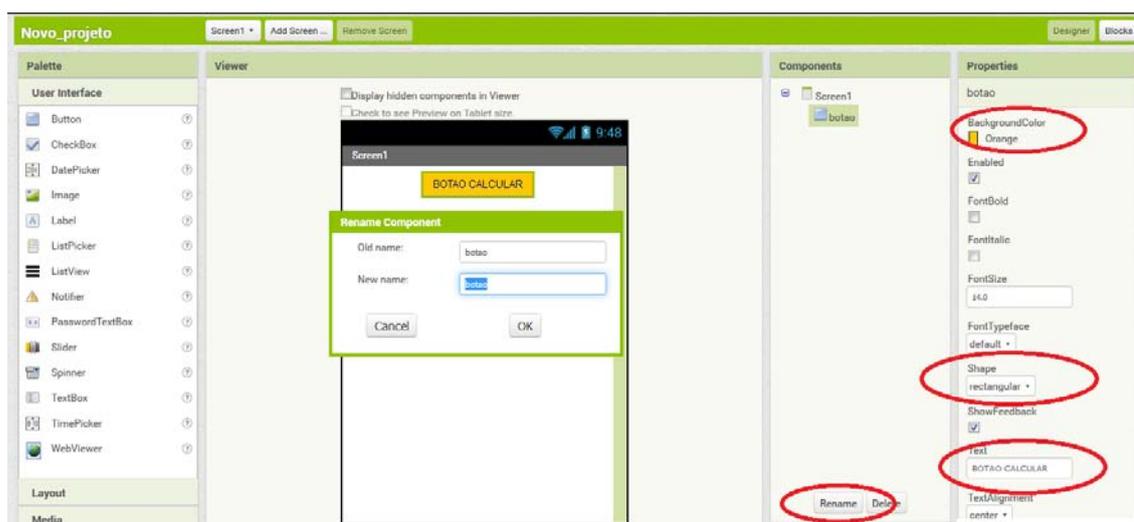


Figura 136 – Imagem aponta os locais onde foram realizadas as modificações no componente *Button*.

Uma vez adicionados todos os componentes que deseja, é necessário criar funções para cada um deles. Assim, clique no botão *Blocks* (Blocos) no canto superior direito, e será aberto a área de trabalho que contém os blocos de programação.

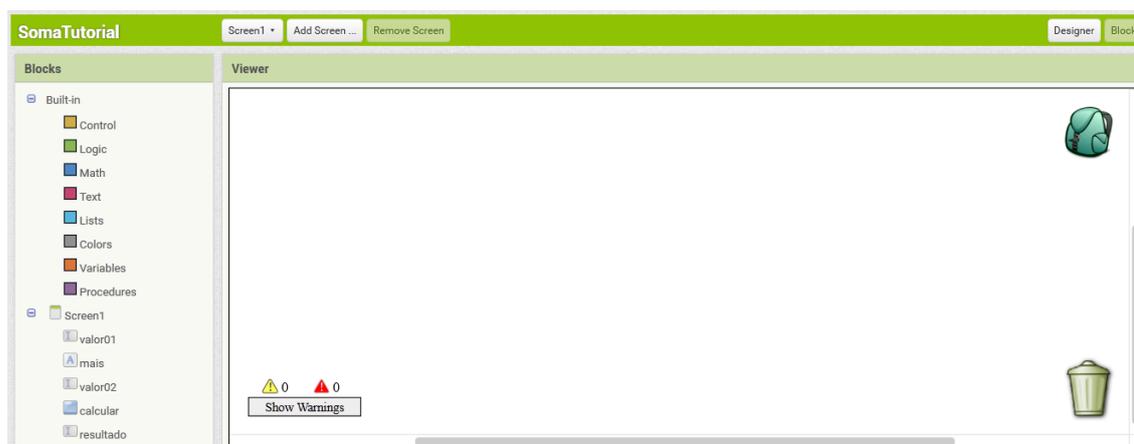


Figura 137 – Ambiente do MAI para manuseio dos blocos responsáveis pelas funções dos componentes inseridos na tela *Designer*.

Neste ambiente, o lado esquerdo é formado por menus que contém o inventário de todos os componentes adicionados na tela *Viewer* e diversas funções para serem associadas aos

objetos.

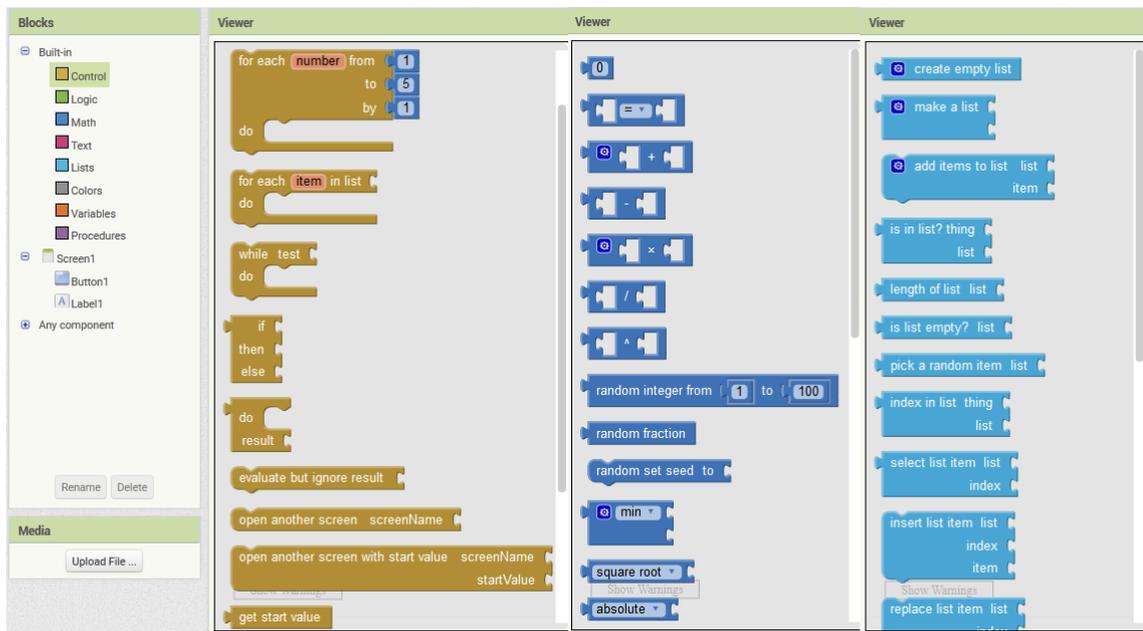


Figura 138 – Os blocos laranjas são da função *Control* (Controle), os azuis escuros de *Math* (Matemática) e os azuis claros de *Lists* (Listas).

Na Figura 139, é ilustrado a construção de um conjunto usando o mesmo sistema de arrastar e soltar composta por três partes: o bloco laranja do botão que ao ser clicado repetirá o que foi escrito na caixa de texto (bloco verde claro) na etiqueta (bloco verde escuro).

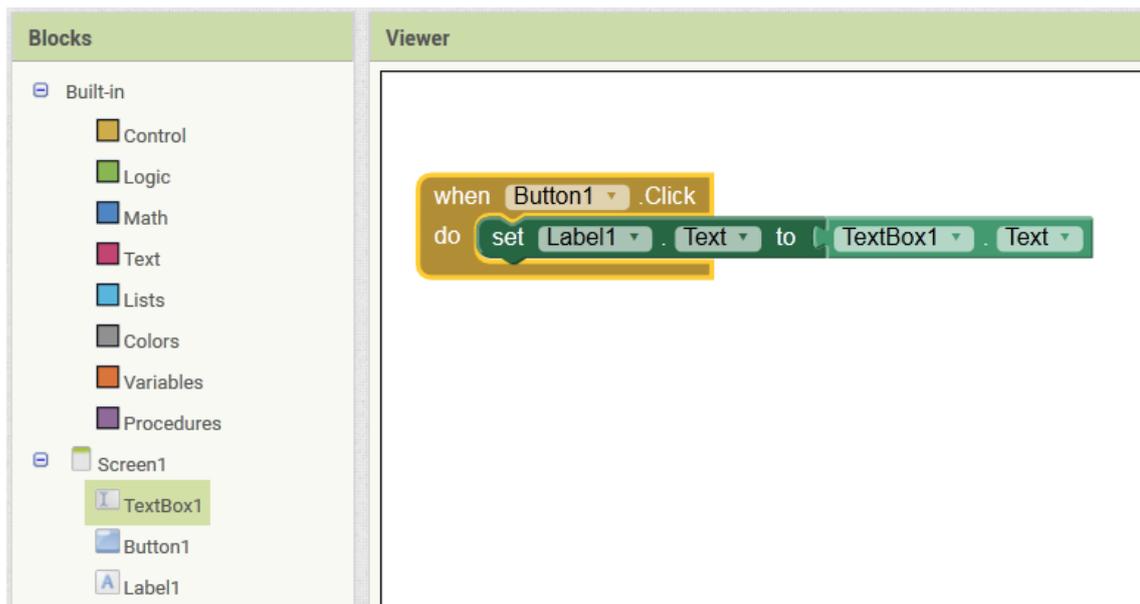


Figura 139 – Exemplo de uma montagem de blocos associando uma regra ao botão.

ANEXO C – CONSTRUINDO O PRIMEIRO APLICATIVO.

Como esta é a primeira experiência com o MAI, comece treinando a construção de uma aplicação simples de somatória para familiarizar com as instruções básicas. Portanto será demonstrado passo a passo usando uma linguagem acessível e devidamente ilustrada.

Na página inicial do MAI (Figura 140), clique em *Projects* (Projetos), e em seguida *Start new project* (Começar novo projeto). Feito isto, será aberto uma caixa de diálogo solicitando um nome para o projeto (Figura 141).

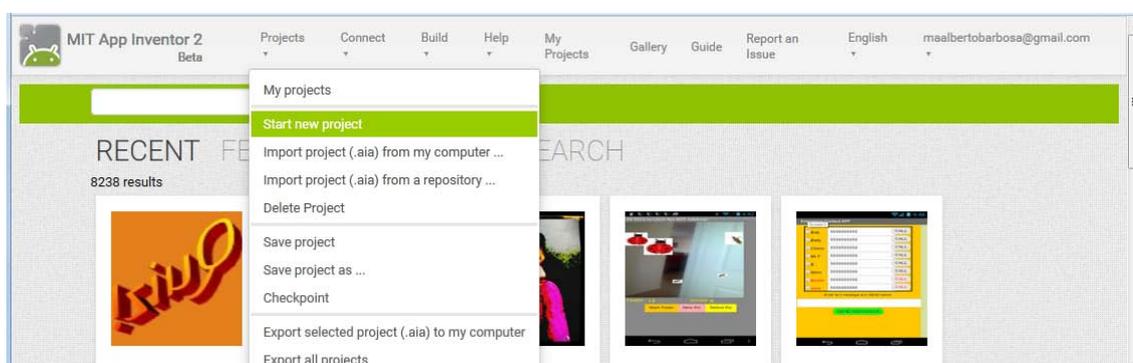


Figura 140 – Página inicial da plataforma do MAI.

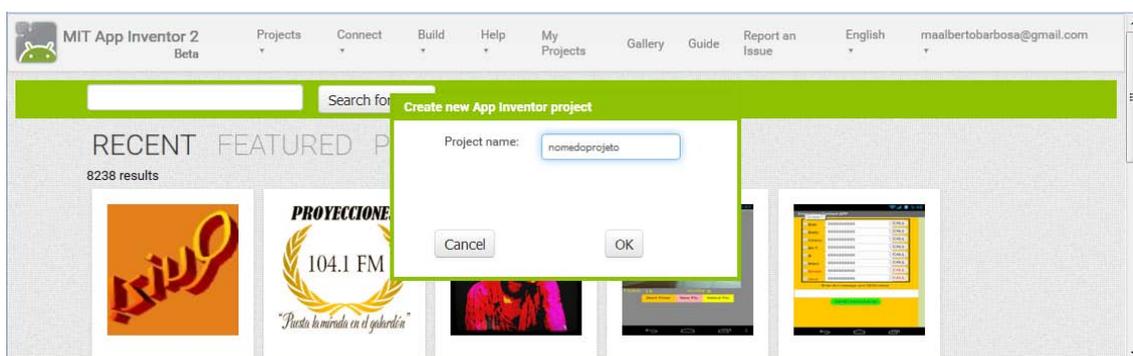


Figura 141 – Ambiente de trabalho do MAI.

Quando a plataforma estiver pronta, adicione alguns componentes disponíveis no menu *Palette* (paleta) à direita da tela. Para inserir, clique no objeto desejado em *Palette* e mantendo o botão direito do mouse apertado arraste e solte dentro da tela que simula um smartphone/tablete no bloco *Viewer* (Visualizador). Coloque uma caixa de texto  **TextBox**, uma etiqueta  **Label**, outra caixa de texto, um botão  **Button** e outra caixa de texto, conforme imagem a seguir.

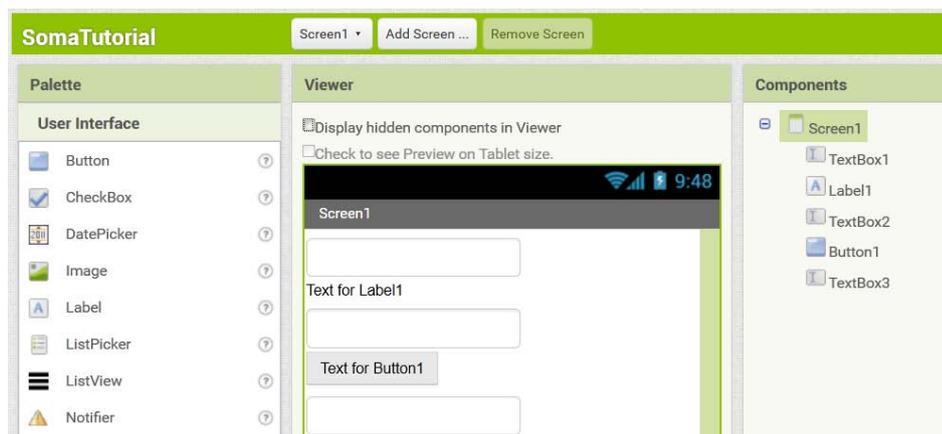


Figura 142 – Componentes inseridos na tela inicial do aplicativo.

O objetivo é que as duas primeiras caixas de textos recebam os números para serem somados, que o botão ordene a operação matemática e o resultado apareça na última caixa de texto.

Prosseguindo, na aba *Components* (Componentes), selecione cada item adicionado, clique no botão *Rename* (Renomear) e substitua as palavras *TextBox1* para “valor01”, *TextBox2* para “valor02”, *Label1* para “mais”, *Button1* para “calcular”, e *TextBox3* para “resultado”.



Figura 143 – Lista de componentes inseridos e editados.

Em seguida clique no botão *Blocks* (Blocos) no canto superior direito, que abrirá a tela ilustrada na Figura 144. Nesta área que será criada a parte a parte lógica do aplicativo através da montagem dos blocos.



Figura 144 – Área de trabalho para edição dos blocos.

Na coluna à esquerda, *Blocks*, no menu *Math* (Matemática)  *Math* e selecione o bloco azul que corresponde a soma .

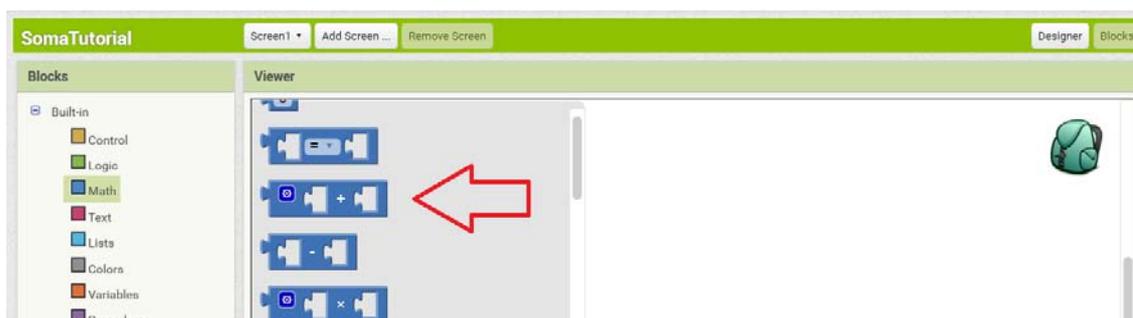
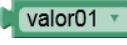
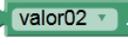


Figura 145 – Bloco de soma.

Clique nos componentes “valor01” e “valor02”, procure os blocos verdes “valor01.Text”  e “valor02.Text”  arrastando para os espaços vazios no bloco azul de soma.

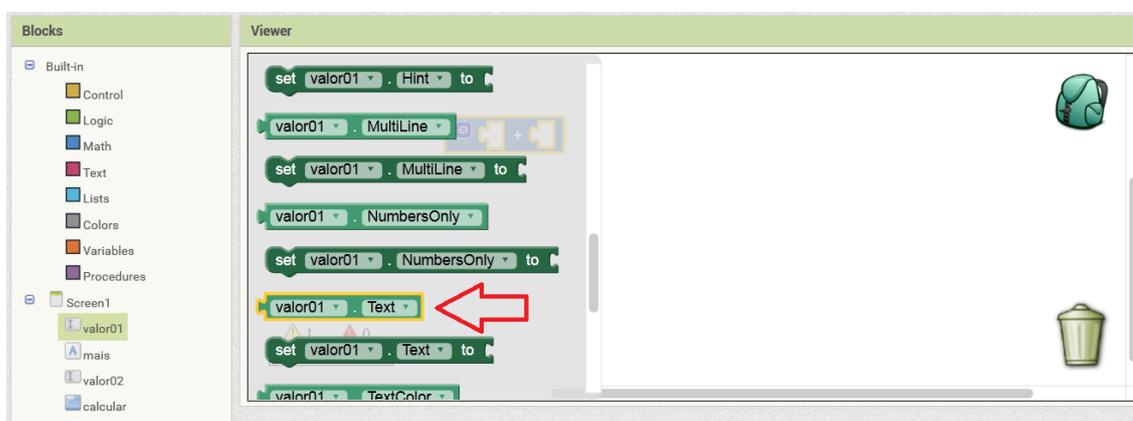


Figura 146 – Bloco relacionado ao atributo texto do componente “valor01”.



Figura 147 – Composição de soma dos componentes “valor01” e “valor02”.

Agora será escolhido o local onde o resultado deverá aparecer. Clique no componente “resultado” e escolha o bloco `set resultado . Text to` fazendo a junção com o bloco azul da soma, conforme Figura 149.

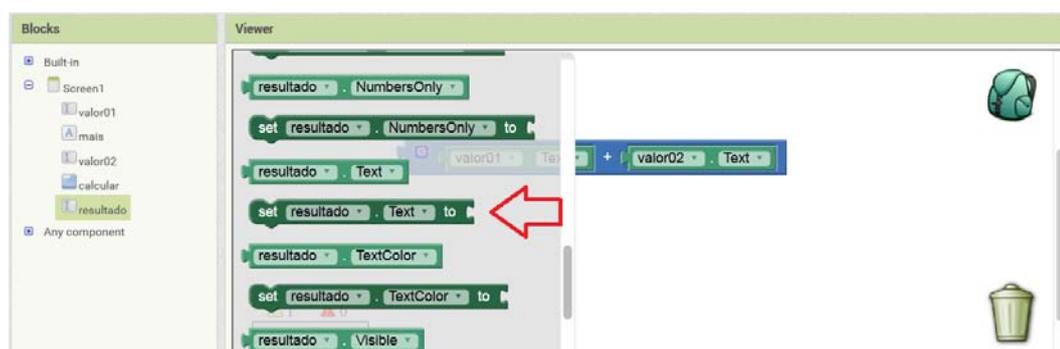


Figura 148 – Bloco relacionado à seleção do componente “resultado”.



Figura 149 – Composição de soma dos componentes “valor01” e “valor02” para ser exibido em “resultado”.

Em seguida, clique no componente “calcular”, que é o botão que vai gerenciar a operação de soma, e selecione o primeiro item `when calcular . Click` (função *Click*).

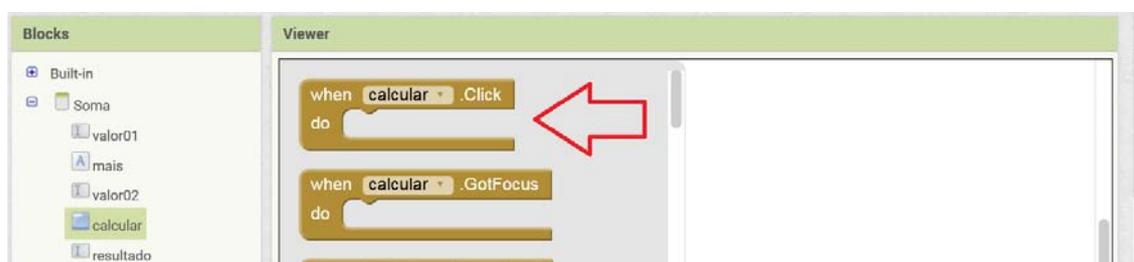


Figura 150 – Função *click* do componente botão “calcular”.

No bloco que foi inserido, possui a seguinte mensagem: “when calcular.Click do”, ou seja: “quando o calcular é clicado/apertado, fazer”. Portanto, basta agregar o bloco da Figura 149 com o bloco da Figura 150.



Figura 151 – Composição final do procedimento de soma entre os componentes “valor01” e “valor02”.

Feito isto, teste⁴⁵ o projeto no smartphone ou no tablet.

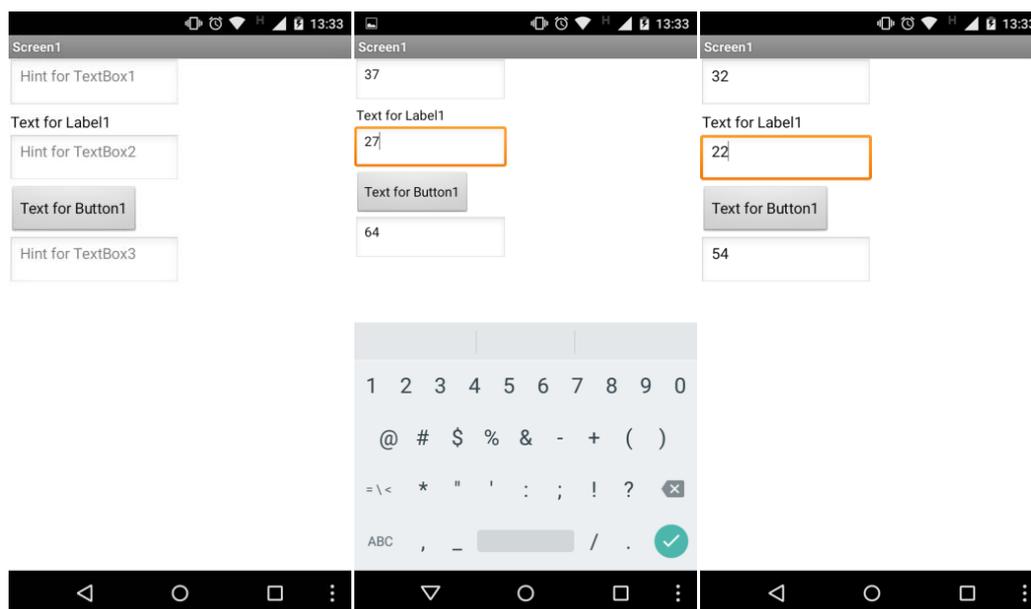


Figura 152 – Capturas de tela do smartphone exibindo o aplicativo funcionando.

Uma vez funcionando, deixe a interface do aplicativo mais atraente. Clique no botão *Designer* (Desenhista), no canto superior direito, e retorne à página de componentes do aplicativo. Para fazer as modificações, selecione cada componente na coluna *Viewer* ou na coluna *Components* para editá-los em *Properties* (Propriedades).

Faça as seguintes mudanças:

(1) Selecione o componente “*Screen1*”, escolha *Center* (Centro) em *AlignHorizontal* (Alinhamento Horizontal). Em *AppName* (Nome do Aplicativo) coloque o nome que preferir. Neste tutorial foi usado “Meu Aplicativo de Soma”. Esta frase é a que aparecerá abaixo do ícone do aplicativo quando estiver instalado no dispositivo. Marque a opção *Scrollable* e inclua em *Title* a frase “Aplicativo de Soma”.

⁴⁵ Para testar o aplicativo, consulte o ANEXO A – O que instalar, onde está descrito detalhadamente as três formas de testes.

(2) Em todas as caixas de texto marque a opção *FontBold* (Negrito), mude *FontSize* (Tamanho da Fonte) para 20, apague o texto no item *Hint* (Sugestão), selecione o item *NumbersOnly* (Somente Números), em *TextAlignment* (Alinhamento de Texto), escolha *Center*;

(3) Na etiqueta “mais”, marque a opção *FontBold*, mude *FontSize* para 20, em *Text* coloque o sinal de “+”, em *TextAlignment*, escolha *Center*;

(4) Para o botão “calcular”, em *BackgroundColor* (Cor de Fundo), escolha *Orange* (Laranja), marque a opção *FontBold*, altere a propriedade *FontSize* para 20, em *Shape* (Forma), escolha *rounded* (arredondado), substitua a frase do item *Text* para “Somar!”, em *TextAlignment*, escolha *Center*.

Veja como fica após as alterações:

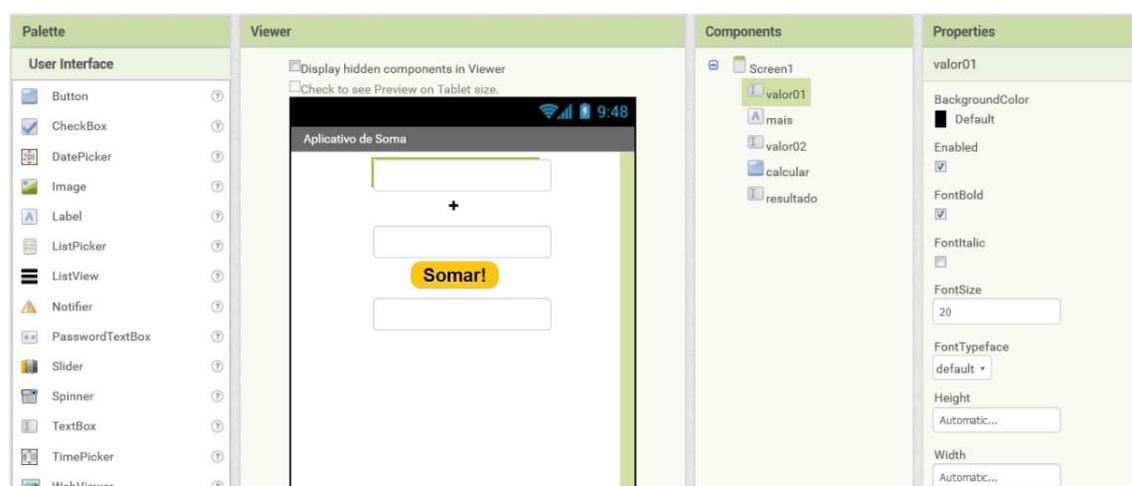


Figura 153 – Ilustração das modificações aplicadas nos componentes.

Quando executar o aplicativo, irá perceber as mudanças feitas no tamanho das letras, no nome do botão, na cor do fundo do aplicativo, bem como que aparecerá apenas o teclado numérico quando você clicar nas caixas de texto. Isso se deve ao fato de ter selecionado o item *NumbersOnly*.

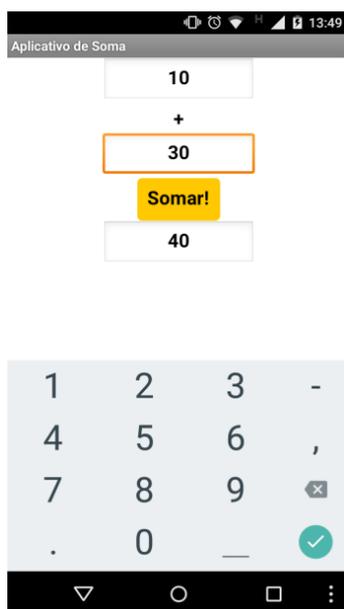


Figura 154 – Captura de tela do smartphone exibindo o aplicativo funcionando além das edições realizadas.

Agora faça modificações para deixar personalizado o aplicativo da forma que achar melhor, alterando as características dos componentes em *BackgroundColor*, *FontBold*, e *FontSize* e a posição do *TextAlignment* para *Center*. Altere as cores das etiquetas, use negritos, insira uma imagem de fundo, escolhendo uma boa figura na internet e adicione através do botão *UploadFile* (Subir arquivo) na aba *Média*⁴⁶. Veja algumas imagens com melhorias aplicadas no aplicativo.



Figura 155 – Modelo diferente de layout do aplicativo.

⁴⁶ Média (Mídia) pode ser arquivos de imagem, áudio ou vídeo.

Quando finalizar todas as modificações que achar conveniente, é possível compartilhar o aplicativo. A primeira opção é clicar em *Build* (Compilar), e depois em *App (provide QR code for .apk)* (App - fornecer o QR code para o .apk) onde será gerado um código para fazer o download diretamente para o smartphone (Figura 156 e Figura 157).

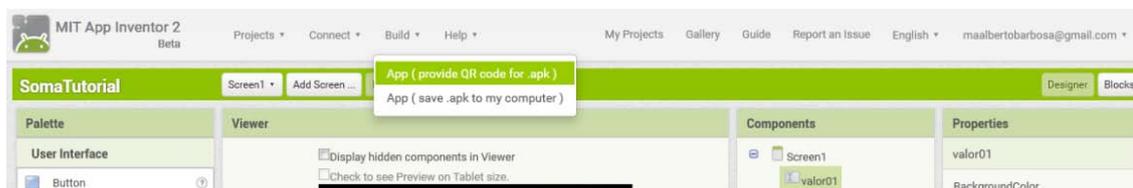


Figura 156 – Procedimento para exportar o projeto construído.

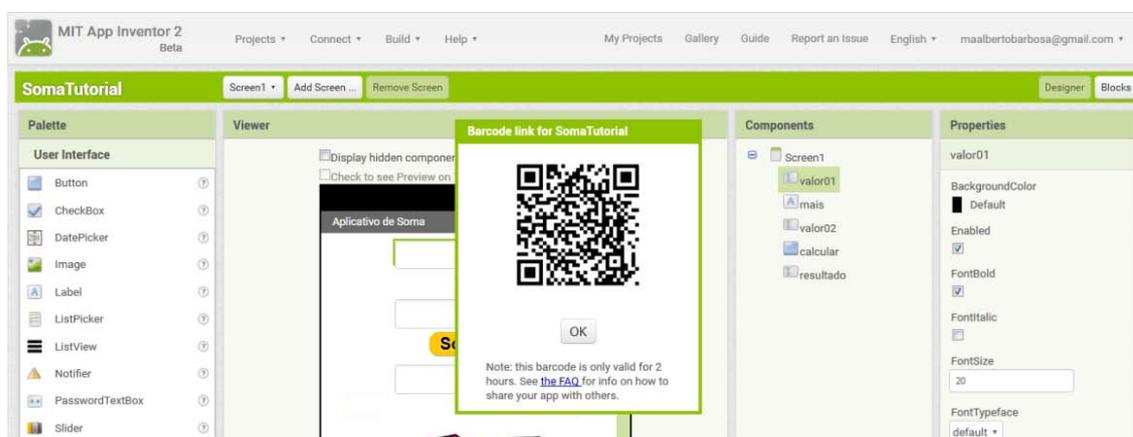


Figura 157 – Código de barra com o atalho para download do projeto direto para o dispositivo móvel.

A segunda opção é escolher *App (save .apk to my computer)* (App - salvar .apk em meu computador), onde será salvo a instalação do aplicativo no seu computador, e daí é possível copiar para o seu dispositivo e compartilhar com outras pessoas.

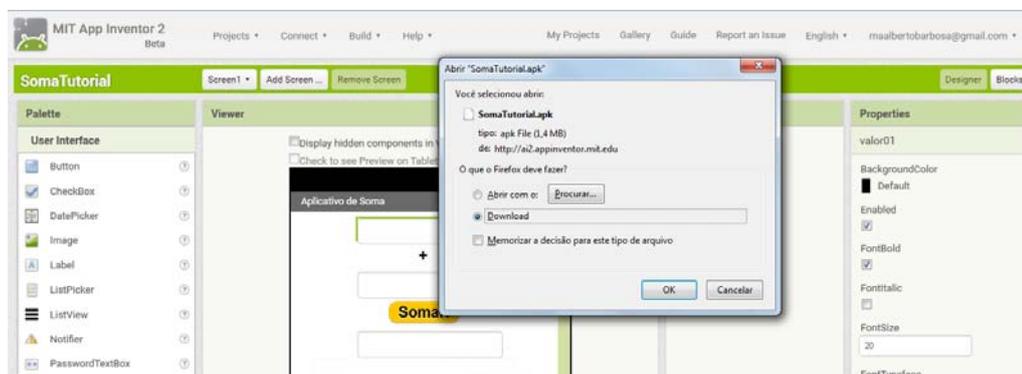


Figura 158 – Janela de diálogo indicando o início do download do aplicativo para o computador.

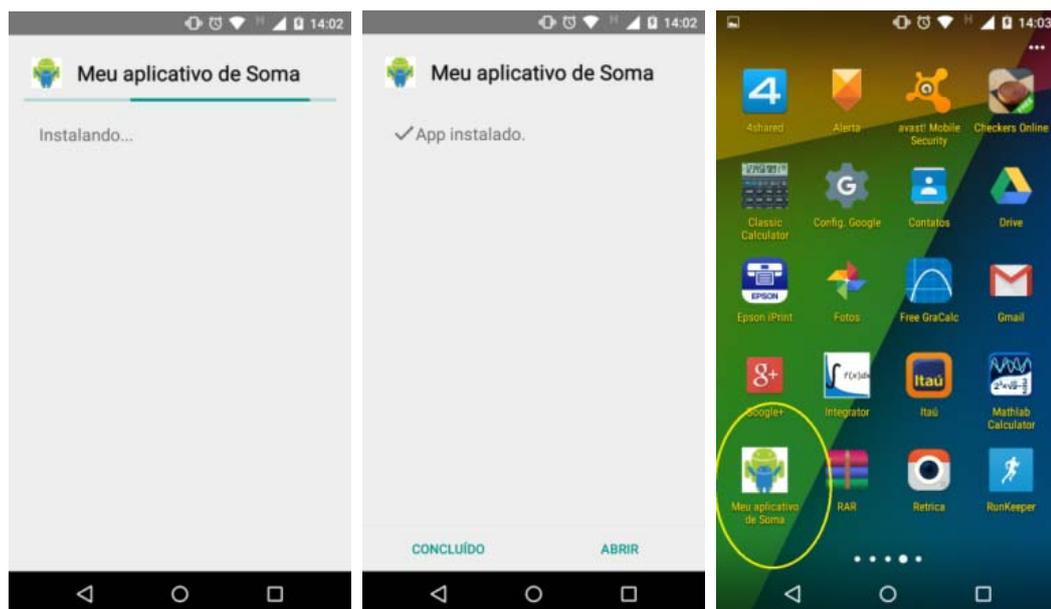


Figura 159 – Processo de instalação do aplicativo no smartphone.

