UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE MATEMÁTICA
PROGRAMA DE PÓS–GRADUAÇÃO EM MATEMÁTICA EM REDE NACIONAL

ERINEU SANTOS DA COSTA

CRIAÇÃO DE APLICATIVOS MATEMÁTICOS PARA DISPOSITIVOS MÓVEIS USANDO A PLATAFORMA THUNKABLE

FORTALEZA
2023

ERINEU SANTOS DA COSTA

CRIAÇÃO DE APLICATIVOS MATEMÁTICOS PARA DISPOSITIVOS MÓVEIS USANDO A PLATAFORMA THUNKABLE

Dissertação apresentada ao Programa de Pós–Graduação em Matemática em Rede Nacional do Departamento de Matemática da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Mestre em Matemática. Área de concentração: Ensino de Matemática.

Orientador: Prof. Dr. Marcelo Ferreira de Melo.

FORTALEZA

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C871c Costa, Erineu Santos da.
CRIAÇÃO DE APLICATIVOS MATEMÁTICOS PARA DISPOSITIVOS MÓVEIS USANDO A PLATAFORMA THUNKABLE / Erineu Santos da Costa. – 2023.
79 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Departamento de Matemática, Programa de Pós-Graduação em Matemática em Rede Nacional, Fortaleza, 2023.
Orientação: Prof. Dr. Marcelo Ferreira de Melo..

1. aplicativos. 2. dispositivos móveis. 3. lógica de programação. 4. programação em blocos. 5. thunkable. I. Título.

CDD 510

ERINEU SANTOS DA COSTA

CRIAÇÃO DE APLICATIVOS MATEMÁTICOS PARA DISPOSITIVOS MÓVEIS

USANDO A PLATAFORMA THUNKABLE

Dissertação apresentada ao Programa de Pós–Graduação em Matemática em Rede Nacional do Departamento de Matemática da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Mestre em Matemática. Área de concentração: Ensino de Matemática.

Aprovada em: xx/xx/xxxx.

BANCA EXAMINADORA

____

Prof. Dr. Xxxxxxxxx Xxxxxxx (Orientador)
Universidade Federal do Ceará (UFC)

____

Prof. Dr. Xxxxxxxxx Xxxxxxx
Universidade Federal do Ceará (UFC)

____

Prof. Dr. Xxxxxxxxx Xxxxxxx
Universidade Estadual do Ceará (UECE)

Aos meus pais, Edmilson Pereira da Costa e Maria Erondina Santos da Costa.

Aos meus filhos, Vanessa, Maria de Lourdes, Felipe e Penélope.

AGRADECIMENTOS

Ao Prof. Dr. Marcelo Ferreira de Melo, pela excelente orientação e por todo esses anos de ensino, os quais muito enriqueceram meus conhecimentos.

Aos professores participantes da banca examinadora Xxxxx Xxxxx Xxxxx e Xxxxx Xxxxx Xxxxx pelo tempo, pelas valiosas colaborações e sugestões.

Aos professores das disciplinas do mestrado, que sempre apoiaram os nossos estudos, tiraram dúvidas, sempre disponíveis às nossas dificuldades.

Aos professores entrevistados, pelo tempo concedido nas entrevistas.

Aos colegas da turma de mestrado, pelo apoio nos momentos mais difíceis, pelas reflexões, críticas e sugestões recebidas.

“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).

RESUMO

Este trabalho propõe o uso da plataforma Thunkable para o ensino de matemática, através da criação de aplicativos para dispositivos móveis. A justificativa para esta proposta é a necessidade de tornar o ensino de matemática mais acessível e atraente para os alunos, especialmente para aqueles que têm dificuldades com a disciplina. Os objetivos que procuramos atingir são: o desenvolvimento de aplicativos que auxiliem na compreensão de conceitos matemáticos, aumentar o interesse dos alunos pela matemática, por meio do uso de ferramentas tecnológicas, e facilitar o aprendizado da disciplina. A execução deste trabalho se baseia em estudos que indicam a eficácia de aplicativos móveis no ensino de matemática, bem como em teorias da aprendizagem que destacam a importância da interatividade e da experiência do usuário. A metodologia adotada consiste em uma abordagem teórica dos conteúdos que compõem o currículo da base comum adotada na edução básica, seguida da criação de protótipos de aplicativos utilizando a plataforma Thunkable e, por fim, a avaliação dos aplicativos pelos usuários. Por meio da aplicação dessa metodologia em disciplinas eletivas de criação de aplicativos, já temos indícios que apontam para uma evolução por parte dos alunos no domínio dos conceitos apresentados, não só da matemática, mas também na área de programação, raciocínio lógico, resolução de problemas e uso de outras ferramentas tecnológicas. A plataforma Thunkable tem se mostrado uma ferramenta eficiente não só para a criação de aplicativos de matemática, que podem melhorar a compreensão e o interesse dos alunos pela disciplina, mas também como um instrumento que pode ser usado para resolver problemas concretos, via implementação de soluções tecnológicas. Além disso, o uso de aplicativos móveis no ensino de matemática pode contribuir para a democratização do acesso à educação e para a formação de cidadãos críticos e conscientes.

Palavras–chave: aplicativos; dispositivos móveis; lógica de programação; programação em blocos; thunkable; kodular; scratch.

ABSTRACT

This work proposes the use of the Thunkable platform for teaching mathematics, through the creation of applications for mobile devices. The justification for this proposal is the need to make mathematics teaching more accessible and attractive for students, especially for those who have difficulties with the discipline. The objectives we seek to achieve are: the development of applications that help in the understanding of mathematical concepts, increase students' interest in mathematics, through the use of technological tools, and facilitate the learning of the discipline. The execution of this work is based on studies that indicate the effectiveness of mobile applications in teaching mathematics, as well as on learning theories that highlight the importance of interactivity and user experience. The methodology adopted consists of a theoretical approach to the contents that make up the common base curriculum adopted in basic education, followed by the creation of application prototypes using the Thunkable platform and, finally, the evaluation of applications by users. Through the application of this methodology in elective disciplines of application creation, we already have indications that point to an evolution on the part of the students in the mastery of the presented concepts, not only in mathematics, but also in the area of programming, logical reasoning, problem solving and use of other technological tools. The Thunkable platform has proven to be an efficient tool not only for creating math applications, which can improve students' understanding and interest in the subject, but also as an instrument that can be used to solve concrete problems, via the implementation of solutions. technologies. In addition, the use of mobile applications in teaching mathematics can contribute to the democratization of access to education and to the formation of critical and aware citizens.

Keywords: applications; mobile devices; programming logic; block programming; thunkable; kodular; scratch.

SUMÁRIO

1 INTRODUÇÃO

A matemática é uma das disciplinas mais importantes para a formação dos indivíduos, pois é uma ciência que proporciona a compreensão do mundo ao nosso redor, além de ser essencial para o desenvolvimento de habilidades cognitivas e de raciocínio lógico. No entanto, muitos estudantes enfrentam dificuldades em aprender matemática, o que pode levar a uma aversão pela disciplina e a resultados insuficientes. A maioria considera a matemática uma disciplina difícil e abstrata, o que leva à falta de motivação e interesse.

Os fatores que levam a essas dificuldades são bastante numerosos. Temos desde questões metodológicas do nosso sistema de ensino, formação de professores para atuar nas séries iniciais da educação básica, condições estruturais das nossas escolas, dentre outros.

Nesse contexto, a criação de aplicativos para dispositivos móveis pode ser uma alternativa promissora para tornar o ensino de matemática mais interessante, atraente e acessível. Dentre diversas opções, foi escolhida a plataforma Thunkable para o desenvolvimento do presente trabalho. Ela é uma plataforma de desenvolvimento de aplicativos móveis que permite criar aplicativos de forma simples e intuitiva, sem a necessidade de conhecimentos avançados em programação. Ela utiliza a linguagem de programação visual Blocky. Essa linguagem é uma linguagem de programação visual baseada em blocos, criada pelo Google em 2011, como parte do projeto Blockly. O objetivo principal do projeto Blockly era criar uma linguagem de programação visual que pudesse ser usada em vários projetos de programação e que fosse fácil de aprender e usar.

A linguagem Blocky é baseada em blocos que representam comandos de programação. Os blocos são arrastados e soltos em uma interface gráfica para criar programas. A linguagem permite a criação de programas em várias linguagens de programação, incluindo Python, JavaScript e outras. Ela é usada em várias plataformas de programação visual, além do Thunkable, incluindo a Code.org, App Inventor, Scratch, dentre outras.

A principal característica da linguagem Blockly é sua facilidade de uso e sua capacidade de ensinar os conceitos básicos de programação de forma acessível para iniciantes. A linguagem é baseada em blocos, o que significa que o usuário não precisa escrever código, mas sim montar blocos que representam as instruções de programação. Isso facilita muito a aprendizagem da linguagem de programação e permite que os usuários se concentrem nos conceitos em vez de se preocuparem em digitar código.

Outra característica importante da linguagem Blocky é sua flexibilidade. A linguagem pode ser usada para criar programas em várias linguagens de programação diferentes, o que permite que os usuários criem programas para várias plataformas. Além disso, a linguagem permite que os usuários criem seus próprios blocos, o que significa que é possível criar blocos personalizados para atender às necessidades específicas do projeto.

Uma das principais vantagens de ensinar matemática através da criação de aplicativos é a possibilidade de tornar o conteúdo mais interativo e dinâmico. Por exemplo, um aplicativo pode apresentar problemas matemáticos que o estudante deve resolver, fornecendo feedback imediato e permitindo que ele aprenda com os próprios erros. Além disso, os aplicativos podem apresentar gráficos, animações e outros recursos visuais que ajudam a ilustrar conceitos matemáticos de forma mais clara e acessível.

Outra vantagem da criação de aplicativos para ensinar matemática é a possibilidade de personalização do conteúdo. Os aplicativos podem ser adaptados para atender às necessidades e interesses específicos dos estudantes, permitindo que eles aprendam no próprio ritmo e com um nível de dificuldade adequado ao seu nível de conhecimento.

Além disso, a criação de aplicativos pode ser uma estratégia metodológica extremamente útil à educação matemática. Esta consiste em um aspecto fundamental do currículo escolar, pois fornece aos alunos habilidades importantes para a vida, incluindo pensamento crítico, resolução de problemas e tomada de decisões. Todas essas vêm a ser habilidades importantes para o mercado de trabalho, um dos focos das séries terminais do ensino médio.

Por fim, a criação de aplicativos para dispositivos móveis usando o thunkable ajuda a criar uma cultura de aprendizagem ao longo da vida. A tecnologia está mudando rapidamente, e os alunos precisam desenvolver habilidades para se adaptar e aprender continuamente. A criação de aplicativos para dispositivos móveis ajuda a desenvolver essas habilidades, pois os alunos estão aprendendo a usar novas ferramentas e tecnologias para resolver problemas matemáticos. Isso os prepara para o futuro, onde a tecnologia será cada vez mais importante.

Neste contexto, o uso de tecnologia pode ser uma ferramenta poderosa para tornar a aprendizagem de matemática mais acessível e envolvente para os alunos.

Em primeiro lugar, a criação de aplicativos para dispositivos móveis permite que os alunos aprendam matemática de forma prática e envolvente. Com o thunkable, os alunos podem criar aplicativos que resolvem problemas matemáticos, criam gráficos e diagramas, simulam situações da vida real e muito mais. Isso torna a aprendizagem mais significativa, uma vez que os alunos estão aplicando conceitos matemáticos em situações reais e estão criando algo tangível e útil.

Além disso, o thunkable é uma plataforma de desenvolvimento de aplicativos visual, o que significa que os alunos não precisam saber programação para criar aplicativos. Isso torna a criação de aplicativos mais acessível para todos os alunos, independentemente do nível de habilidade. Os alunos podem se concentrar na matemática em vez de aprender a programar, o que aumenta o interesse e a motivação para a aprendizagem.

Não bastassem as já apresentadas, mais uma vantagem da criação de aplicativos para dispositivos móveis usando o thunkable é que isso ajuda a melhorar a comunicação e a colaboração entre os alunos. Os alunos podem trabalhar em equipes para criar aplicativos, o que permite que eles compartilhem ideias e aprendam uns com os outros. Isso ajuda a desenvolver habilidades de comunicação e trabalho em equipe, que são habilidades importantes para a vida profissional.

Em suma, a criação de aplicativos para dispositivos móveis usando a plataforma thunkable é uma forma eficaz e envolvente de ensinar matemática. Isso ajuda a tornar a aprendizagem mais prática e acessível, desenvolve habilidades importantes para a vida profissional e cria uma cultura de aprendizagem ao longo da vida. É uma abordagem inovadora e moderna para ensinar matemática que vale a pena explorar.

2 FUNDAMENTAÇÃO TEÓRICA

Embora a plataforma thunkable seja relativamente nova, existem autores que já exploraram seu potencial como ferramenta de aprendizagem na educação básica. Aqui estão alguns exemplos:

Ribeiro, Souza e Silva (2021) relatam em seu estudo a utilização do thunkable para desenvolver aplicativos móveis voltados para o ensino de Matemática em uma escola pública no Brasil. Os autores mostraram que a criação de aplicativos ajudou os alunos a se envolverem mais com a aprendizagem, melhorando seu desempenho em testes e atividades.

Oliveira, Zambon e Andrade (2020) também utilizaram o thunkable para desenvolver aplicativos de ensino de matemática para alunos do ensino fundamental. Os autores relatam que a criação de aplicativos ajudou os alunos a desenvolver habilidades de pensamento crítico e resolução de problemas, além de melhorar sua motivação para a aprendizagem.

Em um estudo realizado por Carvalho e colaboradores (2021), os autores utilizaram o thunkable para criar um jogo educativo para ensinar sobre a água e seu uso consciente para alunos do ensino fundamental. Os resultados mostraram que a criação do jogo ajudou os alunos a entenderem melhor o tema e a se envolverem mais na aprendizagem.

Santos e Silva (2020) exploraram o potencial da plataforma thunkable para o ensino de ciências, desenvolvendo aplicativos móveis que simulam experimentos científicos para alunos do ensino médio. Os autores relatam que a criação de aplicativos ajudou os alunos a compreender melhor os conceitos científicos e a se envolverem mais com a aprendizagem.

Esses exemplos demonstram que a plataforma thunkable pode ser uma ferramenta útil e eficaz para o ensino de diversas disciplinas na educação básica, incluindo matemática e ciências.

2.1 Objetivos

Geral

Apresentar um roteiro de aprendizagem matemática para o Ensino Médio usando a programação visual em blocos

Específicos

Aperfeiçoar o raciocínio lógico e o conhecimento de lógica de programação através da criação de aplicativos;

Desenvolver aplicativos matemáticos para dispositivos móveis usando a plataforma Thhunkable;

3 METODOLOGIA

O desenvolvimento deste trabalho seguirá alguns passos básicos, visando o êxito nos objetivos a alcançar.

1. Seleção dos conteúdos matemáticos: Para começar, deve–se selecionar os assuntos matemáticos que serão abordados no aplicativo. Pode–se optar por conteúdos como geometria, álgebra, estatística, entre outros. É importante observar que todos os conteúdos que compõem o currículo de matemática da educação básica podem ser abordados por meio de aplicativos. Sendo assim, o número de possibilidades é enorme.
2. Definição do público–alvo: É importante definir o público–alvo para o qual o aplicativo será direcionado. Dependendo do público, pode–se optar por uma abordagem mais lúdica ou mais desafiadora. É preciso avaliar as características de todos os alunos que participarão da confecção de aplicativos. Tais aplicativos podem ser desenvolvidos em níveis diferentes de dificuldade e complexidade. Isso permitirá agrupar os alunos de acordo com o estágio de desenvolvimento que estejam.

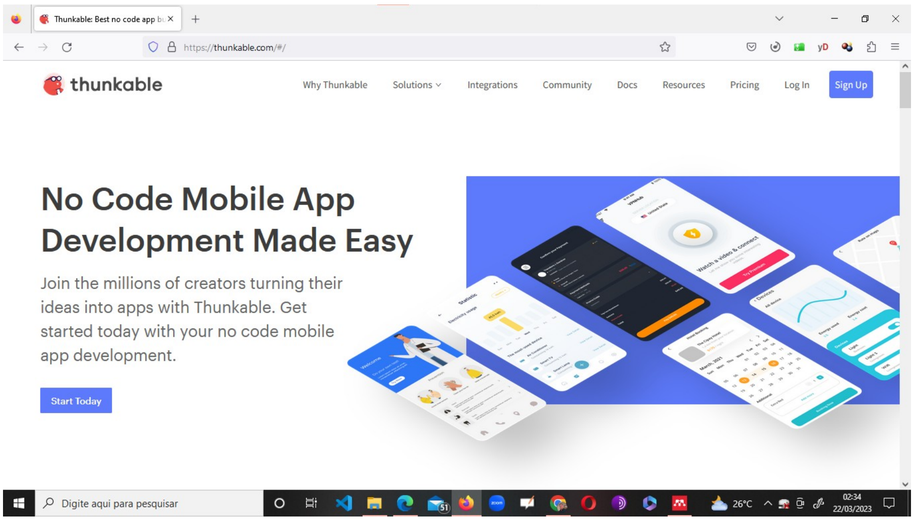
3. Escolha da abordagem pedagógica: A escolha da abordagem pedagógica é fundamental para definir o tipo de atividades e interações que serão disponibilizadas no aplicativo. Deve–se buscar uma abordagem que estimule o pensamento crítico e a resolução de problemas. Apresentar os projetos dos aplicativos na forma de desafios, uso de estratégias de gamificação. Tudo que puder estimular o interesse e a curiosidade dos alunos.
4. Criação do projeto no Thunkable: Após a definição dos conceitos matemáticos, do público–alvo e da abordagem pedagógica, é hora de criar o projeto no Thunkable. Aqui se dará o primeiro contato com a plataforma. Os alunos deverão criar suas contas de acesso e conhecer o ambiente de desenvolvimento do software. Nesta etapa, deve–se escolher os elementos que serão utilizados no aplicativo, como botões, imagens, caixas de texto, entre outros.
5. Desenvolvimento do aplicativo: Com o projeto criado, é hora de desenvolver o aplicativo. Aqui os alunos podem trabalhar em grupos ou individualmente para desenvolver as funcionalidades e conteúdos do aplicativo.
6. Testar o aplicativo: O sexto passo é testar o aplicativo. É importante que os alunos testem o aplicativo em diferentes dispositivos móveis para verificar se ele funciona corretamente e se atende aos objetivos de aprendizagem definidos anteriormente. É também necessário que procurem simular as mais diversas situações de modo a buscar eventuais falhas na concepção do aplicativo. Feitos todos os testes e solucionados os problemas, o aplicativo deve estar pronto para ser apresentado.

7. Apresentar o aplicativo: O último passo é apresentar o aplicativo. Os alunos podem apresentar o aplicativo para a turma ou para outros professores, compartilhando o que aprenderam durante o processo de criação e desenvolvimento. É possível também a realização de um evento ao fim do período de desenvolvimento do trabalho, seja semestral ou anual. Os aplicativos desenvolvidos no Thunkable podem ser publicados tanto na Play Store, para dispositivos que usem o sistema Android, da Google, quanto na Apple Store, no caso de dispositivos que usem o sistema IOs. Para tanto, é necessário a publicação do aplicativo nessas lojas. Isso exige uma conta de desenvolvedor. É possível instalar diretamente o aplicativo, bastando fazer o download do aplicativo e sua instalação. No entanto, essa é uma prática bastante arriscada, que pode trazer danos aos aparelhos, não sendo, portanto, recomendada.

4 CONHECENDO O THUNKABLE

Acesse o endereço eletrônico **thunkable.com**

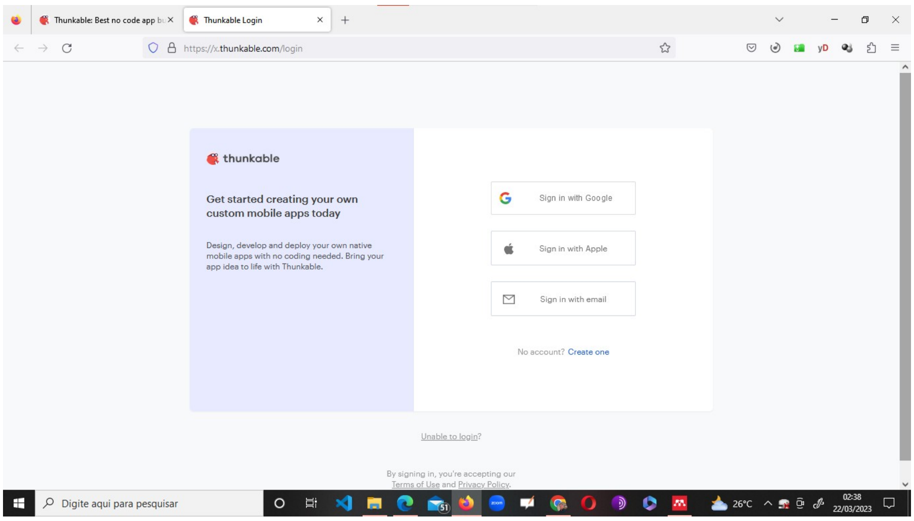
Figura 1 – Página inicial do thunkable



Fonte: elaborada pelo autor.

Clique em **Log in**

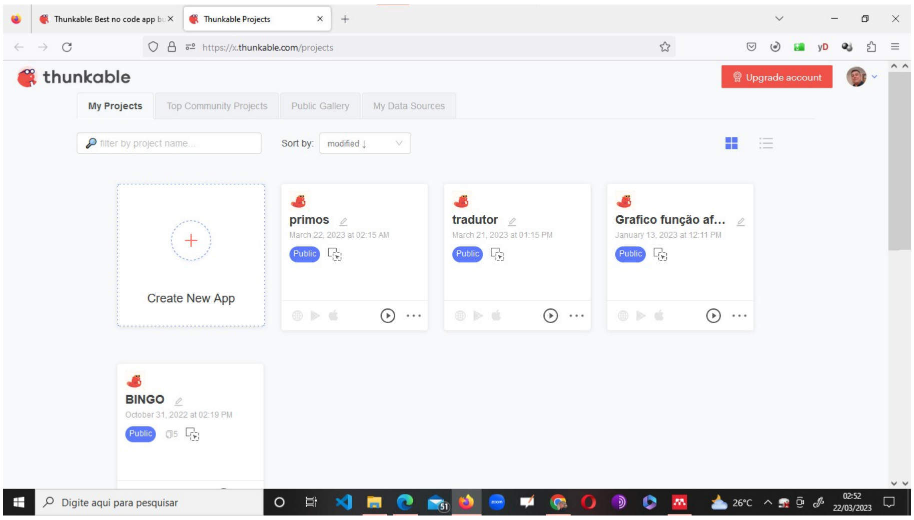
Figura 2 – Tela de login do thunkable



Fonte: elaborada pelo autor.

É possível entrar com uma conta Google, uma conta Apple ou criar uma conta com um e-mail qualquer.

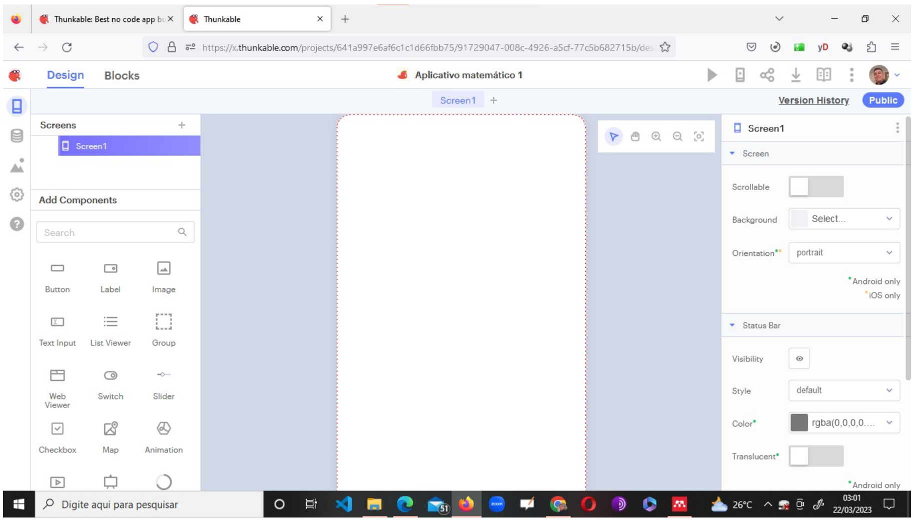
Figura 3 – Tela de inicial do usuário



Fonte: elaborada pelo autor.

Na tela inicial do usuário aparecem todos os aplicativos (projetos) criados. Para criar um aplicativo (projeto) deve–se clicar no “+”. Logo em seguida aparecerá a tela inicial do aplicativo, a área de **design**.

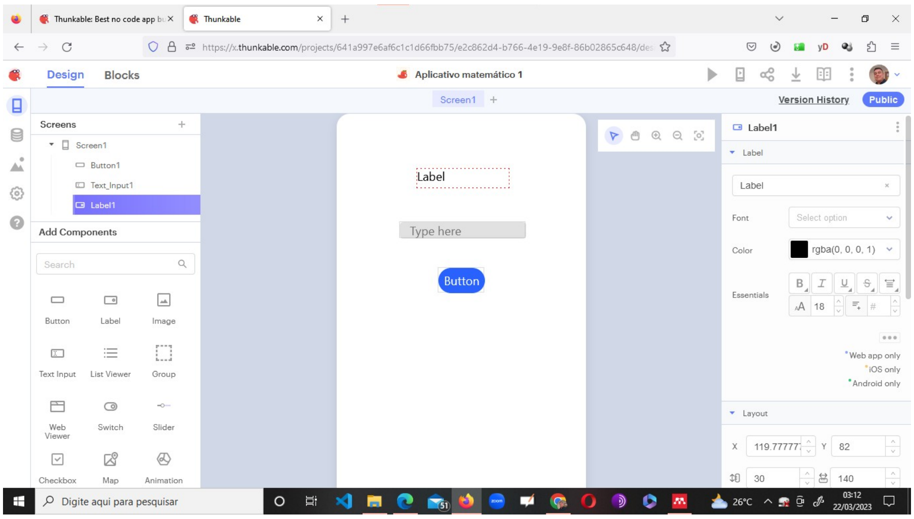
Figura 4 – Tela de inicial do aplicativo



Fonte: elaborada pelo autor.

Na esquerda temos os componentes que podem ser utilizados no aplicativo: botão, rótulo, entrada de texto etc. Na região central da tela temos o simulador do dispositivo, onde colocaremos os componentes para a criação do “visual” do aplicativo. Ele já vem com o nome **Screen1** “tela 1”. Na direita temos as propriedades dos componentes do nosso aplicativo. De início, aparecem as propriedades da Screen1. É possível alterar todas essas propriedades, tanto diretamente na coluna quanto por meio de programação nos blocos.

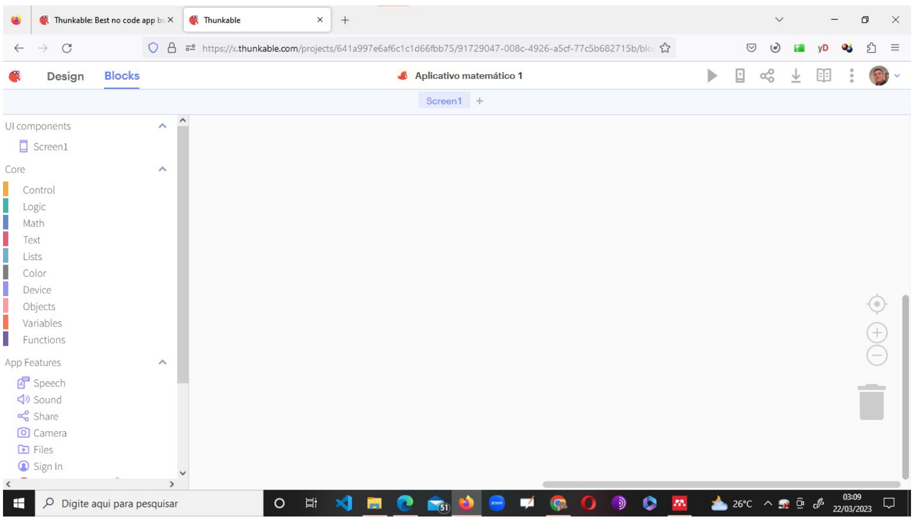
Figura 5 – Inserindo componentes label, text_input e buttom



Fonte: elaborada pelo autor.

A título de exemplo, foram inseridos três componentes: um **Label** (rótulo), um **Text_input** (entrada de texto) e um **Button** (botão). Para programar o comportamento dos componentes, devemos clicar na aba **Blocks**.

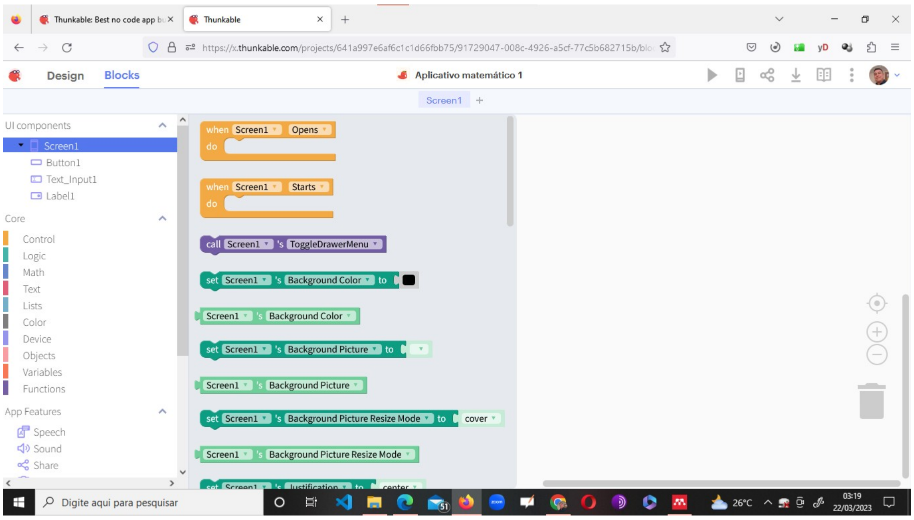
Figura 6 – Área de blocos do aplicativo



Fonte: elaborada pelo autor.

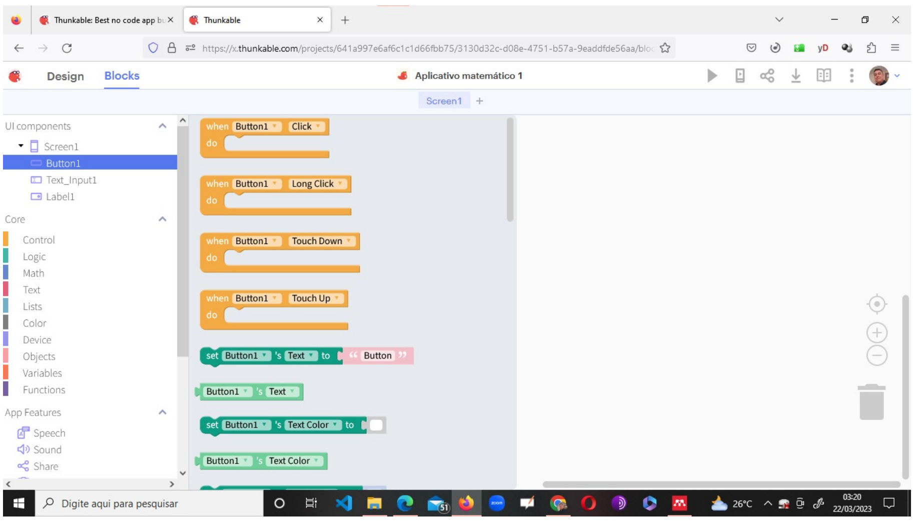
Nessa área, podemos ver os componentes e os blocos de eventos e propriedades de cada um.

Figura 7 – Blocos de eventos e propriedades da screen1



Fonte: elaborada pelo autor.

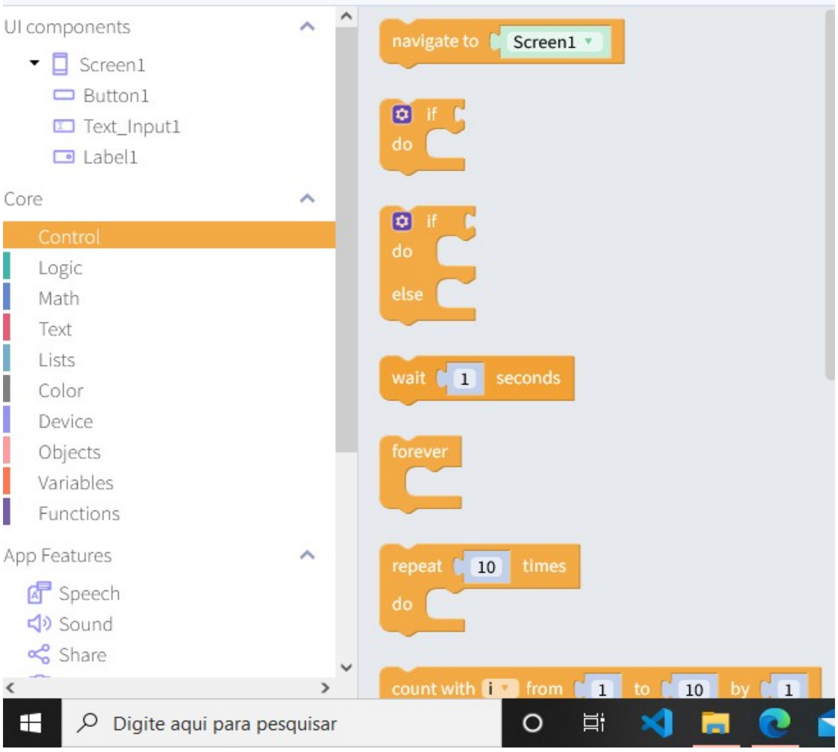
Figura 8 – Blocos de eventos e propriedades do buttom1



Fonte: elaborada pelo autor.

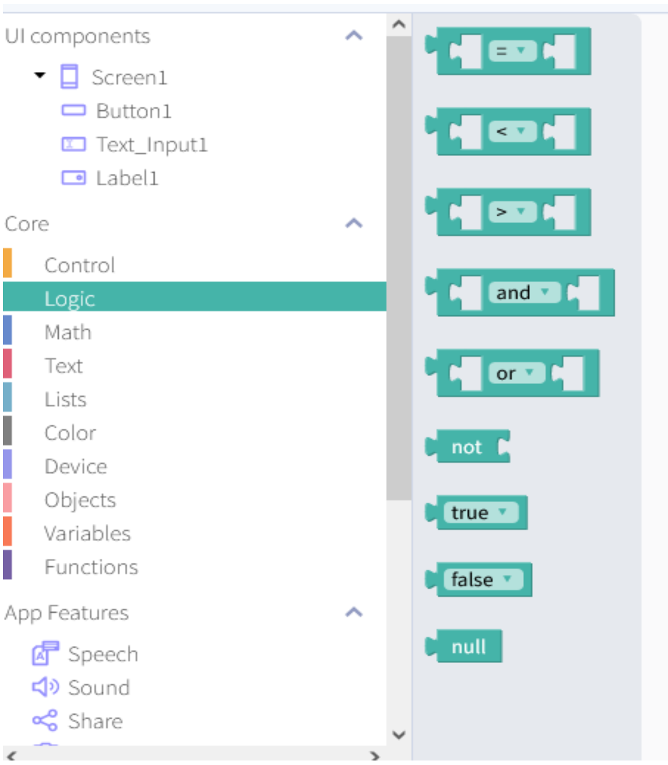
Além dos blocos relativos aos componentes do nosso aplicativo, encontramos grupos de blocos com diversas funcionalidades, tais como: **Controle**, **Lógico**, **Matemático**, **Texto**, **Listas**, **Cores**, **Dispositivo**, **Objetos**, **Variáveis** e **Funções**.

Figura 9 – Blocos de controle



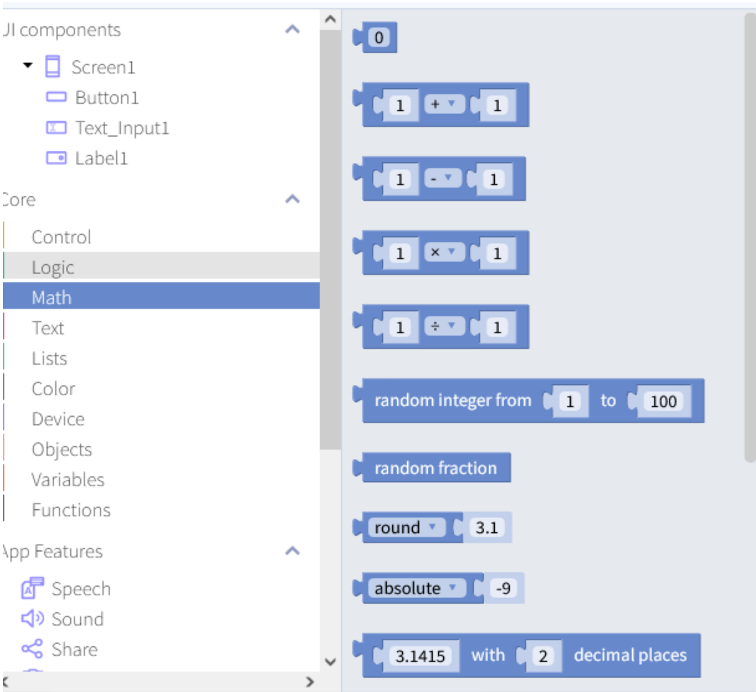
Fonte: elaborada pelo autor.

Figura 10 – Blocos lógicos



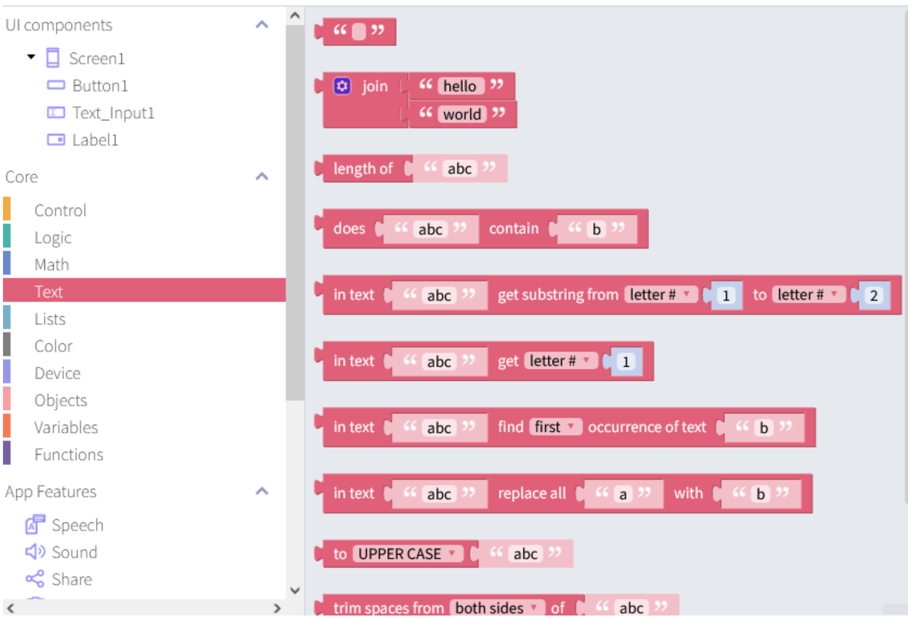
Fonte: elaborada pelo autor.

Figura 11 – Blocos matemáticos



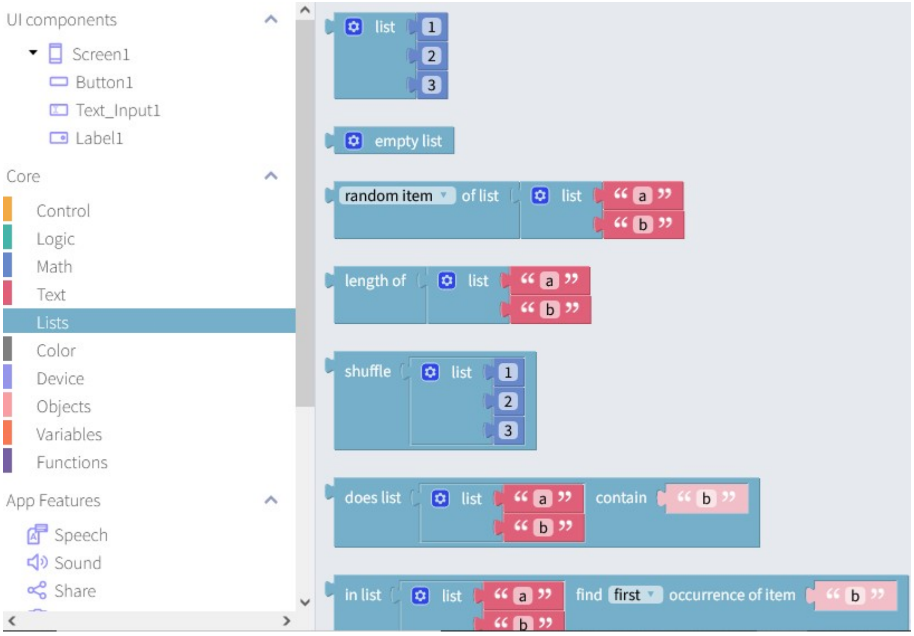
Fonte: elaborada pelo autor.

Figura 12 – Blocos de texto



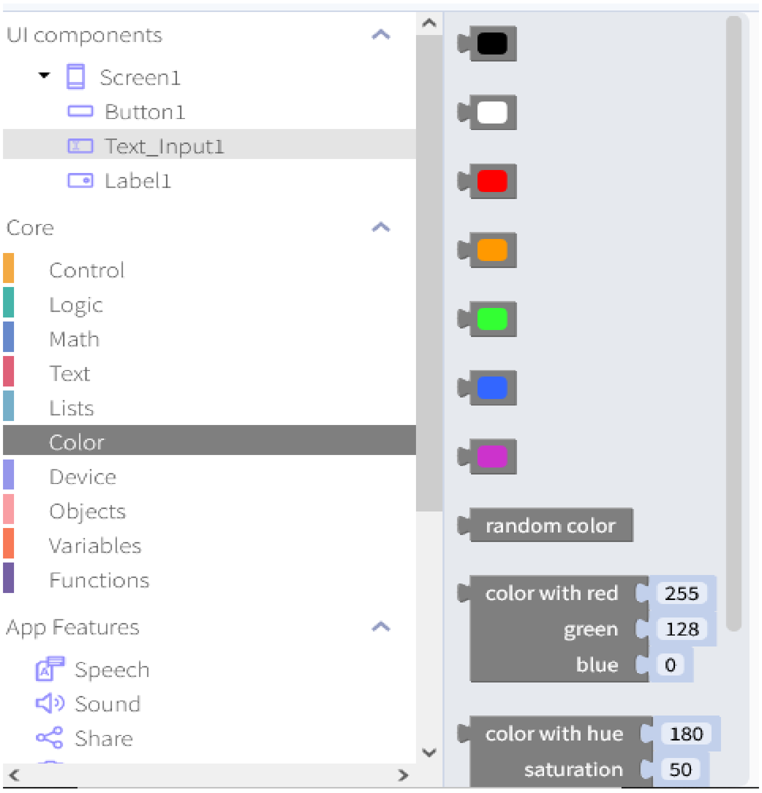
Fonte: elaborada pelo autor.

Figura 13 – Blocos de listas



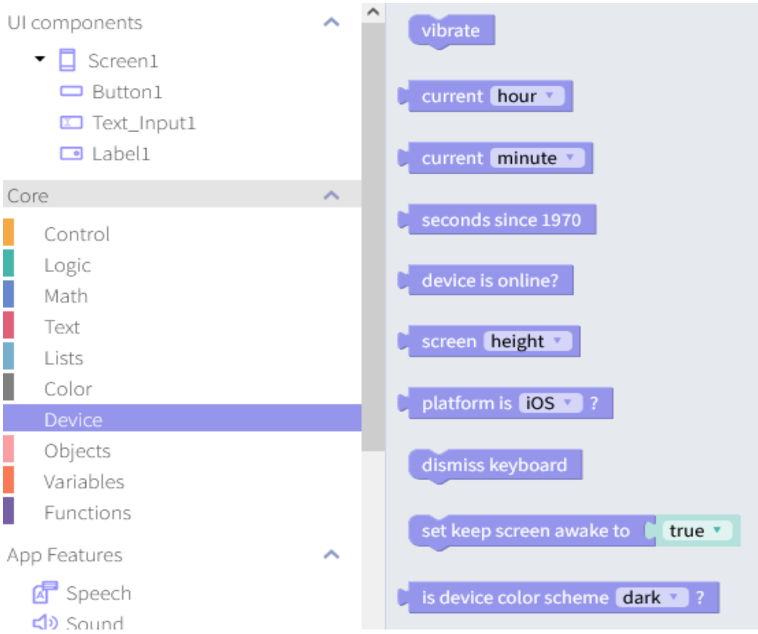
Fonte: elaborada pelo autor.

Figura 14 – Blocos de cores



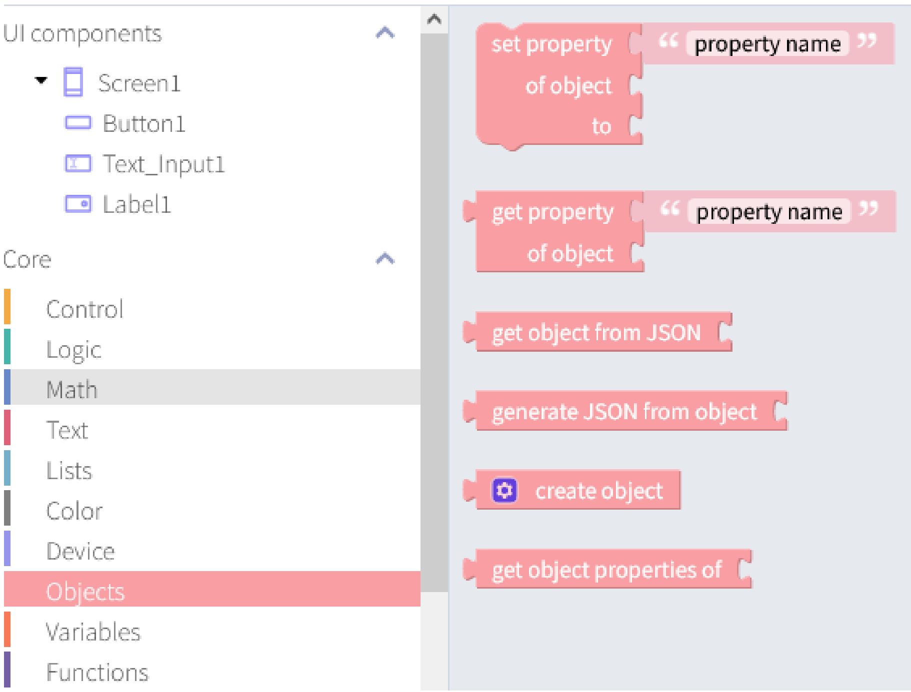
Fonte: elaborada pelo autor.

Figura 15 – Blocos de recursos do dispositivo móvel



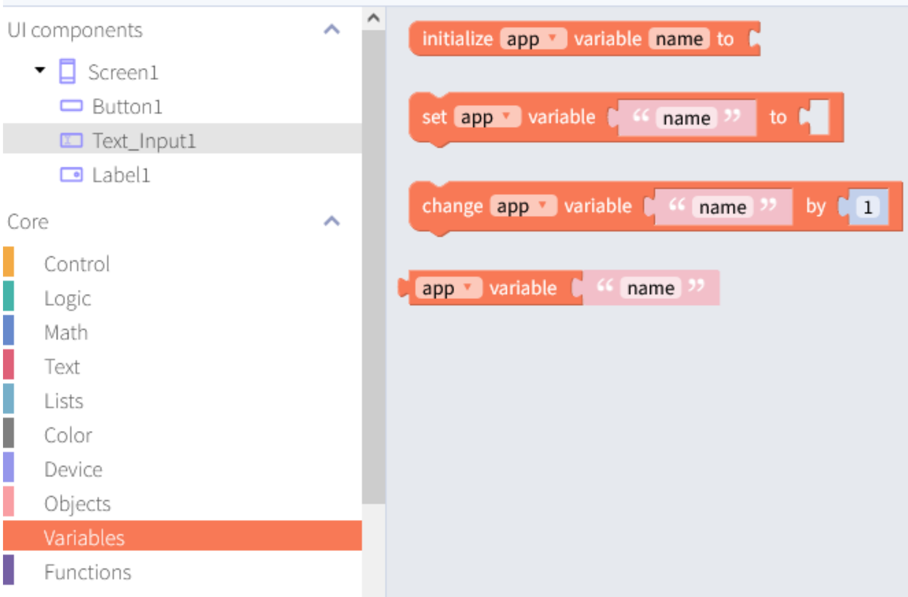
Fonte: elaborada pelo autor.

Figura 16 – Blocos de manipulação de objetos



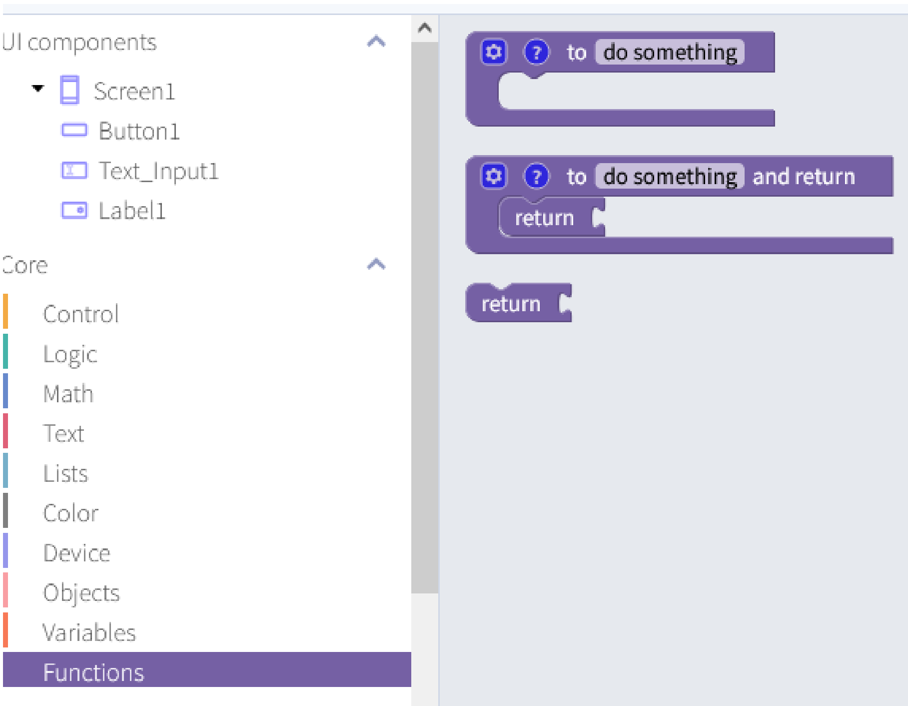
Fonte: elaborada pelo autor.

Figura 17 – Blocos de variáveis



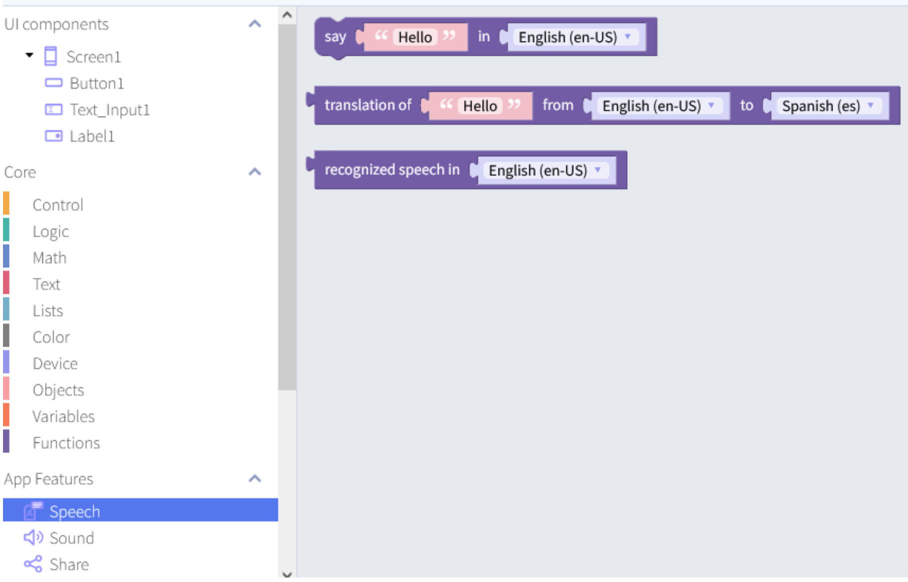
Fonte: elaborada pelo autor.

Figura 18 – Blocos de funções



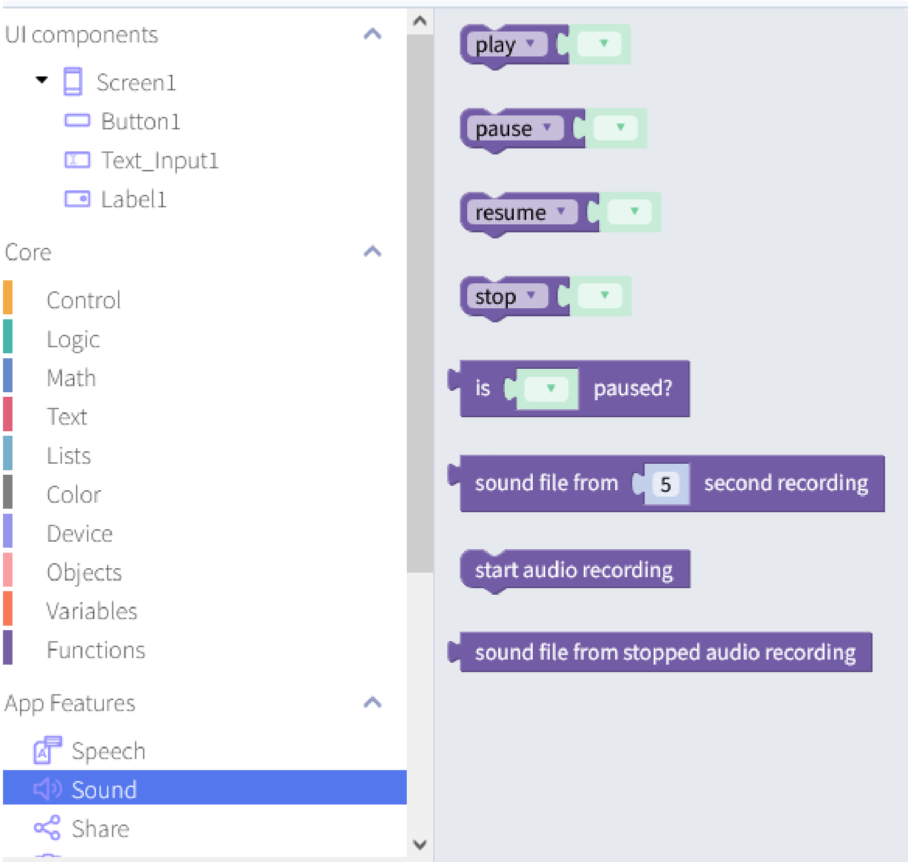
Fonte: elaborada pelo autor.

Figura 19 – Blocos de fala



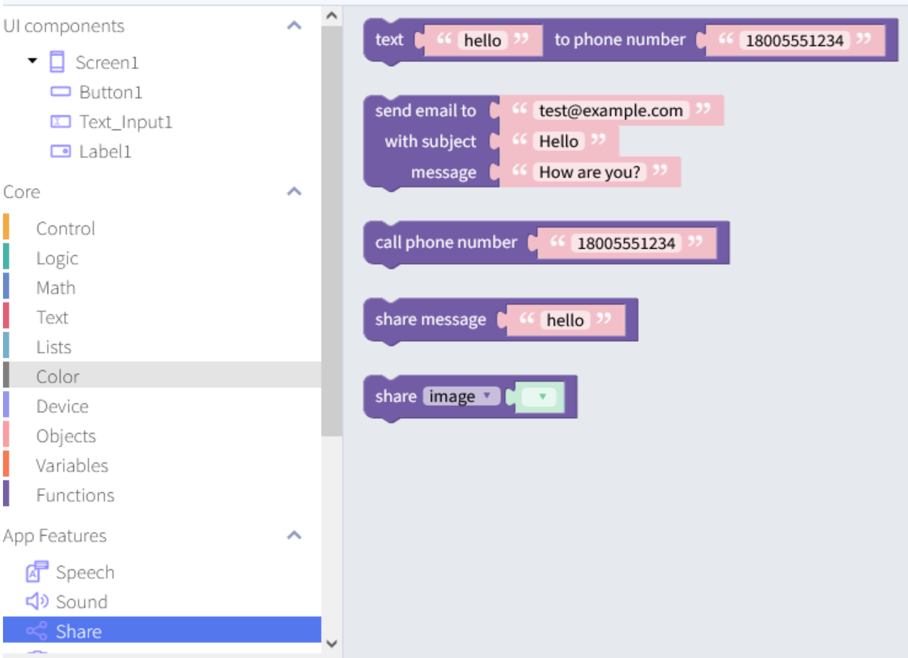
Fonte: elaborada pelo autor.

Figura 20 – Blocos de sons



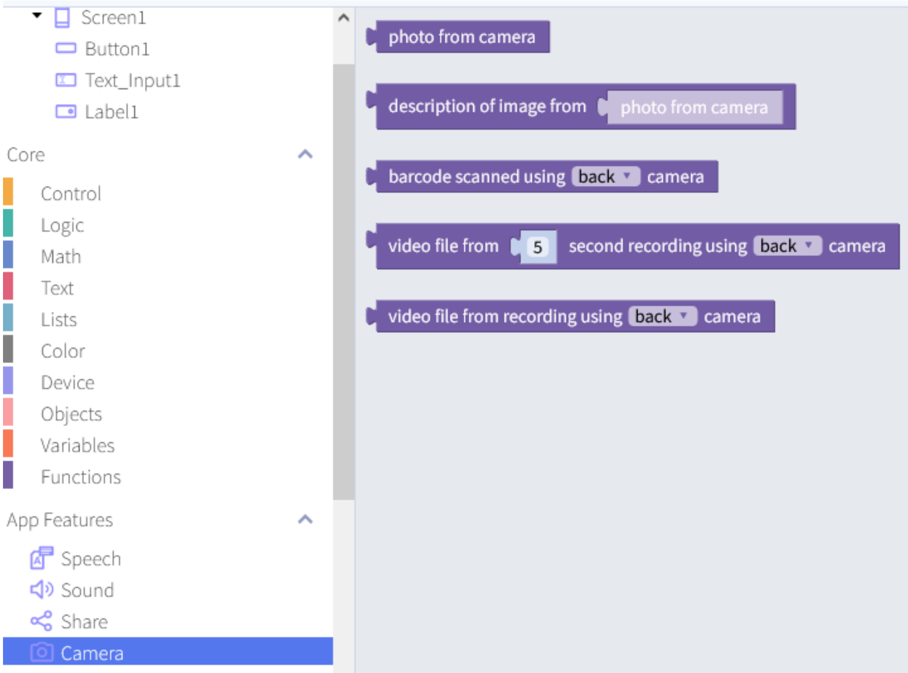
Fonte: elaborada pelo autor.

Figura 21 – Blocos de mensagens e compartilhamento



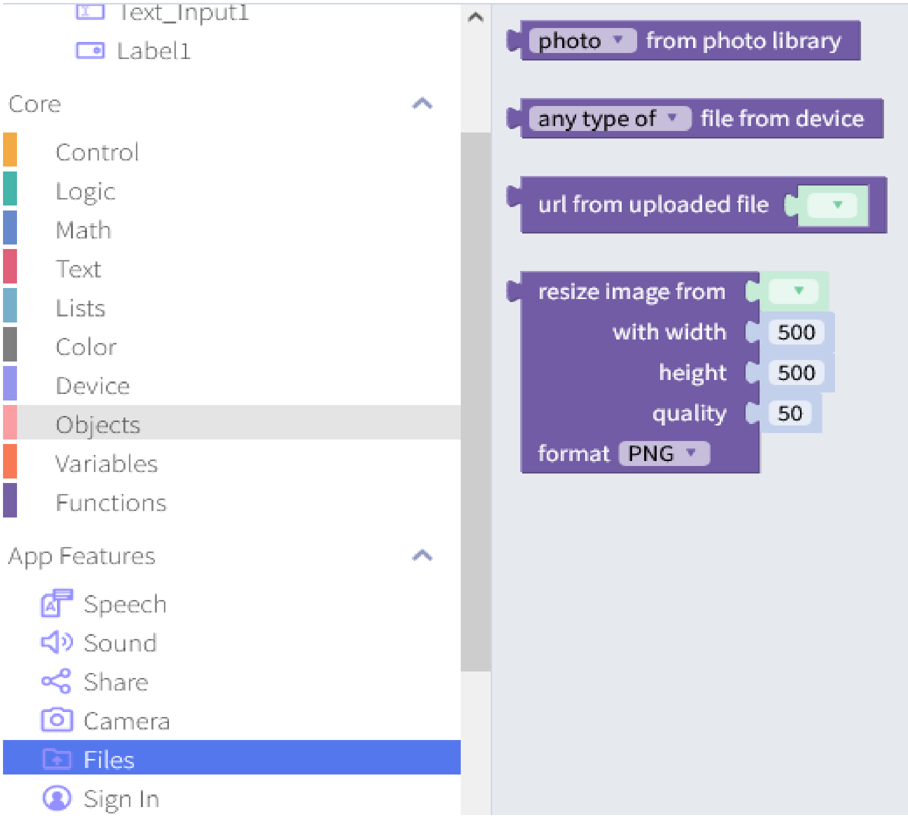
Fonte: elaborada pelo autor.

Figura 22 – Blocos de manipulação de câmera



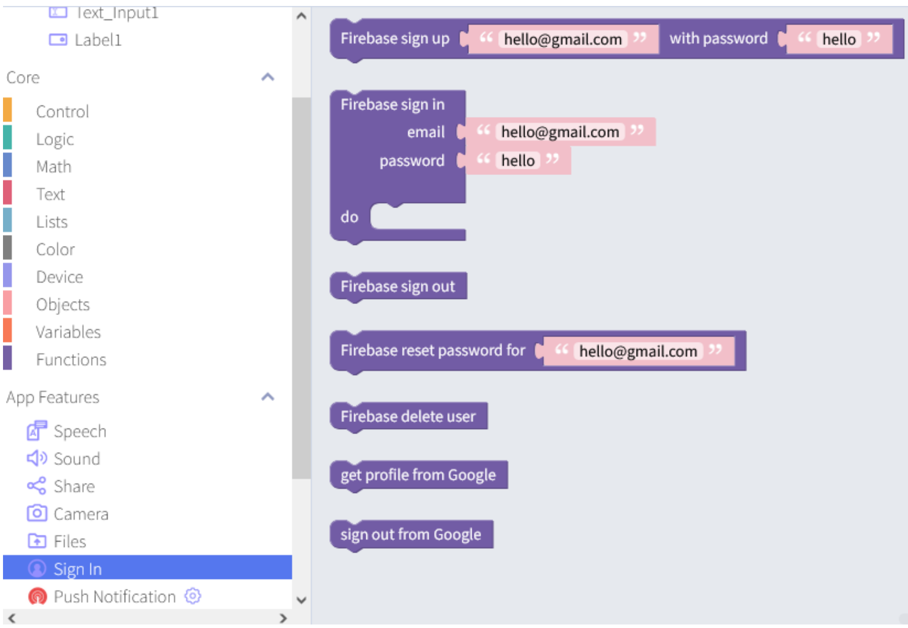
Fonte: elaborada pelo autor.

Figura 23 – Blocos de manipulação de arquivos



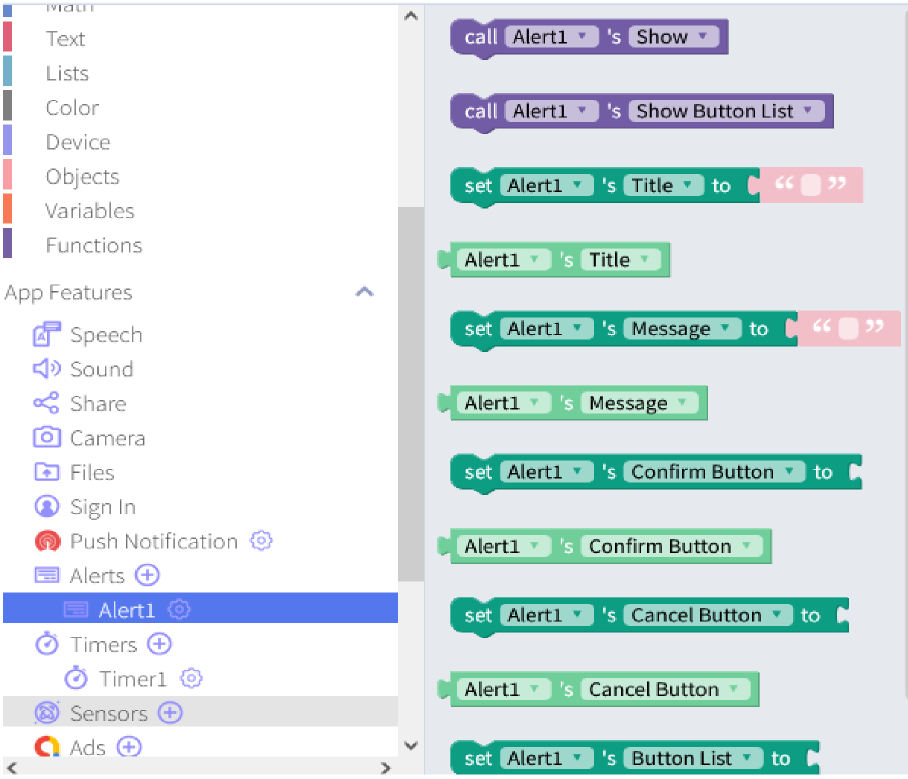
Fonte: elaborada pelo autor.

Figura 24 – Blocos de firebase



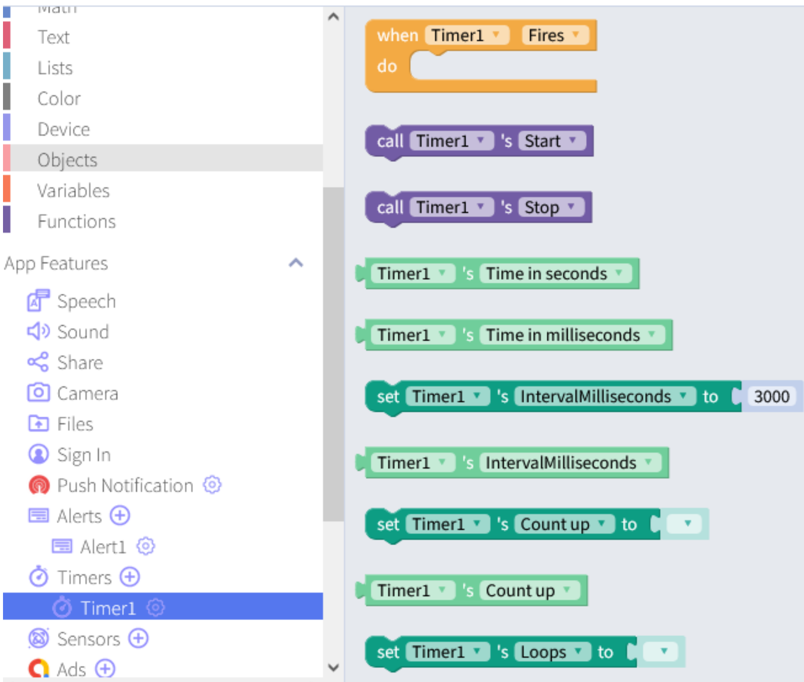
Fonte: elaborada pelo autor.

Figura 25 – Blocos de alertas



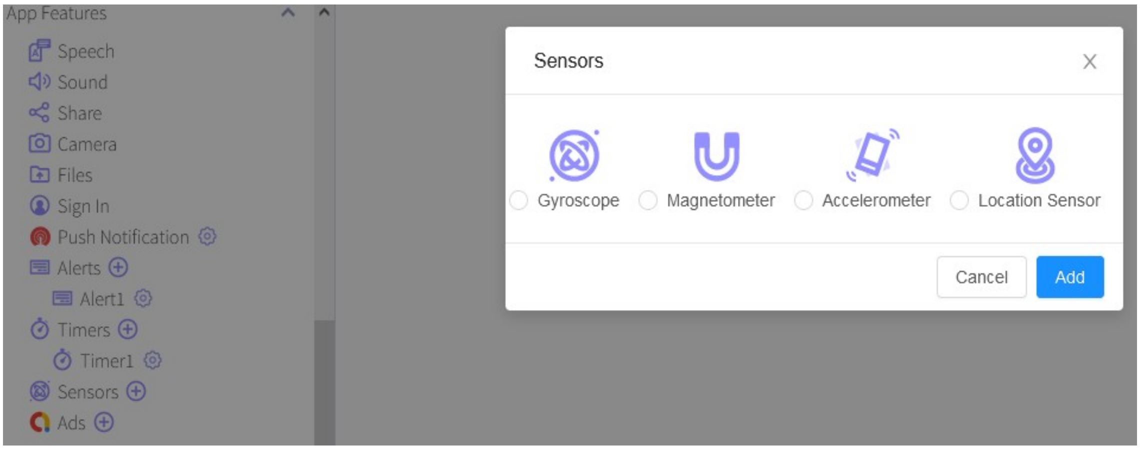
Fonte: elaborada pelo autor.

Figura 26 – Blocos de temporizador



Fonte: elaborada pelo autor.

Figura 27 – Sensores



Fonte: elaborada pelo autor.

5. APLICATIVO 1: NÚMEROS PRIMOS

Esse aplicativo informa se um número é primo ou composto, mostrando todos os divisores do número.

5.1 Múltiplos e divisores

Conforme o Algoritmo da Divisão em Z, ao se dividir a por b, conseguimos únicos q e r tais que a = bq + r.

Nesta seção estamos interessados, particularmente, no caso em que z = 0.

Definição: Sejam a, b $\in$ Z com b $\neq$ 0. Dizemos que b é divisor de a, quando existe n $\in$ Z tal que a = bn. A notação usada para dizer que b é divisor de a é b|a. Pela definição acima, n também é um divisor de a. Segue também que, quando b é divisor de a, |b| $\leq$ |a|.

Observação: De forma análoga, podemos definir a divisão em N. Sejam a, b $\in$ N com b $\neq$ 0. Dizemos que b é divisor de a quando existe n $\in$ N tal que a = nb.

Exemplo: Vejamos alguns exemplos.

1. O número 48 é divisor de 144 pois 144 = 48 . 3.
2. Como 42 = 77 . 6, temos 7|42.
3. Em Z, – 15 é divisor de 90 pois 90 = (– 15) . (– 6).

Pelos exemplos vistos, 48|144 e (– 15)|90. Quando a divisibilidade não ocorre, denotamos por $b \nmid a$. Por exemplo, $3 \nmid 8$. Simbolicamente temos $\forall$ a, b $\in$ Z, b|a $\Leftrightarrow$ $\exists$ n $\in$ Z : a = bn.

São equivalentes as seguintes sentenças:

1. b é divisor de a.
2. b divide a.
3. a é divisível por b.
4. a é múltiplo de b.

Observação: Vamos analisar cuidadosamente o caso em que fosse permitido $b = 0$.

1. Se $a \neq 0$, temos que $a = 0 . n$ nunca acontece. Assim, concluímos que $0 \nmid a$, $a \in Z^*$.
2. Se $a = 0$, vale $0|0$, afinal $0 = 0 . n$ é verdade $\forall\ n \in Z$. Porém, o fato de n poder assumir qualquer valor, não podemos definir o que seria “0 dividido por 0”.

Vejamos algumas propriedades a respeito da divisibilidade.

Proposição 1: Sejam a, b, $c \in Z^*$.

1. $a|a$
2. $b|a$ e $a|b$ $a = b$ ou $a = -b$.
3. $c|b$ e $b|a$ $c|a$.

Demonstração:

1. De fato, $a = a . 1$.
2. Por definição, a hipótese $b|a$ implica que $a = bn$ para algum número inteiro n. Assim como $a|b$ implica $b = am$ para algum inteiro m.

Então $a = amn$, e, como $a \neq 0$, pela propriedade do cancelamento obtemos $1 = mn$. No conjunto dos números inteiros temos, portanto, que $n = m = 1$ ou $n = m = -1$. Logo, $a = b$ ou $a = -b$, respectivamente.

3. A hipótese $c|b$ significa, por definição, que existe número inteiro n tal que $b = cn$, assim como $b|a$ implica a existência de um número inteiro m tal que $a = bm$. Daí obtemos $a = (cn)m = c(nm)$ e disto segue que $c|a$.

Essa proposição nos diz que a divisibilidade pode ser vista como uma relação a) reflexiva e c) transitiva.

Exemplo: Pelo item c) da proposição anterior, $-4|8$ e $8|24$ implicam que $-4|24$.

Existem algumas importantes propriedades que a divisibilidade satisfaz, quando relacionada com as operações em Z.

Proposição 2: Sejam a, b, $c \in Z$.

1. $c|a \Rightarrow c|ab$ e $cb|ab$.
2. $c|a$ e $c|b \Rightarrow c|a + b$ e $c|a - b$.
3. $c|a + b$ e $c|a \Rightarrow c|b$.

Demonstração:

1. De $c|a$ segue que existe $n \in Z$ tal que $a = cn$ e, portanto, $ab = cbn$. Daí, obtemos $c|ab$ e $cb|ab$.
2. De $c|a$ e $c|b$ concluímos a existência de $n, m \in Z$ tais que $a = cn$ e $b = cm$. Logo $a + b = c(n + m)$ e $a – b = c(n – m)$ e o resultado segue.
3. De $c|a + b$ e $c|a$ temos $n, m \in Z$ tais que $a + b = cn$ e $a = cm$. Logo $b = a + b – a = c(n – m)$ e, portanto, $c|b$.

Exemplo: Pela proposição anterior, valem as seguintes implicações.

1. $3|21$ $3|(21 . 2)$ $3|42$.
2. $– 5|15$ $– 5(– 6)|(15(– 6))$ $30|– 90$.
3. $7|21$ e $7|– 42$ $7|(21 + (– 42))$ $7|– 21$.
4. $– 10|30$ e $– 10|80$ $– 10|(30 – 80)$ $– 10|– 50$.
5. $8|(24 + 32)$ e $8|24$ $8|32$.
6. $2|(4 + 40)$ e $2|4$ $2|40$.

Para finalizar esta seção, apresentamos uma definição de terminologia que envolve números inteiros e seus divisores.

Definição: Dado $a \in Z$, os números 1 e a são ditos divisores impróprios de a. Se existirem outros divisores, estes serão chamados divisores próprios de a.

5.2 Números primos

Vejamos o que são números primos, tanto em N quanto em Z.

Definição: Um número natural p é primo quando:

1. $p \neq 0$ e $p \neq 1$.
2. Os únicos divisores de p são 1 e p.

Definição: Um número inteiro p é primo quando $|p|$ é primo em N.

Um inteiro que não seja primo é chamado de número composto. Para descobrir se um dado número p é primo, temos de testar sua divisibilidade quanto aos números estritamente entre 1 e p. Contudo, pelo item c) da proposição 4.1, é necessário apenas checar sua divisibilidade pelos números primos entre 1 e p.

Esse processo, ou algoritmo, possui um nome especial: crivo de Eratóstenes. Além de determinar se um dado número é primo, esse crivo também nos fornecerá todos os números primos entre 1 e aquele número.

Observação: Os únicos números primos pares em Z são 2 e –2, afinal, todo outro número par possui 2 como divisor.

Proposição: Se a $\in$ Z com 1 < |a|, então a admite pelo menos um divisor primo.

Proposição: Se a $\in$ Z, com a > 1, é um número composto, então a é múltiplo de algum primo p tal que $p^2 \leq a$.

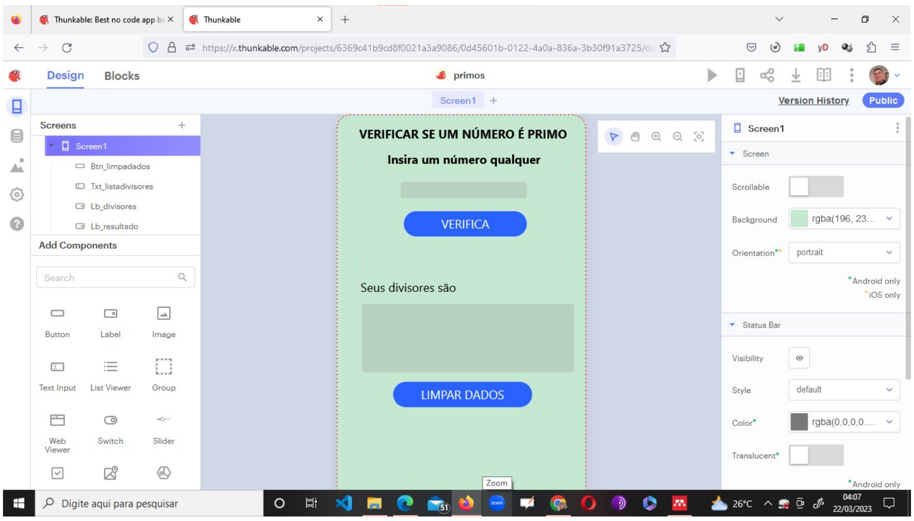
Proposição: (Lema de Euclides) Sejam a, b, p Z com p primo. Se p divide ab então p divide a ou p divide b.

Agora que temos algumas noções sobre divisores, múltiplos e número primos, vamos ao desenvolvimento doa plicativo que verifica se um número dado é primo ou composto.

5.3 DESENVOLVIMENTO DO APLICATIVO

Na tela inicial do usuário, clicar em “+” para criar um aplicativo. Dar o nome de “PRIMOS”, selecionar a categoria “Educação” e clicar no botão “Criar”. Na tela inicial do aplicativo, inserir quatro componentes **Label**, dois componentes **Button** e dois componentes **Text Input**. Renomear os componentes e a propriedade **Text**, conforme a figura abaixo.

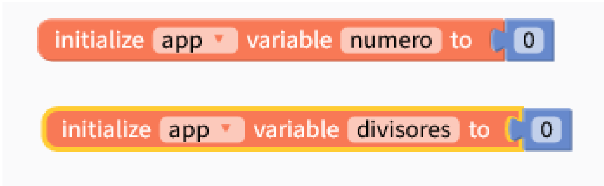
Figura 28 – Tela inicial do aplicativo números primos



Fonte: elaborada pelo autor.

Na área de blocos, criar duas variáveis, conforme a figura.

Figura 29 – variáveis número e divisores



Fonte: elaborada pelo autor.

No componente **Btn_verifica**, selecionar o bloco representado na figura abaixo:

Figura 30 – Bloco de ações para quando o componente Btn_verifica for clicado

Fonte: elaborada pelo autor.

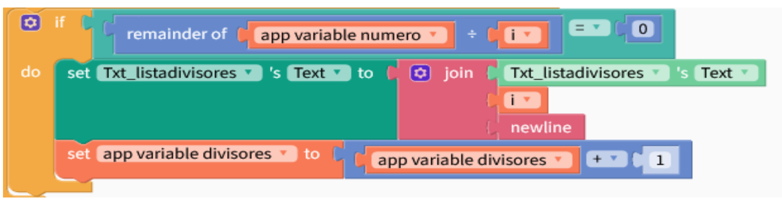
Atribuir à **variável número** o conteúdo do componente **Txt_insere**:

Figura 31 – Atribuindo um valor à variável número

Fonte: elaborada pelo autor.

Atribuir ao conteúdo do componente **Txt_listadivisores** o valore da variável **i,** caso o resto da divisão da **variável número** por **i** seja igual a zero.

Figura 32 – Atribuindo um valor ao componente Txt_listadivisores



Fonte: elaborada pelo autor.

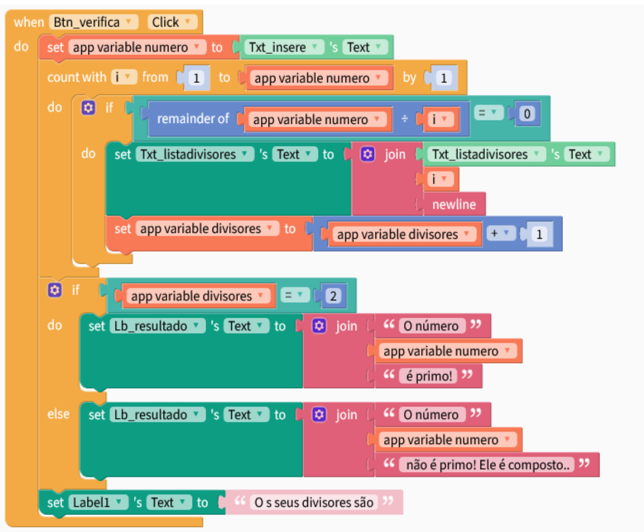
Se o valor da variável **divisores** for igual a 2, atribuir ao componente **Lb_resultado** o conteúdo "O número " + variável número + "é primo". Caso contrário, atribuir ao componente **Lb_resultado** o conteúdo "O número " + variável número + "não é primo! Ele é composto!"

Figura 33 – Verificando se o número é primo ou composto

Fonte: elaborada pelo autor.

De modo que os blocos inseridos dentro do bloco when **Btn_verifica** click ficam:

Figura 34 – Imprimindo a lista de divisores



Fonte: elaborada pelo autor.

Para finalizar, usamos o bloco when **Btn_limpadados** click para apagar os conteúdos dos componentes e variáveis:

Figura 35 – Limpando os conteúdos dos componentes e zerando as variáveis

Fonte: elaborada pelo autor.

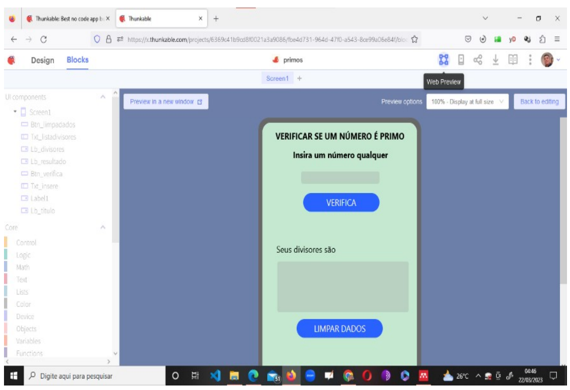
Finalizada a programação do aplicativo, devem ser feitos os testes para ver o seu funcionamento. Para isso, é só clicar no símbolo de **web preview**, que se encontra no menu superior do lado direito:

Figura 36 – Botão de web preview do aplicativo

Fonte: elaborada pelo autor.

Deverá aparecer a seguinte tela:

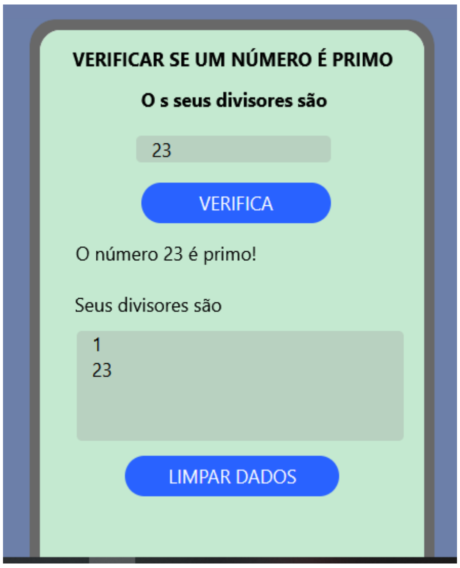
Figura 37 – Testando o aplicativo no web preview



Fonte: elaborada pelo autor.

Vamos digitar um número primo como, por exemplo, **23**, e clicar em verifica:

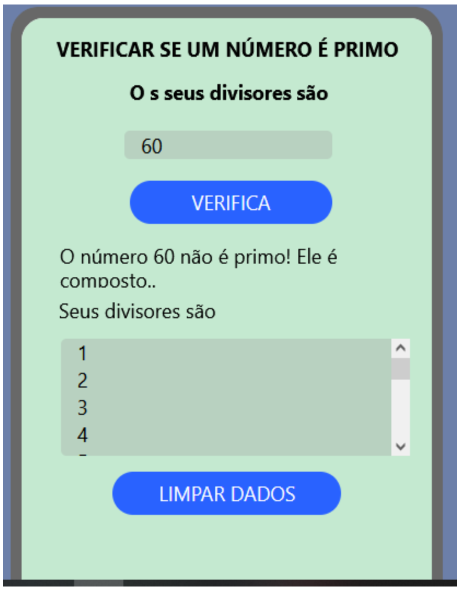
Figura 38 – Verificando se o número 23 é primo



Fonte: elaborada pelo autor.

Vamos digitar um número par como, por exemplo, **60**, e clicar em verifica:

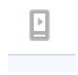
Figura 39 – Verificando se o número 60 é primo



Fonte: elaborada pelo autor.

Como é possível ver, o aplicativo está funcionando bem. É necessário fazer diversos outros exemplos, de modo a garantir que não haja erros. É possível também testar o aplicativo em um dispositivo móvel. Para isso, é preciso fazer o download do app do Thunkable. Depois, basta clicar no símbolo de teste no dispositivo:

Figura 40 – Botão de teste do aplicativo no dispositivo móvel



Fonte: elaborada pelo autor.

6 APLICATIVO 2: GRÁFICOS DE FUNÇÕES

Esse aplicativo lê um número inteiro e informa se ele é primo ou composto, bem como lista os seus divisores positivos.

6.1 Função Linear

Uma função de R em R recebe o nome de função linear quando a cada elemento de x $\in$ R associa o elemento ax $\in$ R em que a $\neq$ 0 é um número real dado, isto é: f(x) = ax.

6.1.1 Gráfico

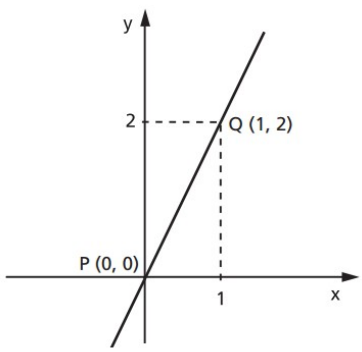
O gráfico da função linear é uma reta que passa pela origem.

Exemplos:

1. Construir o gráfico da função y = 2x.

Considerando que dois pontos distintos determinam uma reta e no caso da função linear um dos pontos é a origem, basta atribuir a x um valor não nulo e calcular o correspondente y = 2x. Por exemplo, x = 1, teremos y = 2 . 1 = 2.

Figura 41 – Gráfico da função y = 2x



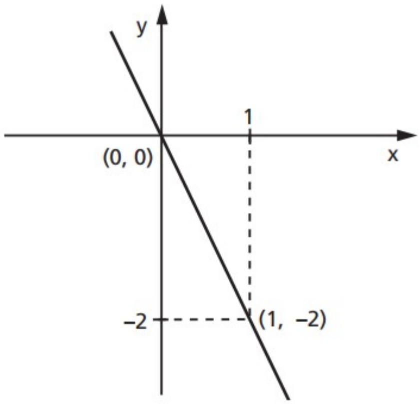
Fonte: Iezzi, 2013

Pelos pontos P(0,0) e Q(1,2) traçamos a reta $\overleftrightarrow{PQ}$, que é precisamente o gráfico da função dada.

2. Construir o gráfico da função y = –2x.

Analogamente, temos: x = 1, y = –2 . 1 = –2.

Figura 42 – Gráfico da função y = –2x



Fonte: Iezzi, 2013

6.2 Função Afim

Uma função de R em R recebe o nome de função afim quando a cada x $\in$ R associa sempre o mesmo elemento (ax + b) $\in$ R em que a $\neq$ 0 e b são números reais dados, isto é: f(x) = ax + b.

Exemplos:

1. y = 3x + 2
2. y = –2x + 1
3. y = x – 3
4. y = 4x

Podemos notar que, para b = 0, a função afim y = ax + b se transforma na função linear y = ax; podemos, então, dizer que a função linear é um caso particular da função afim.

6.2.1 Gráfico

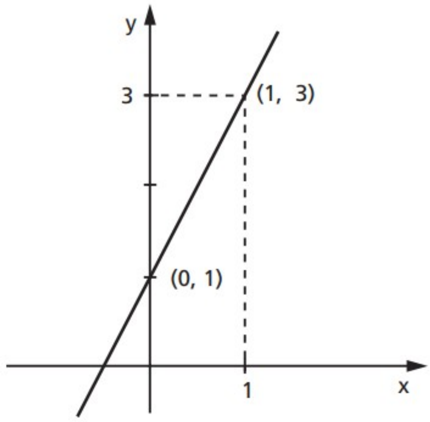
O gráfico da função é uma reta.

Exemplos:

1. Construir o gráfico da função y = 2x + 1.

Considerando que o gráfico da função afim é uma reta, vamos atribuir a dois valores distintos e calcular s correspondentes valores de y. Por exemplo: x = 0, y = 2 . 0 + 1 = 1 e x = 1, y = 2 . 1 + 1 = 3. Assim, o gráfico procurado é a reta que passa pelos pontos (0,1) e (1,3).

Figura 43 – Gráfico da função y = 2x + 1

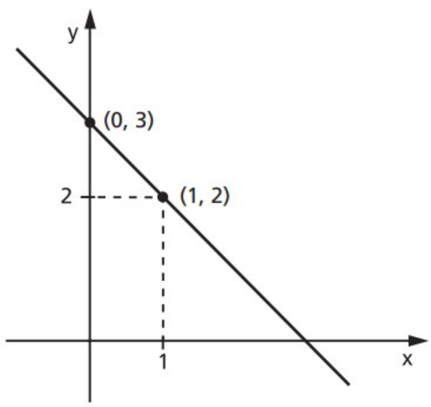


Fonte: Iezzi, 2013

2.Construir o gráfico da função y = – x + 3.

De modo análogo ao exemplo anterior, vamos atribuir a x dois valores distintos e obter os respectivos valores para y. Por exemplo: x = 0, y = – 0 + 3 = 3 e x = 1, y = – 1 + 3 = 2. Assim, o gráfico procurado é uma reta que passa pelos pontos (0,3) e (1,2).

Figura 44 – Gráfico da função y = –x + 3



Fonte: Iezzi, 2013

6.2.2 Sobre os coeficientes da função afim

O coeficiente a da função determina a inclinação da reta que representa a função no plano cartesiano. É chamado taxa de variação ou coeficiente angular da reta. Se $a > 0$, a reta tem inclinação positiva, ou seja, a função cresce à medida que x aumenta. Se $a < 0$, a reta tem inclinação negativa, e a função decresce à medida que x aumenta. Quando $a = 0$, a função se reduz a uma constante, ou seja, uma reta horizontal.

O coeficiente b da função determina o ponto em que a reta intercepta o eixo y (ou seja, quando $x = 0$). Esse ponto é chamado de ponto de interseção com o eixo y. Esse coeficiente é conhecido como valor inicial ou coeficiente linear da reta.

Uma das aplicações mais comuns da função polinomial do 1º grau é na modelagem de problemas de proporcionalidade direta, onde uma variável é proporcional à outra, com uma constante de proporcionalidade a. Nesse caso, a função é escrita como $f(x) = kx$, onde k é a constante de proporcionalidade.

Outra aplicação comum da função polinomial do 1º grau é na análise de dados financeiros, onde a função pode ser usada para modelar o crescimento linear de investimentos ou dívidas.

Alguns conceitos importantes relacionados à função polinomial do 1º grau incluem a determinação da raiz ou zero da função, que é o valor de x em que $f(x) = 0$. Isso pode ser encontrado resolvendo a equação $ax + b = 0$, o que resulta em $x = -b/a$.

Além disso, o coeficiente angular da função, que é a inclinação da reta que representa a função, pode ser encontrado calculando a variação de y dividida pela variação de x entre dois pontos quaisquer da reta.

6.3 Função Quadrática

Uma função f de R em R recebe o nome de função quadrática ou do 2º grau quando associa a cada $x \in R$ o elemento $(ax^2 + bx + c) \in R$, em que a, b e c são números reais dados e $a \neq 0$, isto é: $f(x) = ax^2 + bx + c$.

Exemplos:

1. $f(x) = x^2 - 3x + 2$

2. $f(x) = -3x^2 + 5x - 1$

3. $f(x) = x^2 - 4$

4. $f(x) = -3x^2$

6.3.1 Gráfico

O gráfico da função quadrática é uma parábola.

Exemplos:

1. Construir o gráfico de $y = x^2 - 1$.

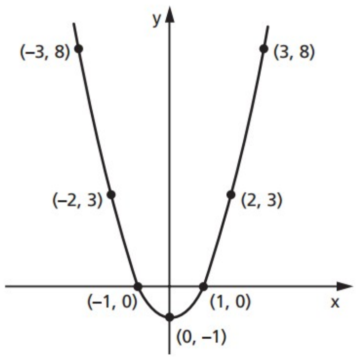
Vamos atribuir valores a x em um intervalo de – 3 a 3, por exemplo, para obtermos os respectivos valores de y.

Tabela 1 – Valores de x e y no intervalo de – 3 a 3

x	$y = x^2 - 1$
– 3	8
– 2	3
– 1	0
0	– 1
1	0
2	3
3	8

Fonte: elaborada pelo autor.

Figura 45 – Gráfico da função $y = x^2 - 1$



Fonte: Iezzi, 2013

2. Construir o gráfico de $y = -x^2 + 1$.

Analogamente ao exemplo anterior, vamos atribuir valores a x, no intervalo entre – 3 e 3 e obter os respectivos valores de y.

Tabela 2 – Valores de x e y no intervalo de – 3 a 3

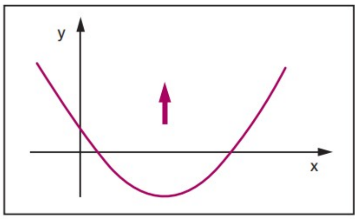
x	**$y = -x^2 + 1$**
– 3	– 8
– 2	– 3
– 1	0
0	1
1	0
2	– 3
3	– 8

Fonte: elaborada pelo autor.

6.3.2 Sobre os coeficientes da função quadrática

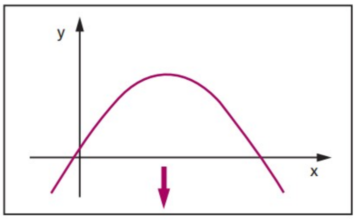
A parábola representativa da função quadrática $y = ax^2 + bx + c$ pode ter a concavidade voltda para cima ou para baixo. O coeficiente "a" determina a concavidade da função quadrática. Se $a > 0$, a função possui concavidade voltada para cima, ou seja, o gráfico da função é uma parábola que abre para cima. Se $a < 0$, a função tem concavidade voltada para baixo, e o gráfico é uma parábola que abre para baixo.

Figura 46 – Concavidade voltada para cima



Fonte: Iezzi, 2013

Figura 47 – Concavidade voltada para baixo



Fonte: Iezzi, 2013

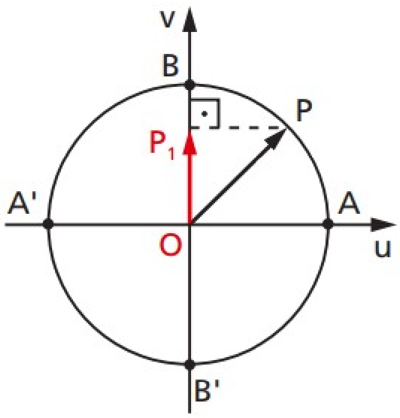
O coeficiente "b" afeta o deslocamento horizontal da parábola. Se $b > 0$, a parábola se desloca para a esquerda. Se $b < 0$, a parábola se desloca para a direita. O valor absoluto de b também influencia na inclinação da parábola.

O coeficiente "c" determina o deslocamento vertical da parábola. Se c > 0, a parábola é deslocada para cima em relação ao eixo y. Se c < 0, a parábola é deslocada para baixo.

6.4 Função Seno

Dado um número real x, seja P sua imagem no ciclo trigonométrico. Denominamos seno de x (e indicamos sen x) a ordenada $\overline{OP_1}$ do ponto P em relação ao sistema uOv. Denominamos função seno a função f: R → R que associa a cada real x o real OP_1 = sen x, isto é: f(x) = sen x.

Figura 48 – Seno de x



Fonte: Iezzi, 2013

6.4.1 Gráfico

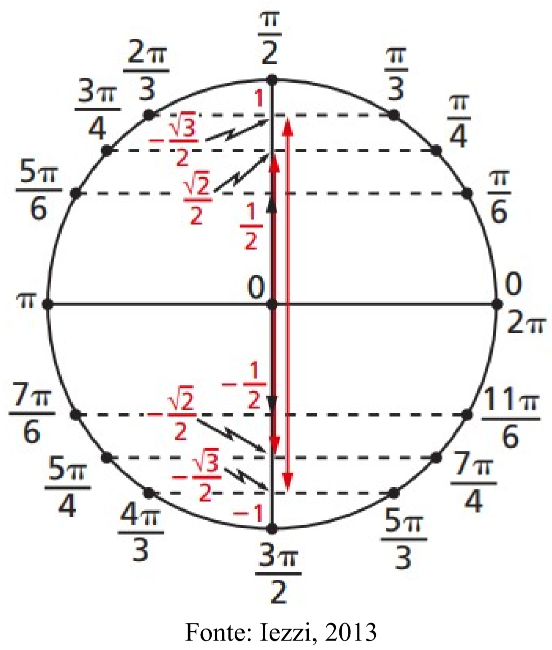
Fazendo uma tabela com x em abcissas e sen x em ordenadas, podemos construir o gráfico da função f(x) = sen x, denominado senoide, que nos indica como varia essa função.

Tabela 3 – Valores de x e y no intervalo de 0 a 2π

x	y = sen x
0	0
$\frac{\pi}{6}$	$\frac{1}{2}$
$\frac{\pi}{4}$	$\frac{\sqrt{2}}{2}$
$\frac{\pi}{3}$	$\frac{\sqrt{3}}{2}$
$\frac{\pi}{2}$	1
π	0
$\frac{3\pi}{2}$	-1
2π	0

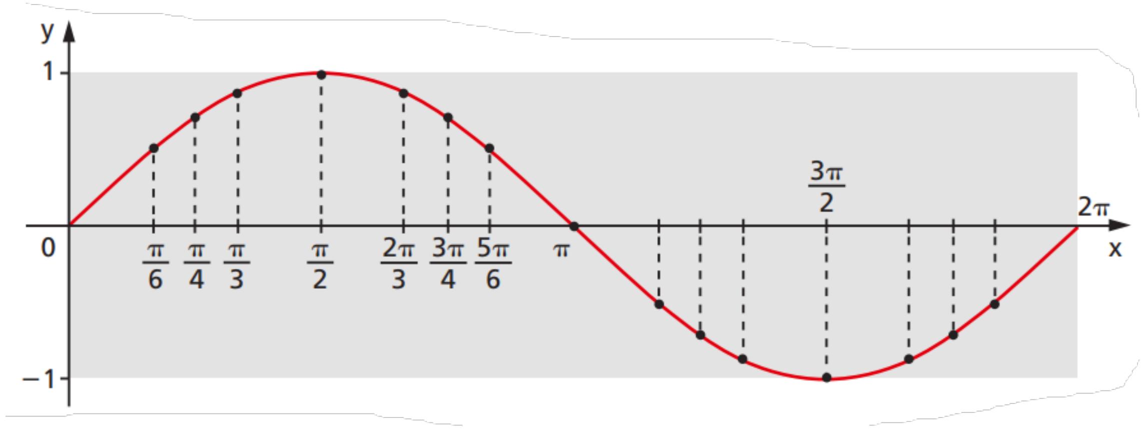
Fonte: elaborada pelo autor.

Figura 49 – Seno de x no ciclo



Fonte: Iezzi, 2013

Figura 50 – Gráfico de f(x) = sen x



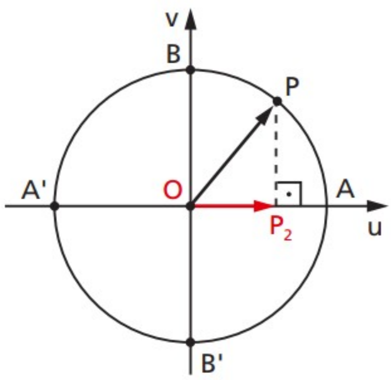
Fonte: Iezzi, 2013

Como o domínio da função é R, a senoide continua a direita de 2π e para a esquerda de 0. A função seno é uma função periódica, com período 2π, ou seja, a cada variação de 2π de x, o gráfico se repete.

6.5 Função Cosseno

Dado um número real x, seja P sua imagem no ciclo trigonométrico. Denominamos cosseno de x (e indicamos cos x) a abscissa $\overline{OP_2}$ do ponto P em relação ao sistema uOv. Denominamos função cosseno a função f: R → R que associa a cada real x o real OP_2 = cos x, isto é: f(x) = cos x.

Figura 51 – Cosseno de x



Fonte: Iezzi, 2013

6.5.1 Gráfico

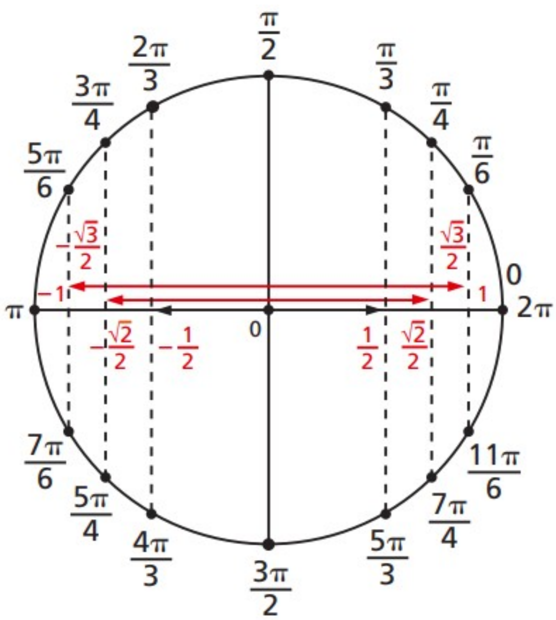
Fazendo uma tabela com x em abcissas e cos x em ordenadas, podemos construir o gráfico da função f(x) = cos x, denominado cossenoide, que nos indica como varia essa função.

Tabela 4 – Valores de x e y no intervalo de 0 a 2π

x	**y = cos x**
0	1
$\frac{\pi}{6}$	$\frac{\sqrt{3}}{2}$
$\frac{\pi}{4}$	$\frac{\sqrt{2}}{2}$
$\frac{\pi}{3}$	$\frac{1}{2}$
$\frac{\pi}{2}$	0
π	– 1
$\frac{3\pi}{2}$	0
2π	1

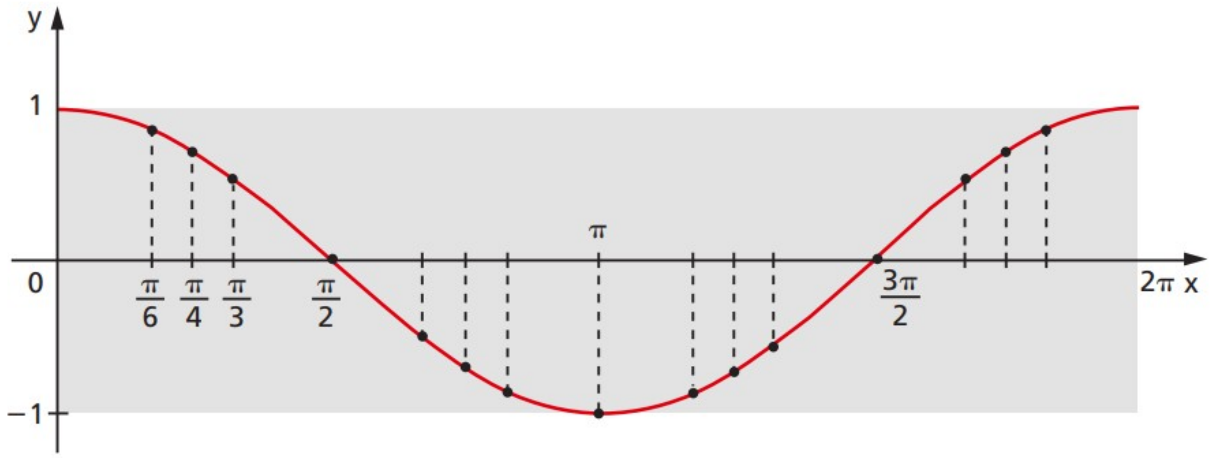
Fonte: elaborada pelo autor.

Figura 52 – Cosseno de x no ciclo



Fonte: Iezzi, 2013

Figura 53 – Gráfico de f(x) = sen x



Fonte: Iezzi, 2013

Como o domínio da função é R, a cossenoide continua a direita de 2π e para a esquerda de 0. A função cosseno é uma função periódica, com período 2π, ou seja, a cada variação de 2π de x, o gráfico se repete.

6.6 DESENVOLVIMENTO DO APLICATIVO

Na tela inicial do usuário, clicar em "+" para criar um aplicativo. Dar o nome de "Gráficos Matemáticos", selecionar a categoria "Educação" e clicar no botão "Criar". Na tela inicial do aplicativo, adicione outra tela. Em seguida, renomear as telas para "**Tela inicial**" e "**Gráficos**", respectivamente.

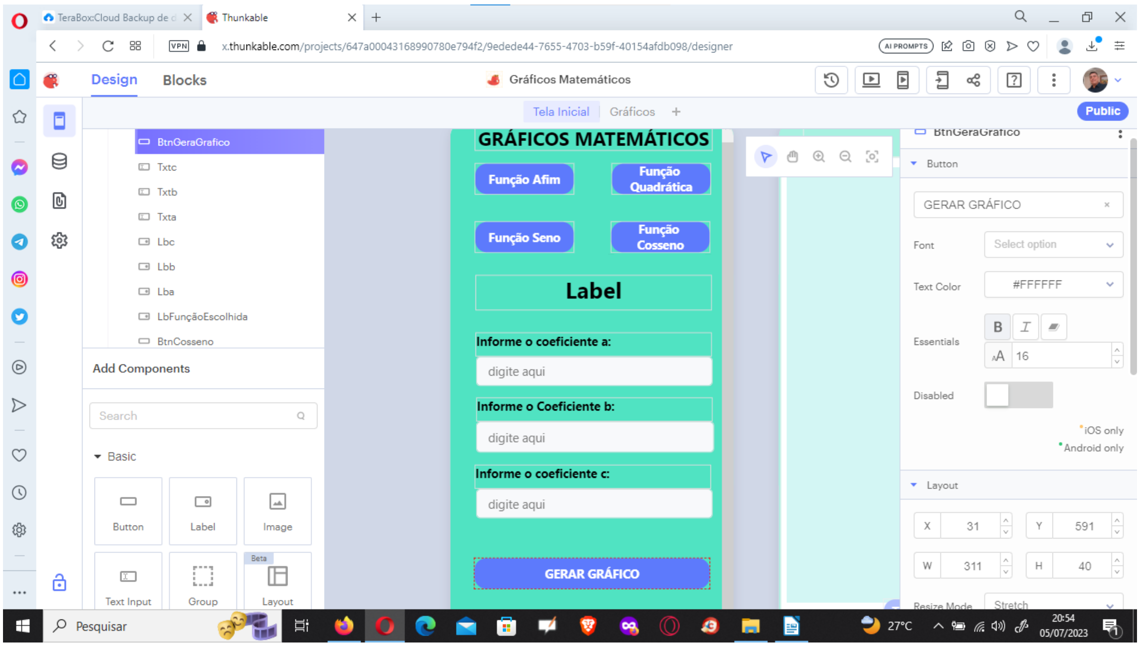
Na tela "**Tela inicial**", inserir um componente **Label**, renomeá-lo como **LbTitulo** e alterar a propriedade texto para "**GRÁFICOS MATEMÁTICOS**". Insira quatro componentes **Button**, renomeando-os, respectivamente, como **BtnAfim**, **BtnQuadratica**, **BtnSeno** e **BtnCosseno.** Em seguida, altere a propriedade texto desses botões para "**Função Afim**", "**Função Quadrática**", "**Função Seno**" e "**Função Cosseno**", respectivamente.

Insira outro componente **Label**, renomeando-o como **LbFunçãoEscolhida**. Insira mais três componentes **Label**, renomeando-os, respectivamente como **Lba**, **Lbb** e **Lbc**. Altere a propriedade **texto** desses componentes, respectivamente, para "**Informe o coeficiente a:**", "**Informe o coeficiente b:**" e "**Informe o coeficiente c:**".

Agora insira três componentes **Text Input**, renomeando-os como **Txta**, **Txtb** e **Txtc**. Altere a propriedade dica (hint) dos três componentes para "**digite aqui**".

Por fim, insira um componente **Buttom**, renomeando-o como **BtnGeraGrafico** e alterando a propriedade texto para "**GERAR GRÁFICO**". O resultado obtido após essas etapas deve ser igual ao da figura abaixo.

Figura 54 – Tela inicial do aplicativo Gráficos Matemáticos

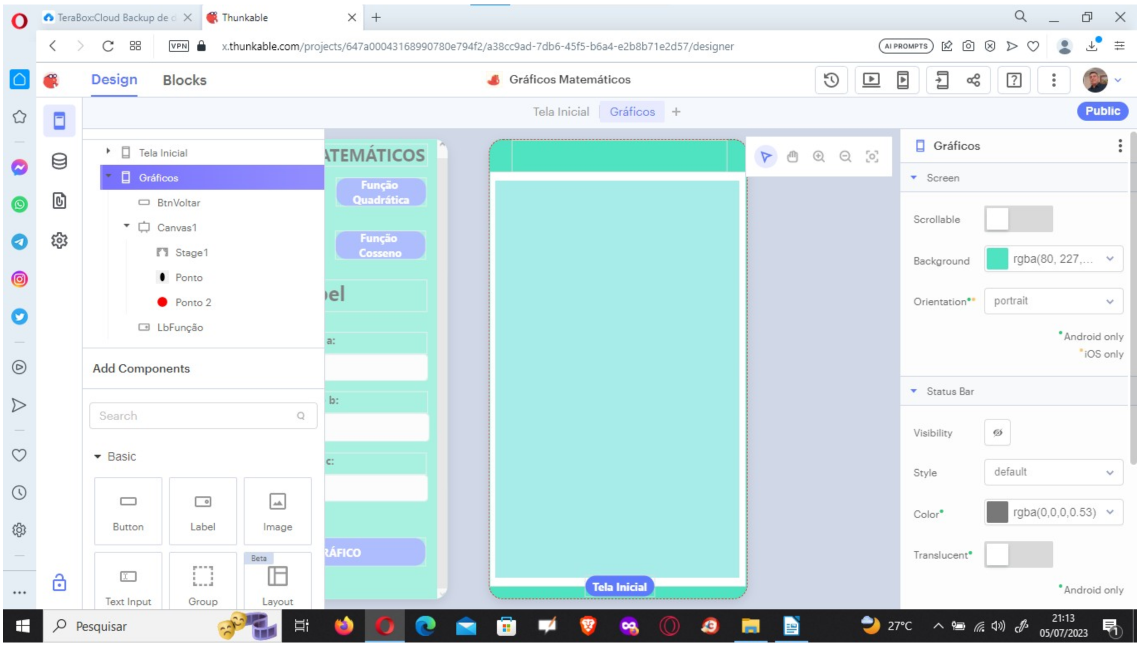


Fonte: elaborada pelo autor.

Na tela “**Gráficos**”, insira um componente **Canvas**. Altere as propriedades W(comprimento) e H(altura) para 370 e 600, respectivamente. Na área de design do Canva, Stage1, adicione (upload) uma imagem de um ponto de cor preta. Altere as propriedades Height(altura) e Width(comprimento) para 5 e 5, respectivamente. Repita esse processo para uma imagem de um ponto de cor vermelha.

Por fim, adicione um componente **Buttom**, renomeando-o como **BtnVoltar**. Em seguida, altere a propriedade texto para “**Tela Inicial**”. O resultado deve ser o semelhante ao da figura a baixo.

Figura 55 – Tela Gráficos do aplicativo Gráficos Matemáticos



Fonte: elaborada pelo autor.

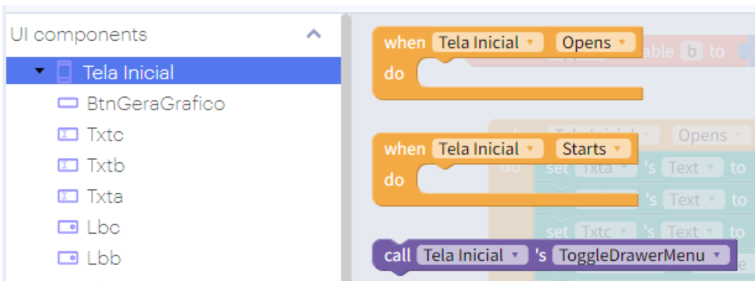
Ná área de blocos da “Tela Inicial”, crie quatro variáveis, nomeadas “a”, “b”, “c” e “GRAFICO”, respectivamente, todas com o valor 0.

Figura 56 – Inicializando variáveis na área de blocos da Tela Inicial

Fonte: elaborada pelo autor.

Clicando no componente Tela Inicial, selecione o bloco “when Tela Inicial Opens”.

Figura 57 – Selecionando o bloco when Tela Inicial Opens



Fonte: elaborada pelo autor.

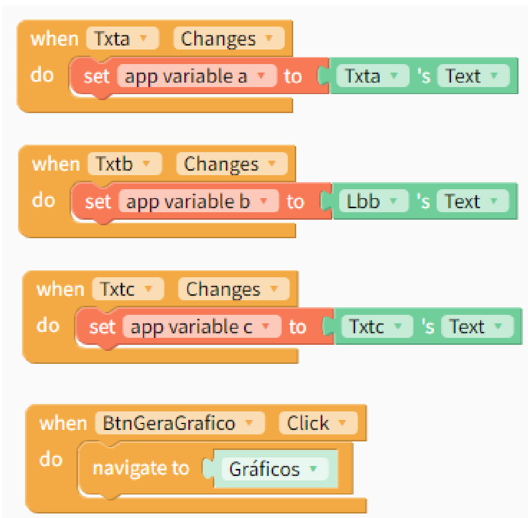
Preencha as propriedades dos componentes de acordo com a figura abaixo.

Figura 58 – Preenchendo as propriedades dos componentes no bloco when Tela Inicial Opens

Fonte: elaborada pelo autor.

Nos componentes Txta, Txtb e Txtc, escolha o bloco "changes" e preencha de acordo com a figura. Já no componente BtnGeraGrafico, escolha o bloco "click". Preencha de acordo com a figura.

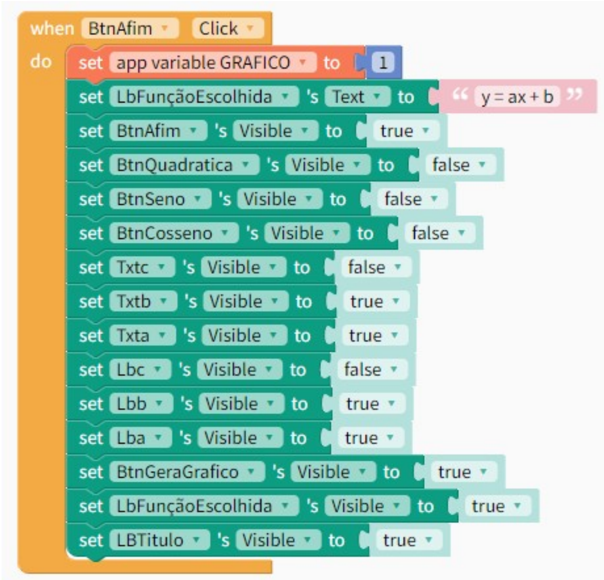
Figura 59 – Preenchendo as propriedades dos componentes no bloco when Tela Inicial Opens



Fonte: elaborada pelo autor.

No componente BtnAfim, selecione o bloco “when BtnAfim Click” e preencha de acordo com a imagem a seguir.

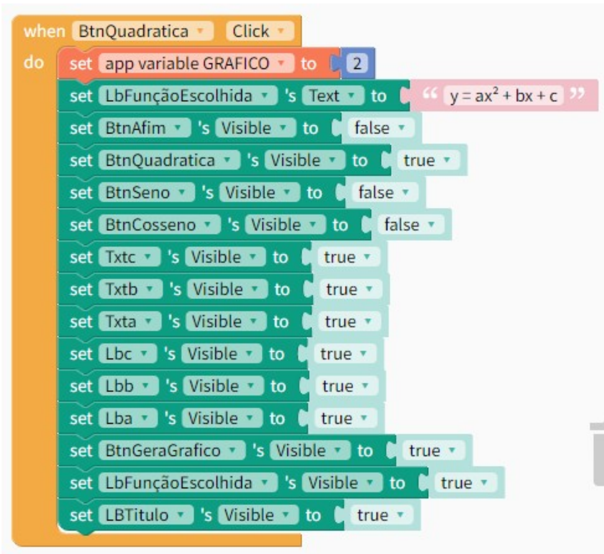
Figura 60 – Quando clicar no botão BtnAfim



Fonte: elaborada pelo autor.

No componente BtnQuadratica, selecione o bloco “when BtnQuadratica Click” e preencha de acordo com a imagem a seguir.

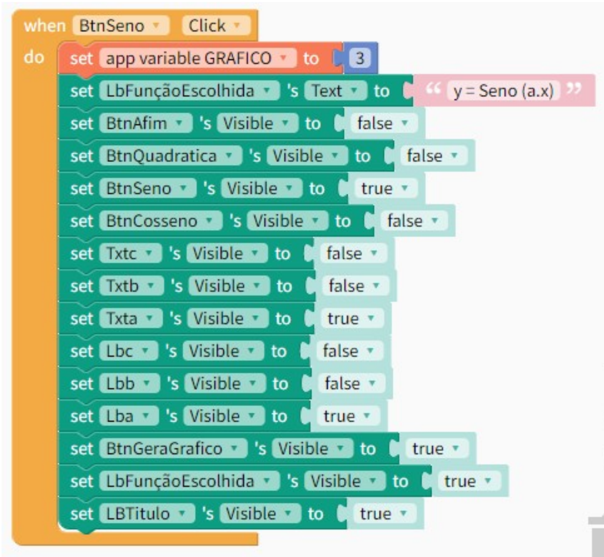
Figura 61 – Quando clicar no botão BtnQuadratica



Fonte: elaborada pelo autor.

No componente BtnSeno, selecione o bloco “when BtnSeno Click” e preencha de acordo com a imagem a seguir.

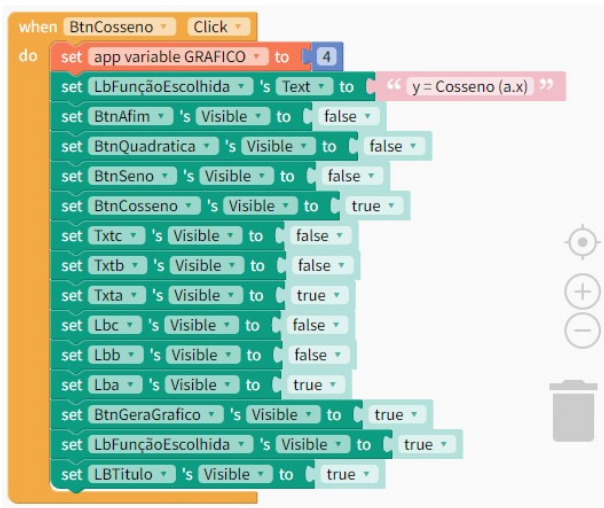
Figura 62 – Quando clicar no botão BtnSeno



Fonte: elaborada pelo autor.

No componente BtnSeno, selecione o bloco “when BtnSeno Click” e preencha de acordo com a imagem a seguir.

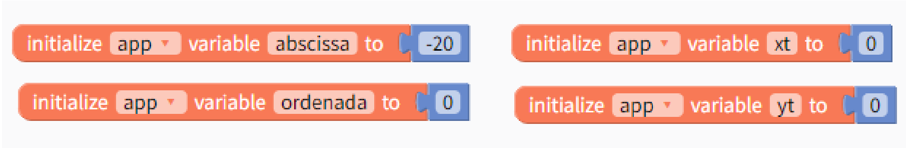
Figura 63 – Quando clicar no botão BtnCosseno



Fonte: elaborada pelo autor.

Ná área de blocos da “Gráficos”, crie quatro variáveis, nomeadas “abscissa”, “ordenada”, “xt” e “yt”, com os respectivos valores: - 20, 0, 0 e 0.

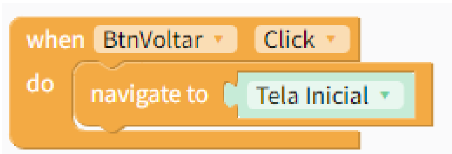
Figura 64 – Inicializando variáveis na área de blocos da Gráficos



Fonte: elaborada pelo autor.

No componente BtnVoltar, selecione o bloco “when BtnVoltar Click” e preencha de acordo com a imagem a seguir.

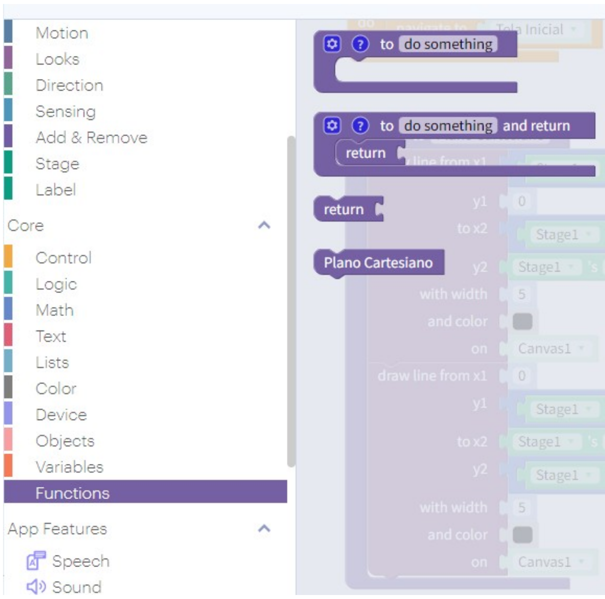
Figura 65 – Clicando no botão BtnVoltar



Fonte: elaborada pelo autor.

Na seção de Functions, selecione o bloco “to do something”. Renomeie a função para “Plano Cartesiano”.

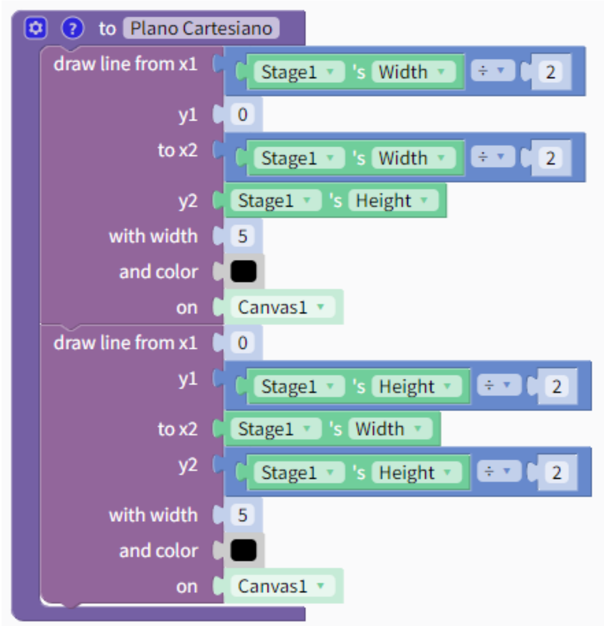
Figura 66 – Criando a função Plano Cartesiano



Fonte: elaborada pelo autor.

No componente Plano Cartesiano, preencha de acordo com a imagem a seguir.

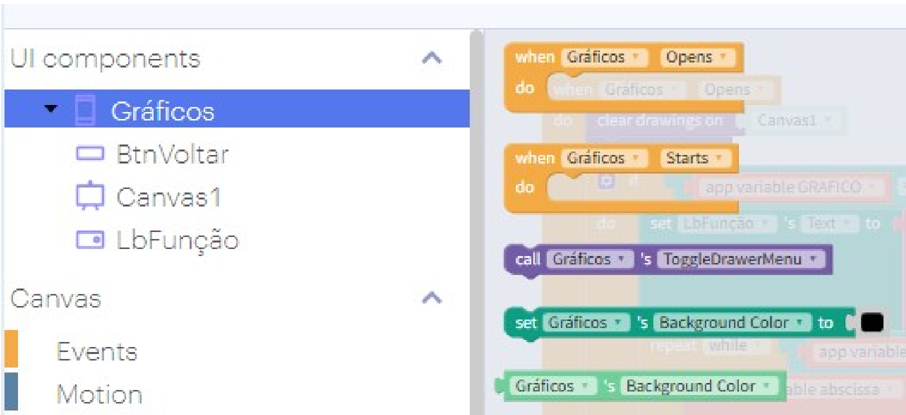
Figura 67 – Preenchendo a função Plano Cartesiano



Fonte: elaborada pelo autor.

Clicando no componente Gráficos, selecione o bloco "when Gráficos Opens".

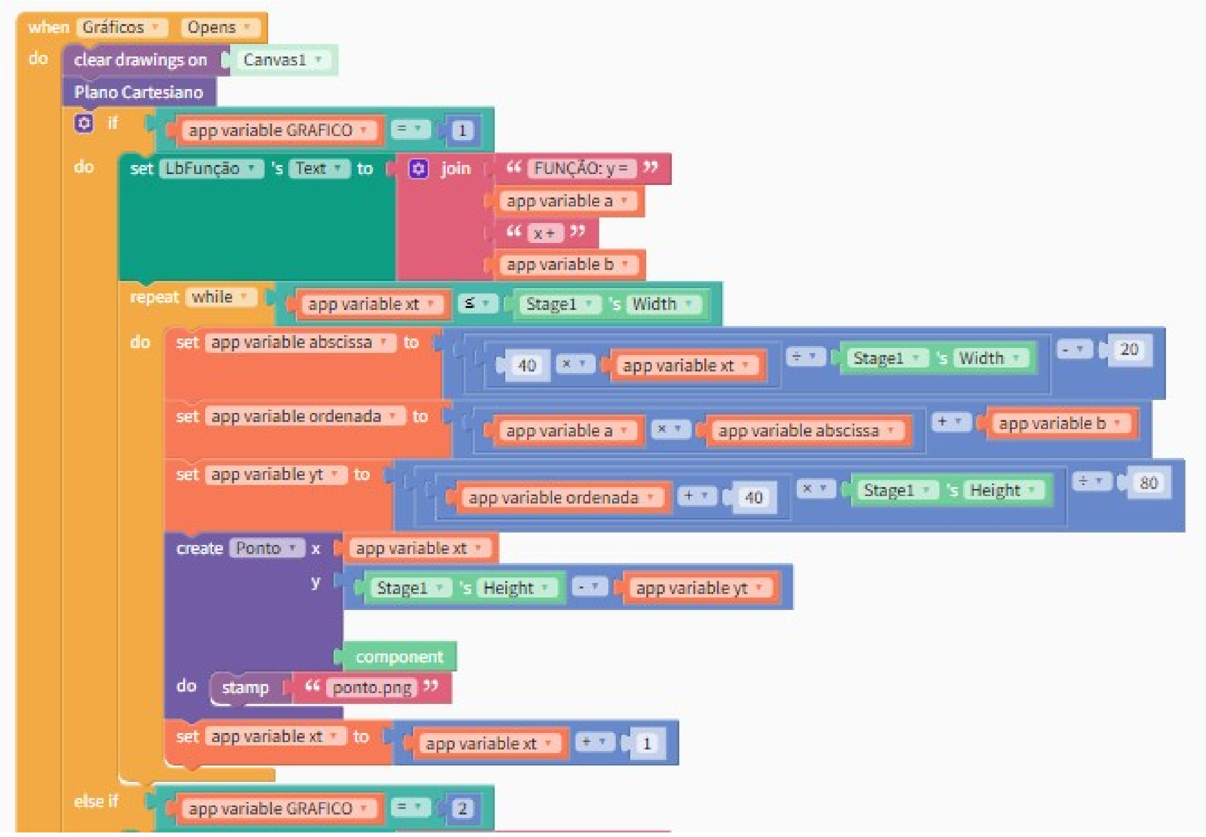
Figura 68 – selecionando o bloco "when Gráficos Opens"



Fonte: elaborada pelo autor.

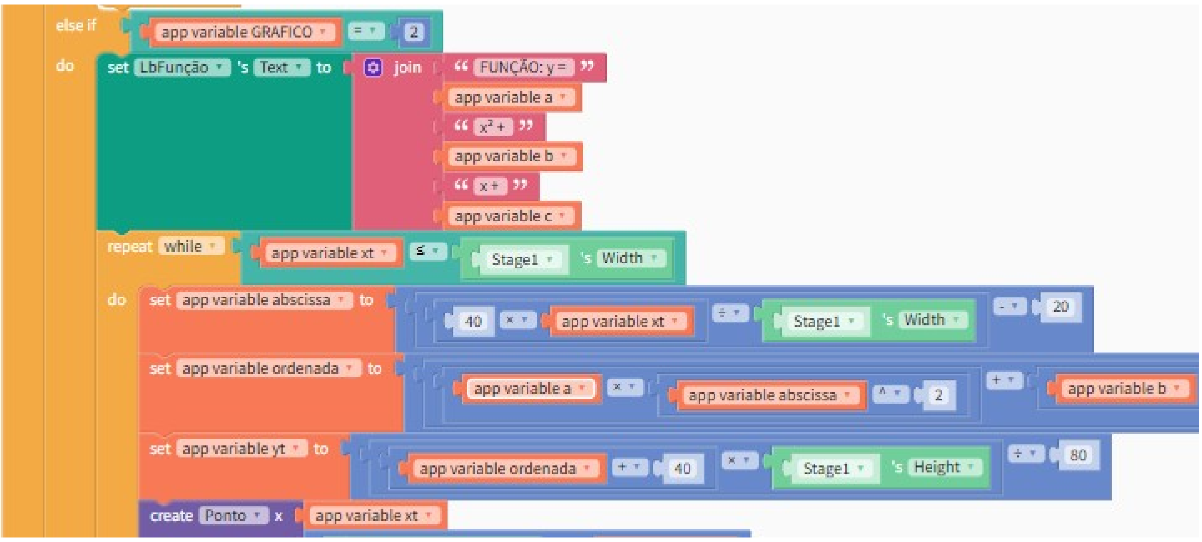
Preencha as propriedades dos componentes dentro do bloco "when Gráficos Opens" de acordo com a figura abaixo.

Figura 68 – Preenchendo as propriedades dos componentes no bloco when Gráficos Opens (parte 1)



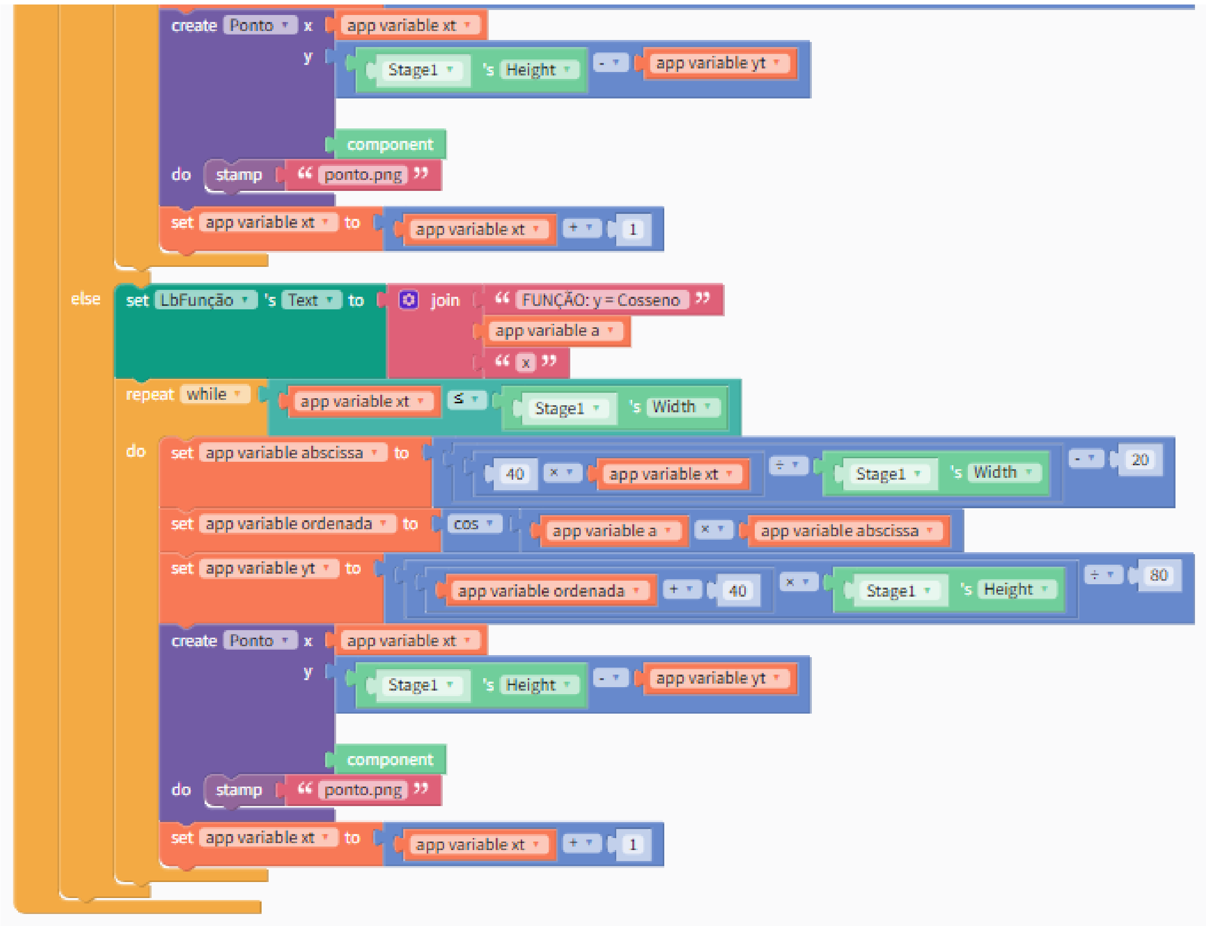
Fonte: elaborada pelo autor.

Figura 69 – Preenchendo as propriedades dos componentes no bloco when Gráficos Opens (parte 2)



Fonte: elaborada pelo autor.

Figura 70 – Preenchendo as propriedades dos componentes no bloco when Gráficos Opens (parte 3)



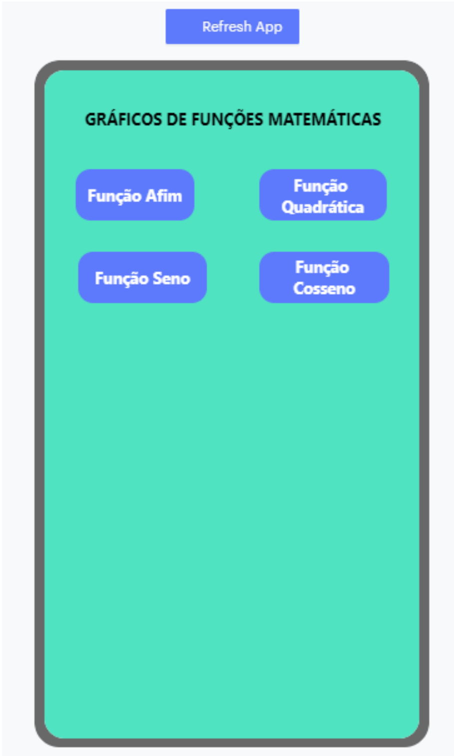
Fonte: elaborada pelo autor.

Finalizada a programação do aplicativo, devem ser feitos os testes para ver o seu funcionamento. Para isso, é só clicar no símbolo de **web preview**, que se encontra no menu superior do lado direito:

Figura 71 – Botão de web preview

Fonte: elaborada pelo autor.

Figura 72 – Tela Inicial do aplicativo Gráficos Matemáticos



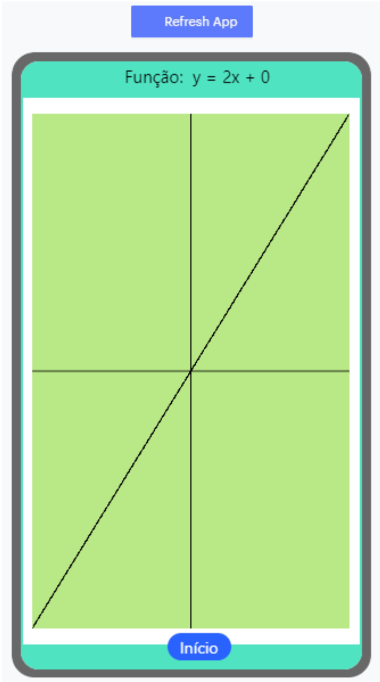
Fonte: elaborada pelo autor.

Como primeiro exemplo, vamos escolher a função afim. Em seguida pedir para o aplicativo gerar o gráfico da função linear y = 2x.

Figura 73 – Escolhendo a função afim e informando os coeficientes

Fonte: elaborada pelo autor.

Figura 74 – Gerando o gráfico da função y =2x



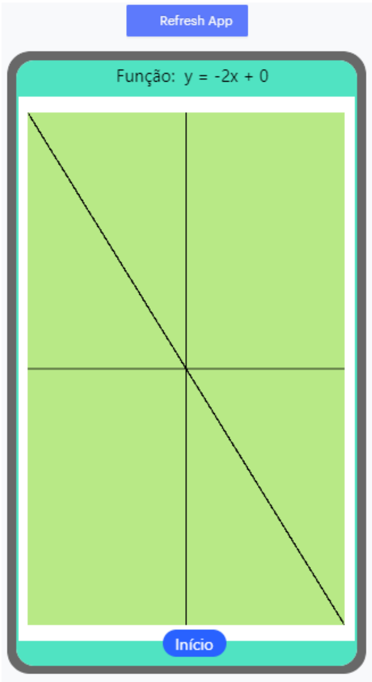
Fonte: elaborada pelo autor.

Clicando no botão “Início”, voltamos à tela inicial do aplicativo. Agora vamos gerar o gráfico da função linear y = – 2x.

Figura 75 – Escolhendo a função afim e informando os coeficientes

Fonte: elaborada pelo autor.

Figura 76 – Gerando o gráfico da função y = –2x



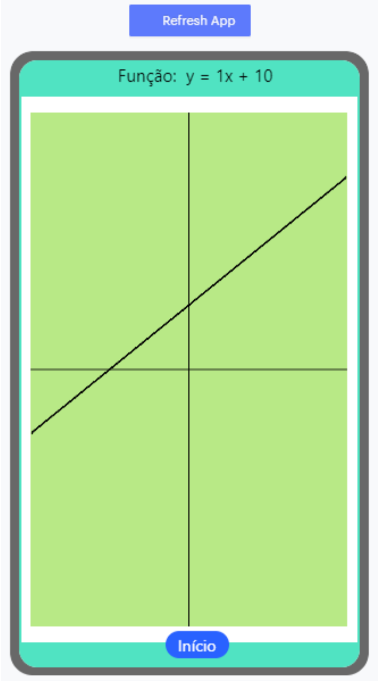
Fonte: elaborada pelo autor.

Clicando no botão "Início" e voltando à tela inicial, vamos escolher a função afim e pedir para gerar o gráfico da função y = x + 10.

Figura 77 – Escolhendo a função afim e informando os coeficientes

Fonte: elaborada pelo autor.

Figura 76 – Gerando o gráfico da função y = x + 10



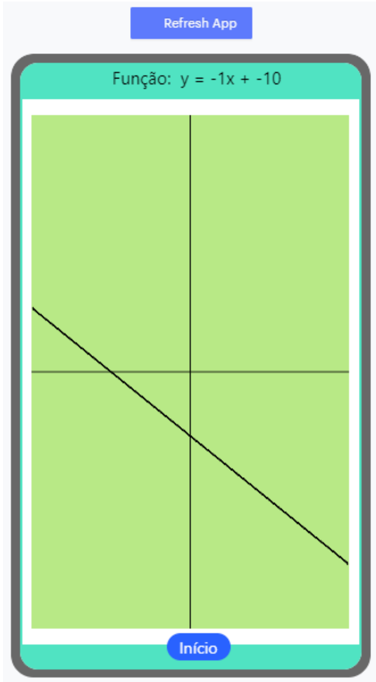
Fonte: elaborada pelo autor.

Para finalizar a função afim, vamos pedir para gerar o gráfico da função y = – x –10

Figura 77 – Escolhendo a função afim e informando os coeficientes

Fonte: elaborada pelo autor.

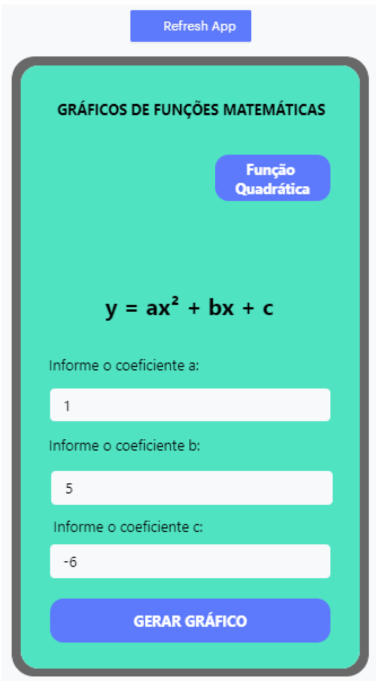
Figura 78 – Gerando o gráfico da função y = – x – 10



Fonte: elaborada pelo autor.

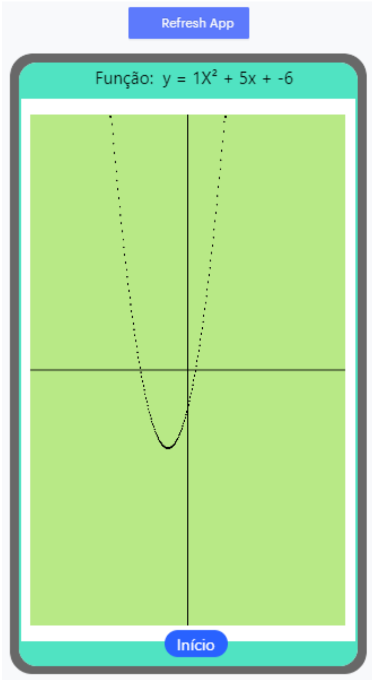
Vamos agora testar o gráfico da função quadrática. Escolhendo o botão "Função Quadrática" e informando os coeficientes da função $y = x^2 + 5x - 6$, teremos:

Figura 79 – Escolhendo a função quadrática e informando os coeficientes



Fonte: elaborada pelo autor.

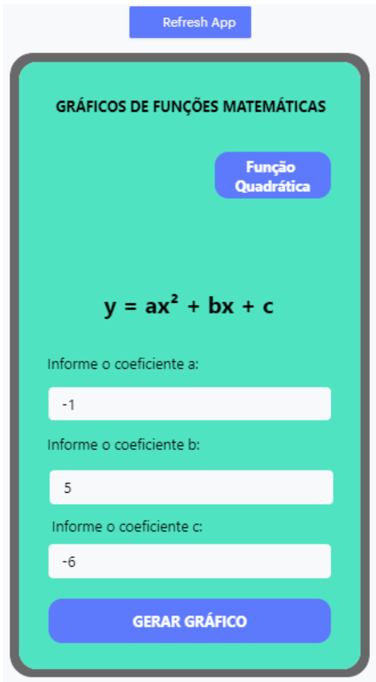
Figura 80 – Gerando o gráfico da função $y = x^2 + 5x - 6$



Fonte: elaborada pelo autor.

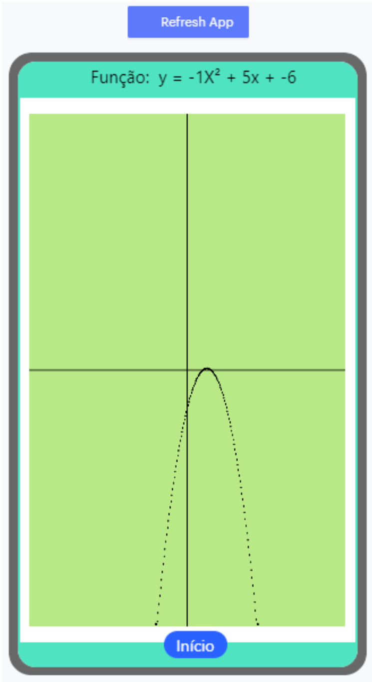
Vamos gerar o gráfico da função quadrática $y = -x^2 + 5x - 6$.

Figura 81 – Escolhendo a função quadrática e informando os coeficientes



Fonte: elaborada pelo autor.

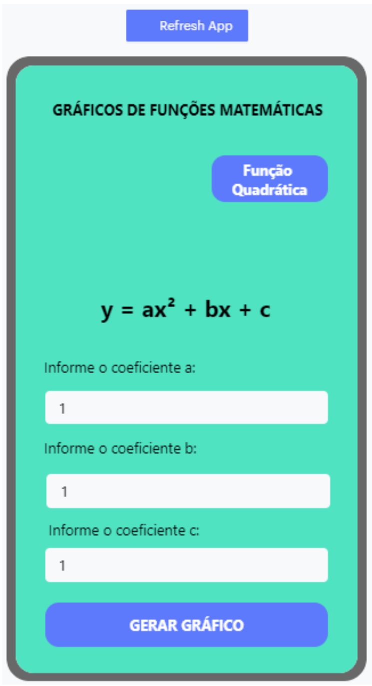
Figura 82 – Gerando o gráfico da função $y = -x^2 + 5x - 6$



Fonte: elaborada pelo autor.

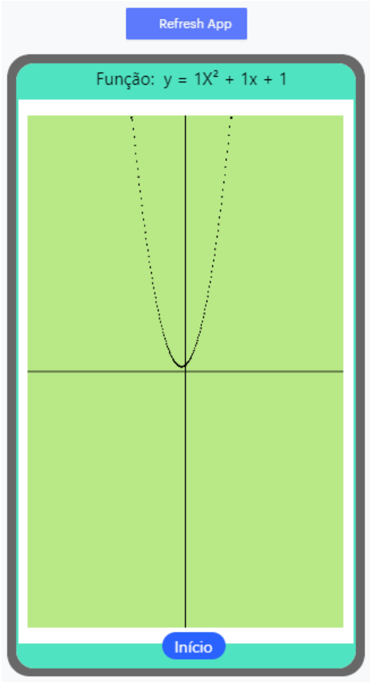
Agora vamos gerar o gráfico da função quadrática $y = x^2 + x + 1$.

Figura 83 – Escolhendo a função quadrática e informando os coeficientes



Fonte: elaborada pelo autor.

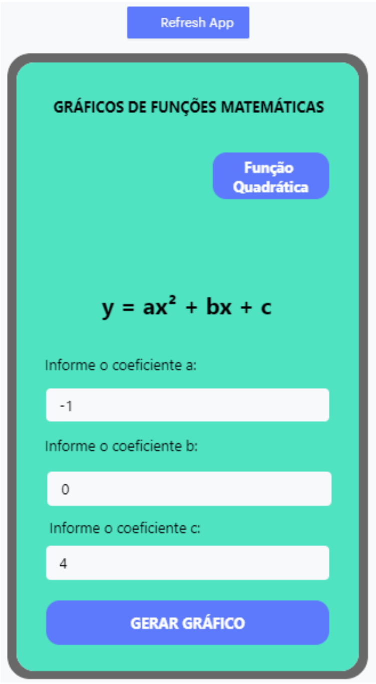
Figura 84 – Gerando o gráfico da função $y = x^2 + x + 1$.



Fonte: elaborada pelo autor.

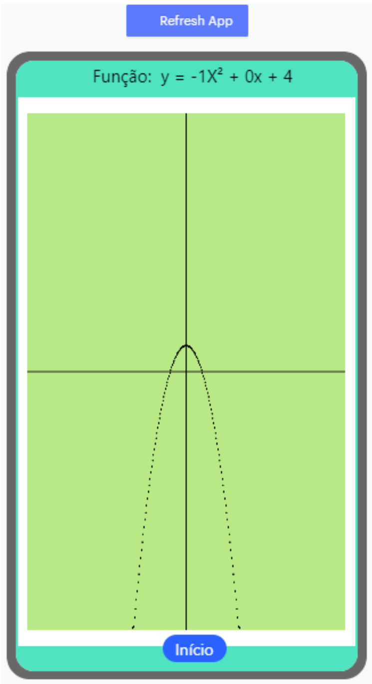
E ainda sobre a função quadrática, vamos gerar o gráfico da função $y = -x^2 + 4$.

Figura 85 – Escolhendo a função quadrática e informando os coeficientes



Fonte: elaborada pelo autor.

Figura 86 – Gerando o gráfico da função $y = -x^2 + 4$.



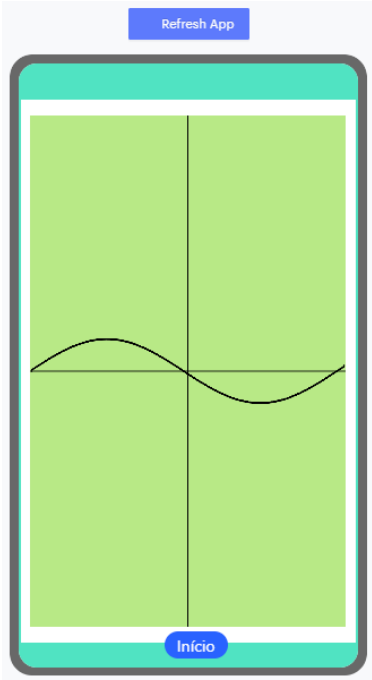
Fonte: elaborada pelo autor.

Vamos agora gerar o gráfico da função seno.

Figura 87 – Escolhendo a função seno e informando o coeficiente.

Fonte: elaborada pelo autor.

Figura 88 – Gerando o gráfico da função y = sen x.



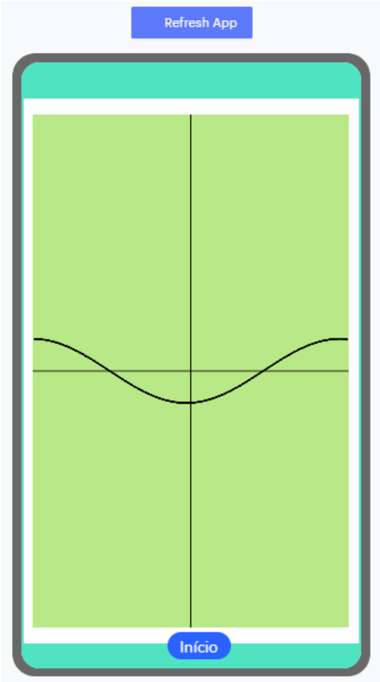
Fonte: elaborada pelo autor.

E para finalizar, vamos gerar o gráfico da função cosseno.

Figura 89 – Escolhendo a função cosseno e informando o coeficiente.

Fonte: elaborada pelo autor.

Figura 90 – Gerando o gráfico da função $y = \cos x$.



Fonte: elaborada pelo autor.

7 APLICATIVO 3: EQUAÇÃO DO 2º GRAU

Esse aplicativo lê os três coeficientes de uma equação do 2º grau e calcula suas raízes. Primeiramente ele calcula o valor do ”delta”, indicando se a equação tem duas raízes reais distintas, duas raízes reais idênticas ou não possui raízes reais.

7.1 Equação do 2º grau

Uma equação do 2º grau com uma incógnita é toda equação que pode ser reduzida à seguinte forma: $ax^2 + bx + c = 0$, com $a \neq 0$. Os números reais a, b e c são os coeficientes da equação do 2º grau.

Exemplos:

1. $5x^2 - 6x + 1 = 0$
2. $-0{,}4x^2 + 9x = 0$
3. $x^2 - 10 = 0$
4. $-5x^2 = 0$

Caso os coeficientes b e c são diferentes de zero a equação é considerada completa. Ela é considerada incompleta quando $b = 0$ ou $c = 0$, ou ainda, quando $b = 0$ e $c = 0$.

7.1.1 raízes da equação do 2º grau

Resolver uma equação do 2º grau é encontrar o(s) valor(es) de x que tornam a igualdade verdadeira. No caso, chamamos de encontrar os zeros ou raízes da equação, o que corresponde a encontrar os zeros ou raízes da função quadrática associada à equação. Para encontrar essas raízes, um dos procedimentos é utilizar a fórmula resolutiva, dada a seguir:

Figura 91 – Fórmula resolutiva da equação do 2º grau

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

Fonte: elaborada pelo autor.

Exemplos:

1. Encontre as raízes da equação $x^2 + 5x + 6 = 0$.

Aplicando a fórmula resolutiva para os coeficientes a = 1, b = 5 e c = 6, obtemos como raízes os valores: $x_1 = -3$ e $x_2 = -2$.

2. Encontre as raízes da equação $x^2 - 4x = 0$.

Analogamente ao exemplo anterior, aplicando a fórmula resolutiva para os coeficientes a = 1, b = – 4 e c = 0, obtemos como raízes os valores: $x_1 = 0$ e $x_2 = 4$.

3. Encontre as raízes da equação $x^2 - 6x + 9 = 0$.

Nesse exemplo, aplicando a fórmula resolutiva para os coeficientes a = 1, b = – 6 e c = 9, obtemos como raízes os valores: $x_1 = x_2 = -3$.

4. Encontre as raízes da equação $x^2 + x + 1 = 0$.

Aplicando a fórmula resolutiva para os coeficientes a = 1, b = 1 e c = 1, nós observamos que não há raízes reais para essa equação.

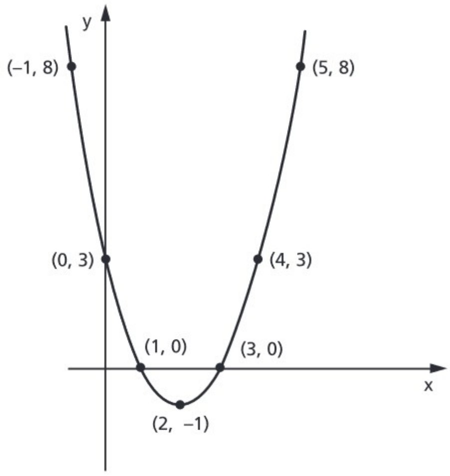
7.1.2 Significado geométrico das raízes da equação do 2º grau

Interpretando geometricamente as raízes da equação do 2º grau, elas são os valores das abscissas dos pontos onde a parábola corta o eixo dos x.

Exemplo:

1. resolvendo a equação do 2º grau $x^2 - 4x + 3 = 0$ e construindo o gráfico da função equivalente $y = x^2 - 4x + 3$, obtemos as raízes 1 e 3.

Figura 92 – Gráfico da função quadrática $y = x^2 - 4x + 3$



Fonte: Iezzi, 2013

Os pontos (1,0) e (3,0) indicam onde a parábola corta o eixo dos x, sendo suas raízes 1 e 3.

7.2 DESENVOLVIMENTO DO APLICATIVO

Na tela inicial do usuário, clicar em "+" para criar um aplicativo. Dar o nome de "Baskara", selecionar a categoria "Educação" e clicar no botão "Criar". Na tela inicial do aplicativo, adicione outra tela.

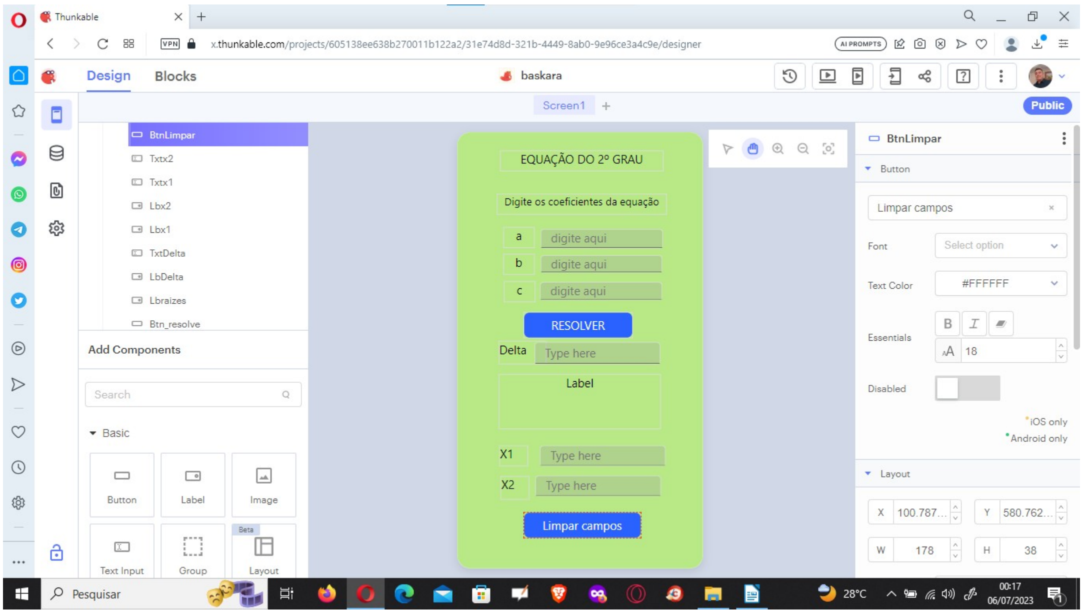
Na tela "Screen1" insira um componente **Label**, renomeando-o como **LbTítulo**. Altere a propriedade texto para "**EQUAÇÃO DO 2º GRAU**". Insira outro componente **Label**, renomeando-o como **LbDigite**. Insira mais três componentes **Label**, renomeando-os, respectivamente, como **Lba**, **Lbb** e **Lbc**. Altere a propriedade texto desses componentes para "**a**", "**b**" e "**c**", respectivamente.

Insira três componentes **Text Input**, renomeando-os para **Txta**, **Txtb** e **Txtc**, respectivamente. Altere a propriedade hint (dica) desses três componentes para "**digite aqui**". Insira um componente **Label**, renomeando-o para **LbDelta**. Altere a propriedade tesxto para "**Delta**". Insira um componente **Text Input**, renomeando-o para **TxtDelta**.

Insira um componente **Label**, renomeando-o para **Lbraizes**. Insira dois componentes **Label**, renomeando-os como **Lbx1** e **Lbx2**. Altere a propriedade texto desses componentes para "**X1**" e "**X2**", respectivamente. Insira dois componentes **Text Input**, renomeando-os como **Txtx1** e **Txtx2**, respectivamente.

Por fim, insira um componente **Buttom**, renomeando-o como **BtnLimpar**. Altere a propriedade texto para "**Limpar campos**". Após essas etapas, você deve obter algo semelhante à figura a seguir.

Figura 93 – tela inicial do aplicativo Baskara



Fonte: elaborada pelo autor.

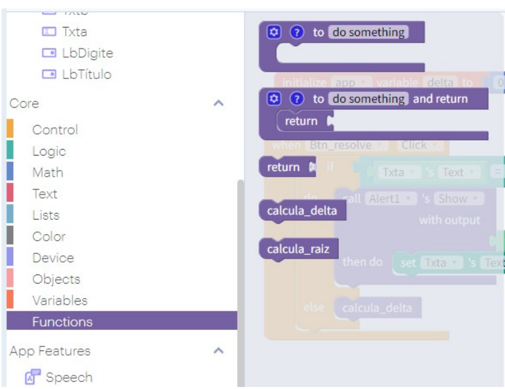
Ná área de blocos da “Screen1”, crie a variável nomeadas “delta”, com o valor 0.

Figura 94 – tela inicial do aplicativo Baskara

Fonte: elaborada pelo autor.

Na seção de **Functions**, selecione o bloco “**to do something**”. Renomeie a função para “**calcula_delta**”.

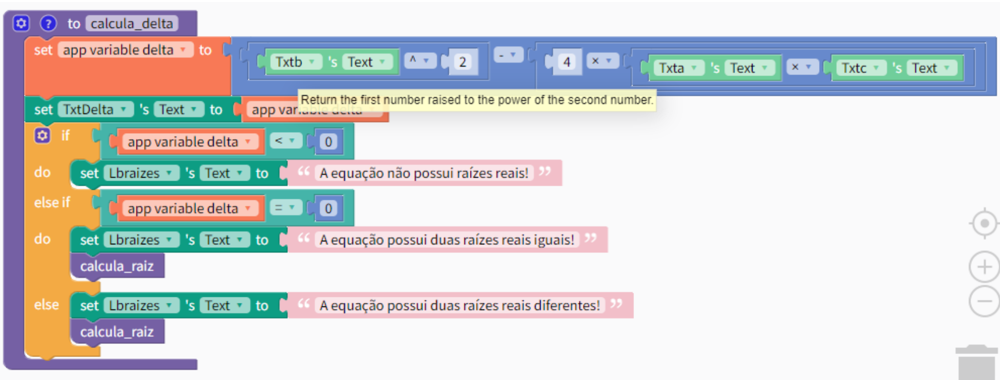
Figura 95 – Criando a função calcula_delta



Fonte: elaborada pelo autor.

No componente **calcula_delta**, preencha de acordo com a imagem a seguir.

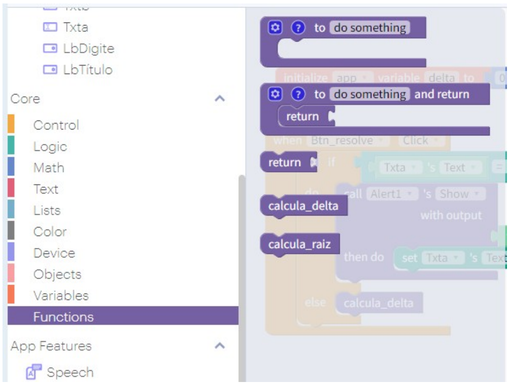
Figura 96 – Preenchendo a função calcula_delta



Fonte: elaborada pelo autor.

Na seção de **Functions**, selecione o bloco “**to do something**”. Renomeie a função para “**calcula_raiz**”.

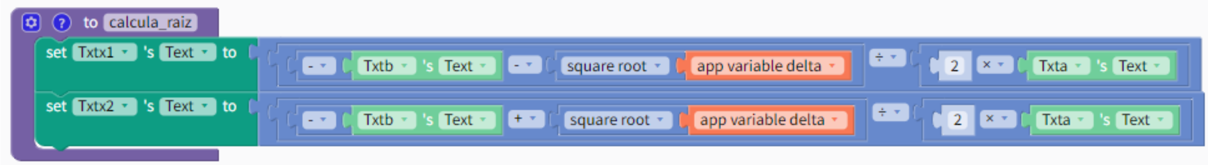
Figura 97 – Criando a função calcula_raiz



Fonte: elaborada pelo autor.

No componente **calcula_raiz**, preencha de acordo com a imagem a seguir.

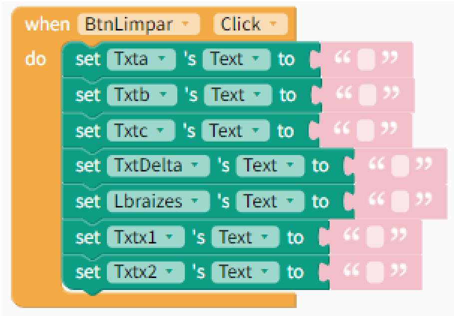
Figura 98 – Preenchendo a função calcula_raiz



Fonte: elaborada pelo autor.

No componente **BtnLimpar**, selecionar o bloco “when BtnLimapr Click” e preencher de acordo coma figura abaixo.

Figura 99 – Preenchendo o bloco BtnLimpar Click



Fonte: elaborada pelo autor.

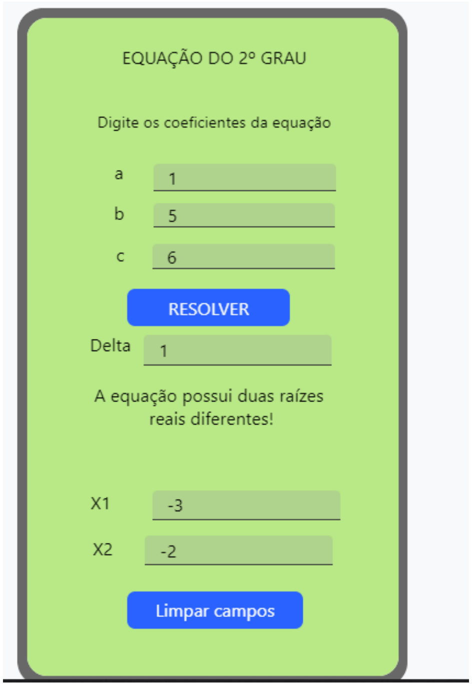
Finalizada a programação do aplicativo, devem ser feitos os testes para ver o seu funcionamento. Para isso, é só clicar no símbolo de **web preview**, que se encontra no menu superior do lado direito:

Figura 100 – Botão de web preview

Fonte: elaborada pelo autor.

Vamos verificar a solução da equação do 2º grau $x^2 + 5x + 6 = 0$.

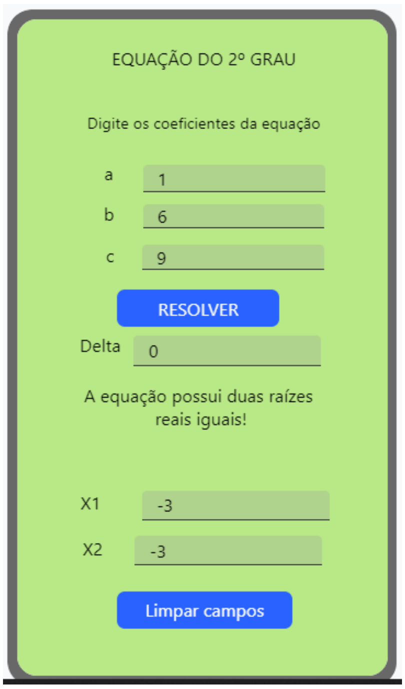
Figura 101 – Solução da equação do 2º grau $x^2 + 5x + 6 = 0$



Fonte: elaborada pelo autor.

Agora vamos verificar a solução da equação do 2º grau $x^2 + 6x + 9 = 0$.

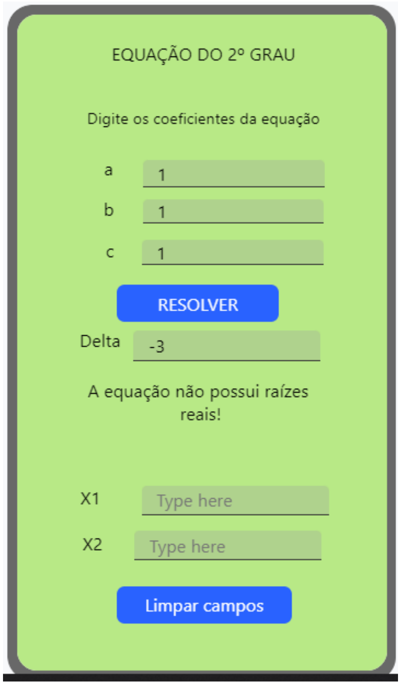
Figura 102 – Solução da equação do 2º grau $x^2 + 6x + 9 = 0$



Fonte: elaborada pelo autor.

E para finalizar, vamos verificar a solução da equação $x^2 + x + 1 = 0$.

Figura 103 – Solução da equação do 2º grau $x^2 + x + 1 = 0$



Fonte: elaborada pelo autor.

8 SUGESTÕES DE OUTROS APLICATIVOS A SEREM DESENVOLVIDOS

No presente trabalho apresentamos algumas ideias de aplicativos a serem desenvolvidos com foco no conteúdo de Matemática da Educação Básica. Tendo em vista os recursos disponíveis na plataforma thunkable e em outras similares, como o MIT APP Inventor e o Kodular, é possível a criação de aplicativos que abordem todo o conteúdo de matemática, não só do Ensino Fundamental como do Ensino Médio. Na verdade, muitos conteúdos do Ensino Superior são passíveis de serem abordados por meio de aplicativos criados nessas plataformas. A seguir, sugerimos alguns assuntos que podem ser abordados de imediato. Uma espécie de guia ou roteiro a ser trabalhado nas aulas de Matemática. Para facilitar, dividiremos por área.

ARITMÉTICA

- P.A. e P.G.: aplicativo que determina qualquer uma das variáveis da fórmula do termo geral. Aplicativo que calcule a soma de n termos de uma P.A. ou P.G. finita. Aplicativo que calcule a soma dos infinitos termos de uma P.G. infinita.
- Combinatória: aplicativo que calcule o fatorial de um número inteiro n. Aplicativo que calcule o número de arranjos de n elementos agrupados p a p. Aplicativo que calcule o

número de combinações de n elementos agrupados p a p.

- Probabilidade: aplicativo que calcule a probabilidade de ocorrer um evento com n situações favoráveis dentre p possíveis.

GEOMETRIA

- Área: aplicativo que calcula a área de um triângulo pela fórmula de Herão. Aplicativo que calcula a área de um triângulo de acordo com os valores da base e da altura. Aplicativo que calcula a área de um losango, de acordo com os valores das diagonais. Aplicativo que calcula a área de um quadrado de acordo com a medida do lado. Aplicativo que calcula a área de um paralelogramo de acordo com as medidas da base e da altura. Aplicativo que calcula a área de um trapézio de acordo com as medidas das bases e da altura. Aplicativo que calcula a área de um losango de acordo com as medidas das diagonais. Aplicativo que calcula a área de um polígono de acordo com as coordenadas dos vértices do polígono no Plano Cartesiano.
- Volume: Aplicativo que calcula o volume do sólido de acordo com as suas medidas.

ÁLGEBRA

- Aplicativo que encontra a solução de uma equação do 1º grau. Aplicativo que encontra a solução de um sistema de equações lineares. Aplicativo que gera o gráfico de uma função real qualquer, dada a sua fórmula.

Como podemos ver, as possibilidades são inúmeras da criação de aplicativos com foco na Matemática. É importante salientar, no entanto, que o domínio dos recursos dessas plataformas permite que sejam desenvolvidas soluções para as mais diversas áreas de conhecimento. Isso faz com que essas ferramentas sejam úteis não só como ferramentas educacionais, mas como ferramentas profissionais.

9 CONCLUSÃO

Ao finalizar este trabalho, podemos concluir que a plataforma Thunkable é uma ferramenta muito útil para o ensino de matemática, uma vez que permite a criação de aplicativos móveis de forma fácil e acessível. Isso pode ajudar a tornar o aprendizado da matemática mais interessante e envolvente para os estudantes. Eles podem aplicar os conceitos matemáticos aprendidos em sala de aula em situações práticas do dia–a–dia, o que pode ajudá–los a compreender melhor a importância da matemática em suas vidas. Além disso, é possível desenvolver habilidades de pensamento crítico, resolução de problemas e trabalho em equipe, uma vez que os estudantes trabalham juntos para criar um produto final.

O Thunkable é uma ferramenta que pode ser usada tanto em sala de aula quanto em casa, permitindo que os estudantes trabalhem em seus próprios ritmos e desenvolvam habilidades de programação de forma autônoma. A plataforma também é bastante intuitiva e fácil de usar, permitindo que mesmo os estudantes mais jovens possam criar aplicativos com facilidade.

Por fim, ela oferece uma maneira inovadora e envolvente de ensinar matemática, tornando o aprendizado mais interessante e prático para os estudantes. A criação de aplicativos pode ajudar a aumentar o interesse e a motivação dos estudantes pela matemática, o que pode levar a melhores resultados acadêmicos e a uma compreensão mais profunda da disciplina.

REFERÊNCIAS

BARBOSA, Marcos Alberto. **Desenvolvendo aplicativos para dispositivos móveis através do MIT App inventor 2 nas aulas de matemática.** Ilhéus, Bahia: UESC 2016.

BIANCHINI, Edwaldo. **Matemática - Bianchini** : manual do professor / Edwaldo Bianchini. 9. ed. São Paulo : Moderna, 2018.

GAMA, Bruno Nogueira Gonçalves. **Aplicativos Educacionais como Ferramentas no Ensino da Matemática [manuscrito]** / Bruno Nogueira Gonçalves Gama. 2021.

IEZZI, Gelson. **Fundamentos de matemática elementar**, 1: conjuntos, funções / Gelson Iezzi, Carlos Murakami. 9. ed. São Paulo : Atual, 2013.

IEZZI, Gelson. **Fundamentos de matemática elementar**, 3 : trigonometria : 506 exercícios propostos com resposta, 167 testes de vestibulares com resposta / Gelson Iezzi. 9. ed. São Paulo : Atual, 2013.

OLIVEIRA, José Marcelo Velloso de. **Criação de aplicativo para dispositivos móveis e sua utilização como recurso didático em aulas de geometria analítica** / José Marcelo Velloso de Oliveira. 2016.

SANTOS, José Plínio de Oliveira. **Introdução à Teoria dos Números**. Associação Instituto Nacional de Matemática Pura e Aplicada, Rio de Janeiro, 2003.

APÊNDICE A – LINKS DOS APLICATIVOS DESENVOLVIDOS

1. Números primos

https://x.thunkable.com/copy/31e91c4c404ab5e227b0dbd88ad03c91

2. Gráficos Matemáticos

https://x.thunkable.com/copy/64b1ba2decddd8f13ea97a12141c4c01

3. Equação do 2º grau

https://x.thunkable.com/copy/cb97ccf6dfbb08128076b05fe7a3399f