2022

PENSAMENTO COMPUTACIONAL:

desenvolvimento de aplicativos sobre Análise Combinatória no App Inventor 2

Lucas Valois Leal de Oliveira

Cinthia Cunha Maradei Pereira

Fábio José da Costa Alves

OLIVEIRA, Lucas Valois Leal de; PEREIRA, Cinthia Cunha Maradei; ALVES, Fábio José da Costa. PENSAMENTO COMPUTACIONAL: desenvolvimento de aplicativos sobre Análise Combinatória no App Inventor 2. Produto Educacional do Programa de Pós-Graduação em Ensino de Matemática (PPGEM), Curso de Mestrado Profissional em Ensino de Matemática da Universidade do Estado do Pará (UEPA), 2022.

ISBN: 978-65-84998-19-3

Ensino de Matemática; Pensamento Computacional; App Inventor 2; Análise Combinatória.

APRESENTAÇÃO

O trabalho a seguir foi contemplado em conjunto ao Programa de Pós-Graduação em Ensino de Matemática (PPGEM) da Universidade do Estado do Pará (UEPA) e foi desenvolvido durante a disciplina de Tecnologias de Informática Aplicadas ao Ensino de Matemática, que está em vigor no curso de Mestrado Profissional em Ensino de Matemática.

Este livro tem como objetivo trabalhar o pensamento computacional dos estudantes por meio da criação de aplicativos de calculadoras acerca do conteúdo matemático de Análise Combinatória no programa App Inventor 2, além de buscar as potencialidades presentes nesse processo de ensino.

O desenvolvimento desse estudo se deu em virtude de o pensamento computacional contribuir para a ampliação do pensamento matemático, bem como da autonomia dos alunos em sala de aula.

A partir disso, esse trabalho propõe a construção de três calculadoras relativas a alguns tópicos trabalhados no conteúdo Análise Combinatória: Arranjo simples, Combinação simples e Combinação completa.

Em primeiro lugar, será evidenciado o conceito matemático formal dos tópicos que são trabalhados na calculadora e, em seguida, será apresentado algumas situações problemas que deverão ser resolvidas com a construção das calculadoras no App Inventor 2.

Em segundo lugar, será revelado o designer de cada uma das calculadoras (que deverá ser construído em sala pelos discentes, a partir da orientação do docente responsável) com alguns conselhos para utilizar as ferramentas do aplicativo e, em seguida, desenvolver a programação dessas calculadoras, a fim de conciliar o objeto matemático em questão com o pensamento computacional, por meio de diferentes representações: a linguagem materna, a linguagem matemática e a linguagem computacional.

Dessa forma, o material aqui oferecido pode ser utilizado por professores de matemática que desejem trabalhar o pensamento computacional com seus alunos em sala de aula e, vale ressaltar, que esse trabalho desse ser feito sempre com o auxílio do docente em cada etapa, mas também deve proporcionar aos discentes a liberdade e a autonomia durante as etapas de construção.

1. INTRODUÇÃO

Este trabalho foi pensado durante a disciplina de *Tecnologias de Informática no Ensino de Matemática*, ministrada no Programa de Pós-Graduação em Ensino de Matemática (PPGEM) da Universidade do Estado do Pará (UEPA) e busca trabalhar alguns tópicos de Análise Combinatória com auxílio dos pressupostos do pensamento computacional.

De acordo com Barcelos e Silveira (2012), o pensamento computacional define competências e habilidades que se tornam fundamentais para o efetivo domínio da tecnologia por todos em um mundo em que os dispositivos computacionais são cada vez mais pervasivos. Em virtude disso, trabalhar esse pensamento com as crianças e jovens, já em sala de aula, pode vir a ser vantajoso para o desenvolvimento intelectual desses indivíduos.

Além disso, Barcelos e Silveira (2012) afirmam que incorporar o pensamento computacional à educação básica envolve a análise sistemática de sua potencial sinergia com outras áreas do conhecimento, como a Matemática. Com isso, trabalhar os conteúdos matemáticos com o auxílio de instrumentos que estimulem o pensamento computacional pode se mostrar como uma metodologia de ensino efetiva para ser utilizada com os estudantes da educação básica.

Tal pensamento corrobora com as ideias de Azevedo e Maltempi (2020), que discutem sobre o papel fundamental que o pensamento computacional pode ter durante o processo de aprendizagem da matemática, uma vez que este incentiva os alunos a desenvolver novos modos de pensar e elaborar caminhos para a mobilização de conhecimentos, bem como estimula a autonomia e a criatividade daqueles envolvidos no desenvolvimento dos saberes matemáticos.

Observa-se ainda textos oficiais que destacam a importância da implementação do pensamento computacional durante o ensino de matemática na educação básica. Um desses documentos é a Base Nacional Comum Curricular (BNCC), que apresenta a seguinte competência especifica, direcionada para a área de Matemática e suas Tecnologias:

> Compreender e utilizar, com flexibilidade e fluidez, diferentes registros de representação matemáticos (algébrico, geométrico, estatístico, computacional etc.), na busca de solução e comunicação de resultados de problemas, de modo a favorecer a construção e o desenvolvimento do raciocínio matemático (BRASIL, 2018, p. 530).

Nessa competência pode-se observar o direcionamento para a utilização de registros variados de representação matemática e, dentre eles, é possível destacar o registro computacional que, para ser feito de forma proveitosa aos alunos, deve-se trabalhar em conjunto com o pensamento computacional.

No que diz respeito ao conteúdo de Análise Combinatória e à utilização de elementos voltados para o pensamento computacional no ensino de matemática, duas habilidades da área de Matemática e suas Tecnologias na BNCC podem ser citadas:

> (EM13MAT310) Resolver e elaborar problemas de contagem envolvendo diferentes tipos de agrupamento de elementos, por meio dos princípios multiplicativo e aditivo, recorrendo a estratégias diversas como o diagrama de árvore. (BRASIL, 2018, p.529)
>
> (EM13MAT406) Utilizar os conceitos básicos de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática (BRASIL, 2018, p. 531)

A partir disso, mostrou-se interessante desenvolver um estudo que pudesse conciliar o desenvolvimento do pensamento computacional nos alunos, em conjunto com o ensino de tópicos presentes no conteúdo de Análise Combinatória. Dessa forma, elaborou-se a seguinte pergunta de pesquisa: "é possível trabalhar o pensamento computacional em conjunto com o ensino de Análise Combinatória?".

Para responder tal questionamento, tem-se como objetivo trabalhar o pensamento computacional por meio do desenvolvimento de calculadoras relacionadas à alguns tópicos do conteúdo de Análise Combinatória, a partir do programa App Inventor 2, para então analisar as contribuições dessa metodologia no processo de ensino e aprendizagem.

2. APRESENTAÇÃO DO PROGRAMA

O programa conhecido como MIT App Inventor é uma ferramenta online que foi originalmente criada pelo MIT (Instituto Tecnológico de Massachusetts) e que, posteriormente, foi adquirida pela empresa *Google*, para oferecer aos seus usuários uma solução tecnológica para a criação de aplicativos para dispositivos *Android* de uma forma mais simples e intuitiva (PRIETO, 2019).

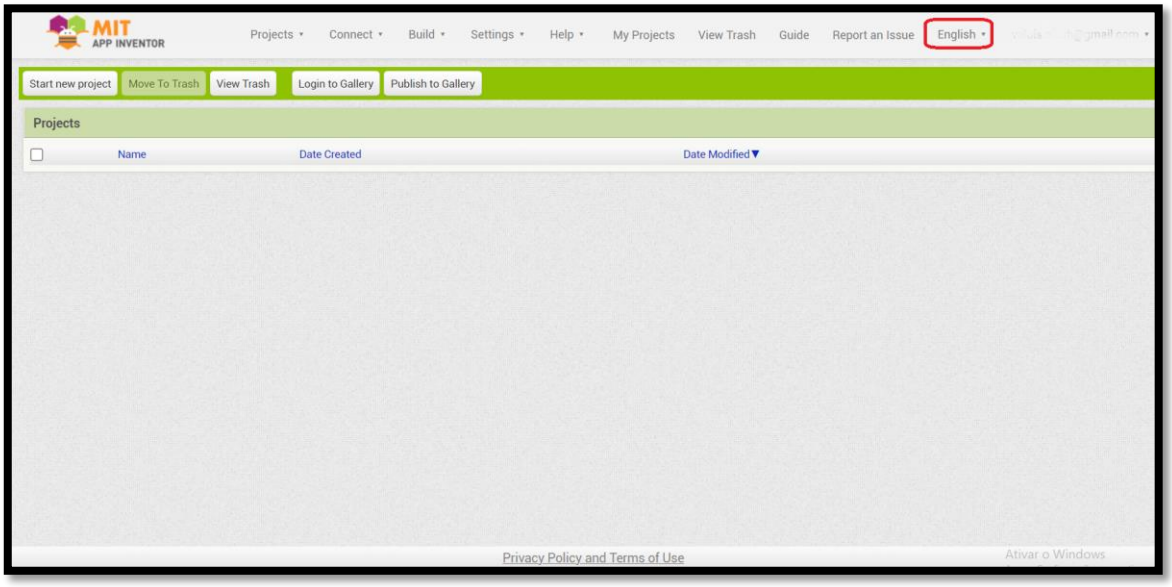
Os requisitos para utilizar o programa são: possuir um computador com conexão à *internet*; possuir uma conta de *e-mail* do *Google*; e é recomendado a utilização de um dos seguintes navegadores de *internet*: *Google Chrome* ou *Mozilla Firefox* (PRIETO, 2019). Vale ressaltar que o App Inventor pode ser acessado pelo endereço online a seguir: http:///ai2.appinventor.mit.edu.

Além disso, destaca-se que o processo de criação de um aplicativo através do App Inventor pode ser dividido em três etapas: Designer de telas; Programação em blocos; e Geração do aplicativo (PRIETO, 2019). Tais etapas serão retratadas a seguir, após evidenciar os passos para iniciar a construção de um novo aplicativo.

2.1. Criar um novo aplicativo

Assim que o usuário acessar o link do programa e fazer o *login* em sua conta do *Google*, ele terá acesso a tela inicial do programa, que inicialmente estará em com os comandos em inglês, mas isso pode ser alterado, como indicado na Figura 1.

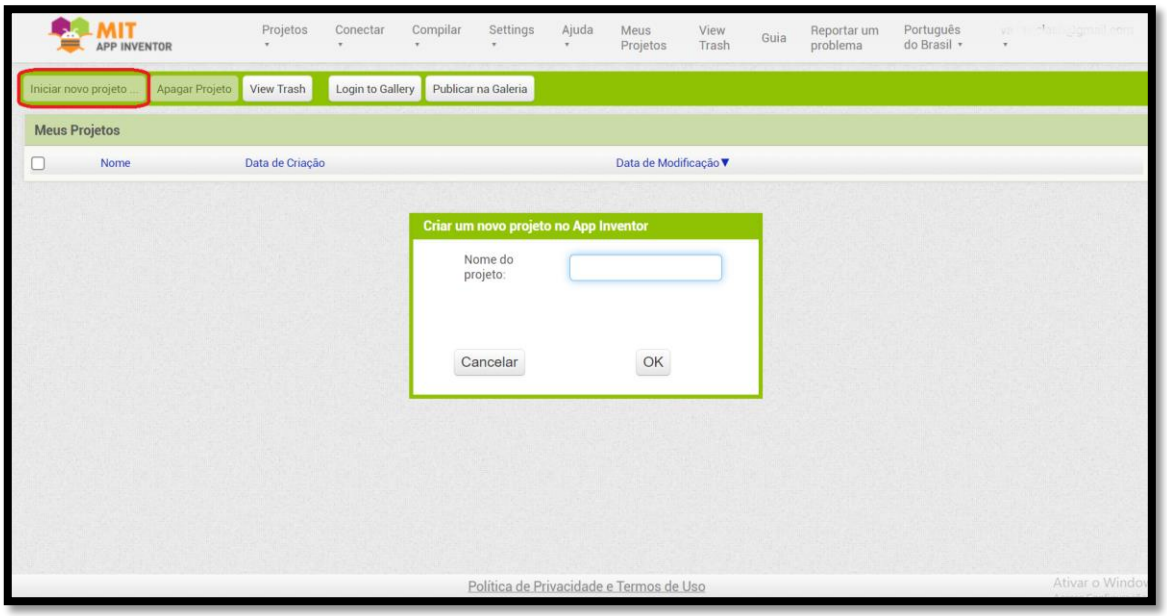
Figura 1: alterar o idioma da tela inicial do App Inventor 2.



Fonte: Acervo do autor (2022).

Feita a alteração do idioma, será possível iniciar a criação do aplicativo, basta selecionar a opção “Iniciar novo projeto ...” como indicado na Figura 2 e, em seguida, dar um nome ao seu aplicativo. Vale ressaltar que, não é possível utilizar caracteres especiais para nomear o seu projeto, como o *ç*, os sinais de adição, subtração, multiplicação e divisão, bem como os acentos da língua portuguesa e também não são permitidos os espaços.

Figura 2: iniciar um novo projeto no App Inventor 2.

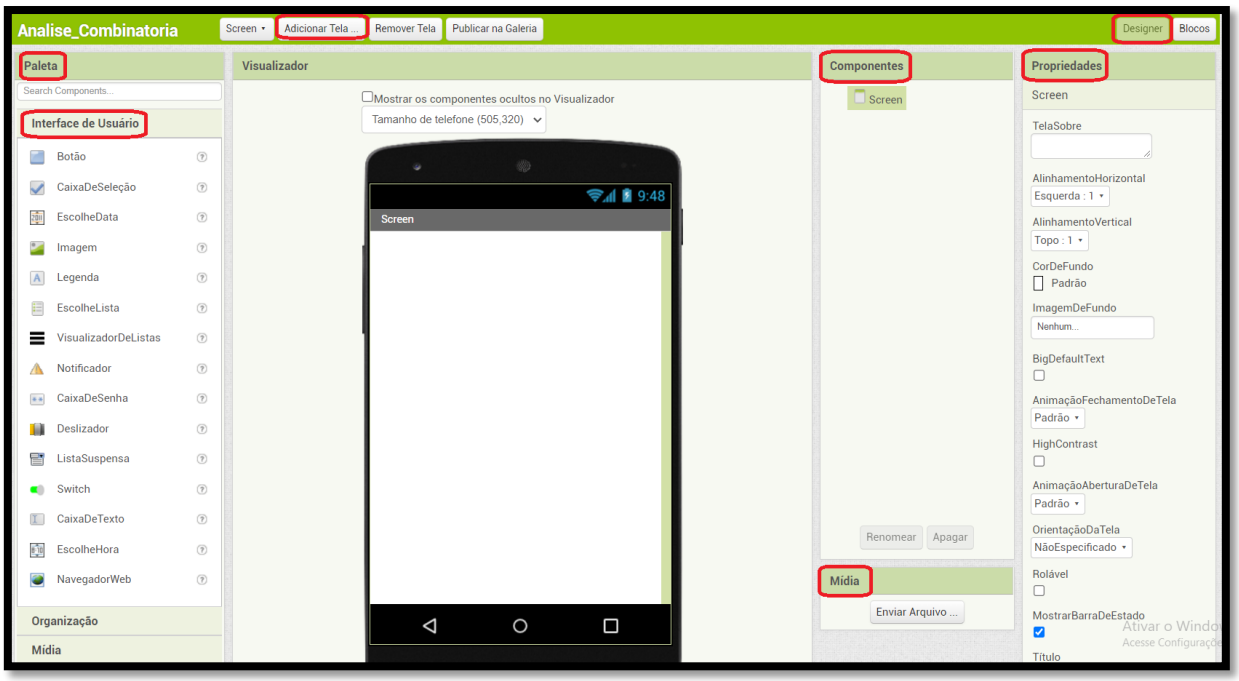


Fonte: Acervo do autor (2022).

2.2. Designer de telas

Após iniciar o projeto, o usuário será direcionado para a tela inicial do programa, que inicialmente se encontra em branco e deverá ser preenchida pelo criador do aplicativo. Ao observar a Figura 3, percebe-se os seguintes elementos de edição para realizar o designer das telas do aplicativo: a “Paleta” onde é possível pesquisar pelos componentes que serão adicionados a tela; a “Interface de Usuário” que é uma subdivisão da paleta, onde encontramos elementos como botões, legendas, caixas de texto, imagens, entre outros que podem compor a tela; a opção de “Adicionar Tela”, para criar uma nova tela que poderá, por meio da programação, estar aberta ou fechada durante o uso do aplicativo; os “Componentes” e “Mídia” que foram inseridos na tela ; e as “Propriedades” da tela, como cor de fundo, título, animação, alinhamento horizontal e vertical, entre outros.

Figura 3: tela de designer do App Inventor 2.



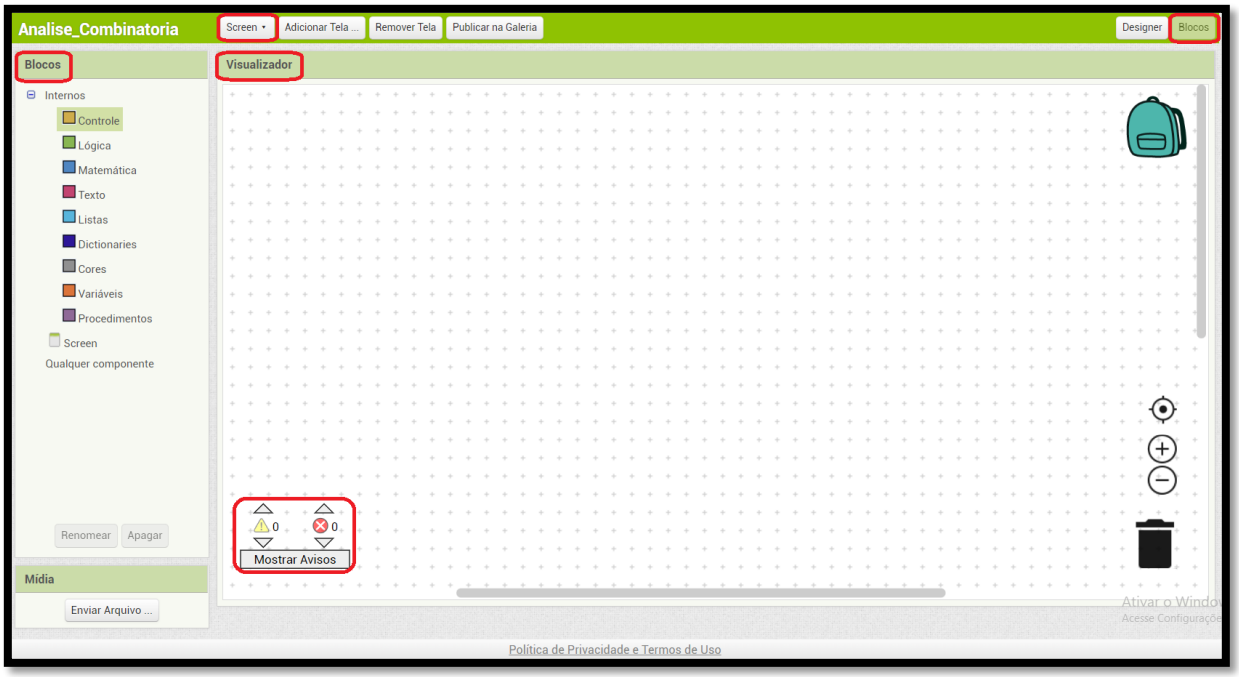
Fonte: Acervo do autor (2022).

Vale ressaltar que todos os componentes que forem inseridos na tela apresentarão um nome, que poderá ser renomeado com as mesmas condições apresentadas na criação do projeto: não podem ser utilizados caracteres especiais, acentos e nem espaço. Recomenda-se a renomeação dos componentes que serão programados, como botões e caixas de texto, por exemplo.

2.3. Programação em blocos

Feito o designer das telas, é necessário realizar a programação dos componentes necessários para o funcionamento do aplicativo. Na Figura 4 fica destacada a tela de programação em blocos e nela observa-se os “Blocos”, que são os comandos e elementos disponíveis para realizar as programações, o “Visualizador” onde é possível enxergar quais conteúdos são utilizados na programação da tela em questão, uma lista suspensa indicada pelo termo “Screen”, que indica em qual tela o usuário realiza a programação (e cada tela deverá ser programada individualmente) e ainda uma indicação chamada “Mostrar Avisos”, que diz respeito a possíveis incoerências feitas na programação que, quando indicadas pelo triângulo amarelo apenas sinalizam que algum bloco está sem utilidade, mas as indicadas pelo círculo vermelho alertam para algo que pode interferir no funcionamento do aplicativo.

Figura 4: tela de blocos do App Inventor 2.



Fonte: Acervo do autor (2022).

Cada bloco tem uma característica diferente dos demais, e cada um deles é indicado por uma cor, como visto na Figura 5, o que facilita para uma programação mais intuitiva. Durante a criação das calculadoras a seguir, as os blocos utilizados terão as cores correspondentes as categorias nas quais eles podem ser encontrados.

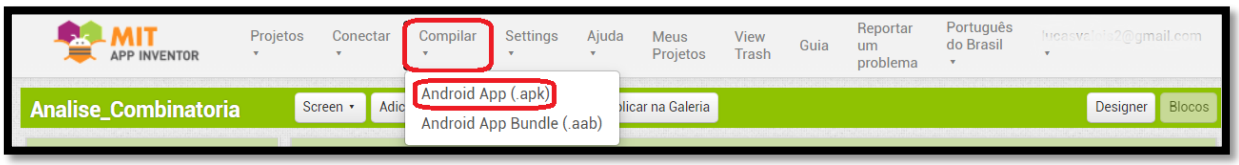
Figura 5: blocos de programação do App Inventor 2.

Fonte: Acervo do autor (2022).

2.4. Geração do aplicativo

Terminado o designer e a programação do aplicativo, é necessário compila-lo para poder testá-lo em um dispositivo *Android*. Para isso, busque na barra superior do site a opção "Compilar", como visto da Figura 6, e nela selecione a guia "Android App (.apk)".

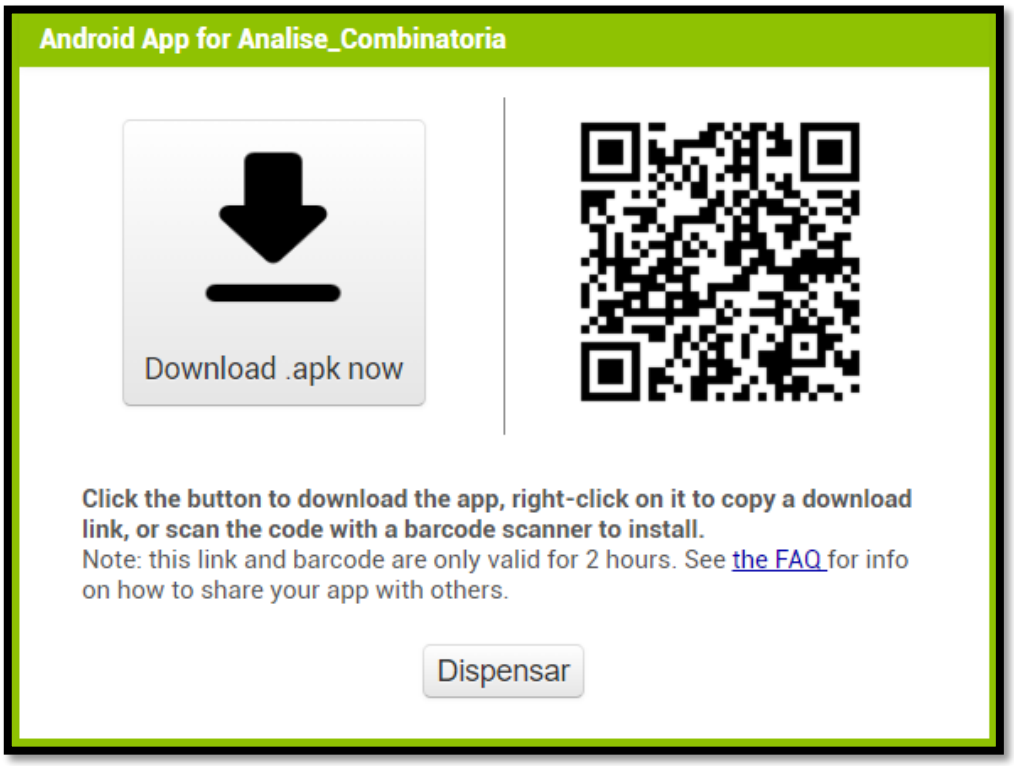
Figura 6: compilar aplicativo do App Inventor 2.



Fonte: Acervo do autor (2022).

Com isso, o programa começará o processo de compilação, que pode demorar alguns minutos, a depender do tamanho de seu aplicativo e da qualidade do computador e da internet utilizados no momento. Feita a compilação o usuário terá as duas opções indicadas na Figura 7 para instalar o aplicativo em seu dispositivo: fazer o *download do apk* para o seu computador e em seguida transferir esse arquivo para seu celular para então instala-lo; ou baixar e instalar o aplicativo *MIT AI2 Companion* em seu smartphone (disponível na Play Store ou na App Store) e então utiliza-lo para ler o *QR Code* gerado pelo site, que lhe dará a oportunidade de instalar diretamente o aplicativo.

Figura 7: instalar aplicativo do App Inventor 2.



Fonte: Acervo do autor (2022).

Feita a apresentação do programa que será utilizado nesse trabalho, deve-se partir para o alinhamento dos conhecimentos obtidos sobre o App Inventor 2 com o conteúdo matemático de Análise Combinatória, que será realizado a partir da construção de uma calculadora que trabalhe elementos desse objeto matemático.

3. ANÁLISE COMBINATÓRIA NO APP INVENTOR 2

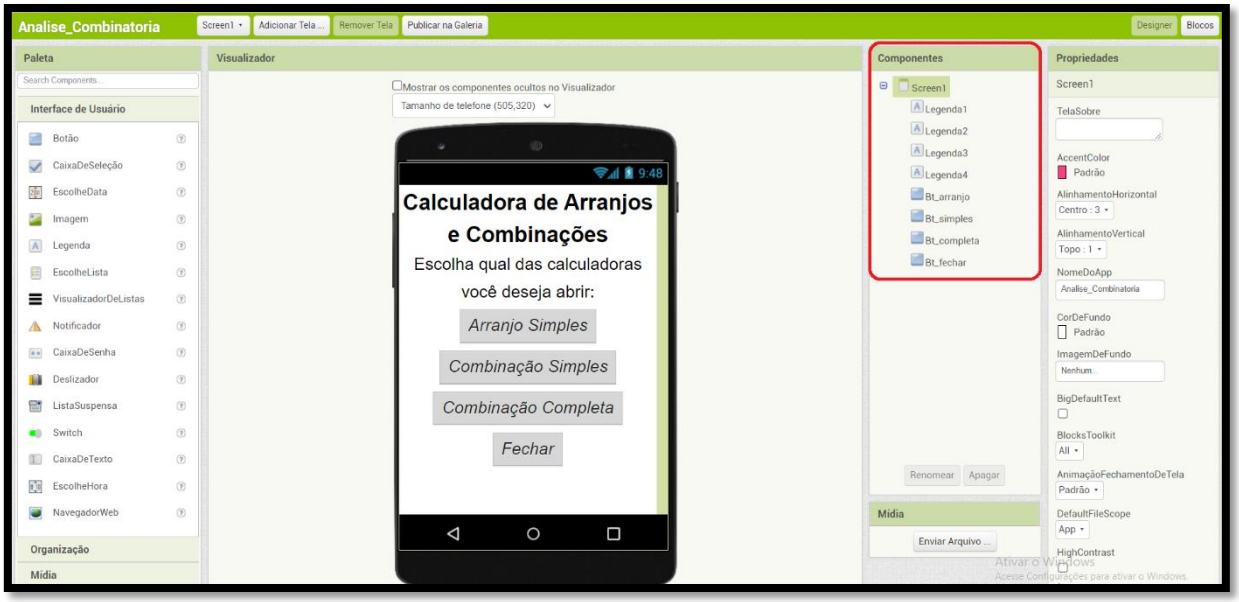
Como já exposto, esse trabalho pretende conciliar o ensino de alguns tópicos de Análise Combinatória com o pensamento computacional, por meio da utilização do App Inventor 2. Para isso, será apresentada a construção de três calculadoras, as quais trabalham, cada uma delas, com um tópico diferente, relacionados ao conteúdo matemático em questão.

Segundo Hazzan (2004), a Análise Combinatória visa desenvolver métodos que permitam contar o número de elementos de um conjunto, sendo estes elementos agrupamentos formados sob certas condições. Dentre essas condições, destacam-se três a seguir: os agrupamentos formados a partir de um conjunto com *n* elementos, no qual tomamos *p* elementos *distintos* para formar uma *sequência*, que recebe o nome de *Arranjo Simples* (HAZZAN, 2004); os agrupamentos formados a partir de um conjunto com *n* elementos, no qual tomamos *p* elementos *distintos* para formar um *subconjunto*, que recebe o nome de *Combinação Simples* (HAZZAN, 2004); e os agrupamentos formados a partir de um conjunto

com *n* elementos, no qual tomamos *p* elementos *distintos ou não* para formar um *novo conjunto*, que recebe o nome de *Combinação Completa* (MORGADO et al, 2016).

A partir disso, o aluno precisará, para resolver as situações problemas apresentadas a seguir, primeiramente identificar em qual das três situações citadas acima a questão se enquadra e, feita essa identificação, ele escolherá a calculadora que será utilizada em sua resolução. Na Figura 8 observa-se a tela inicial do aplicativo, que será construída pelos estudantes, e que possibilita o acesso às calculadoras disponíveis.

Figura 8: design da tela inicial do aplicativo.

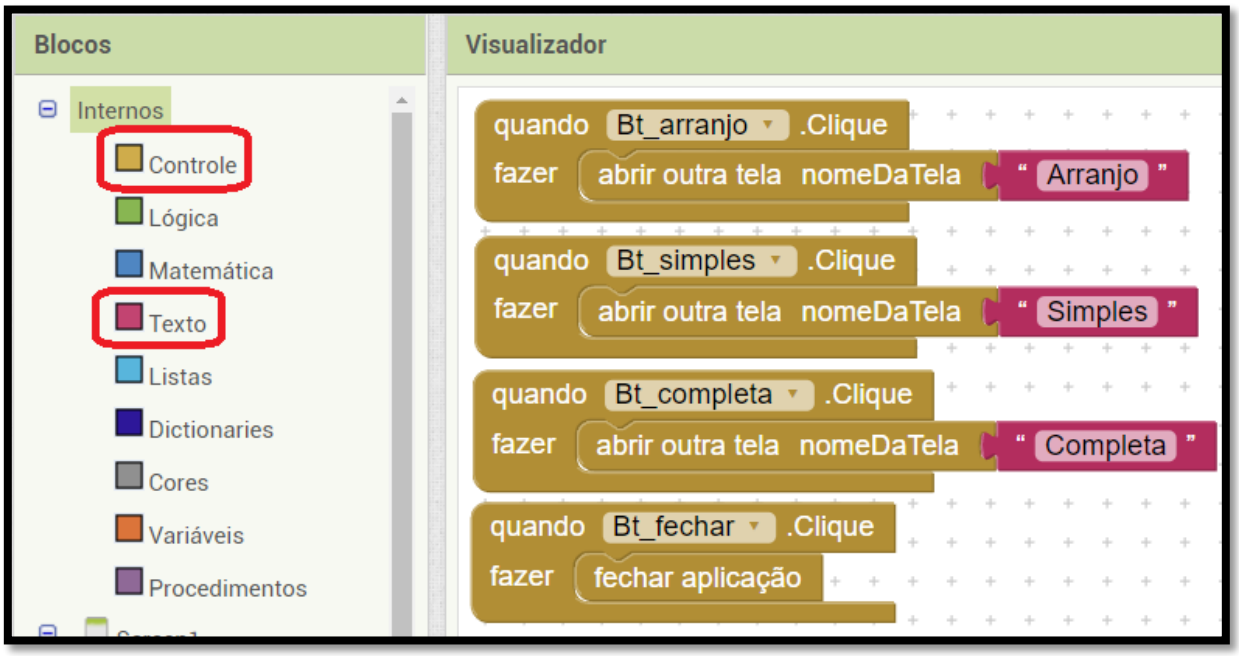


Fonte: Acervo do autor (2022).

Para construir esse menu inicial, foram utilizados os seguintes componentes: “legendas” para os textos presentes na parte superior da tela do celular, que não necessitam ser renomeadas, visto que não serão utilizadas na programação; e “botões” para acessar as três demais telas presentes no aplicativo, e para fechar o programa após sua utilização. Recomenda-se que os alunos renomeiem os botões na aba “Componentes”, como visto na Figura 8, pois se fará necessário diferencia-los durante a programação.

Concluído o design do menu inicial, deve-se realizar a programação dessa tela, para que os botões tenham funcionalidade. Essa programação, como observado na Figura 9, fará uso apenas dos blocos presentes na categoria “Controle”, que serão responsáveis por dar o comando que será efetuado ao clicar nos botões, e blocos de “Texto”, nos quais estarão presentes os nomes das demais telas do aplicativo.

Figura 9: programação da tela inicial do aplicativo.



Fonte: Acervo do autor (2022).

É importante ratificar que os nomes escolhidos para as telas nessa programação devem ser rigorosamente iguais aos nomes dados para as futuras telas do aplicativo pois, caso contrário, o software não funcionará corretamente.

3.1. Arranjo Simples

De acordo com Hazzan (2004) seja N um conjunto com *n* elementos, chamamos de arranjo simples dos *n* elementos tomados *p* a *p* ($1 \leq p \leq n$) a qualquer sequência de *p* termos formada com elementos de N, todos distintos. Por se tratar de uma sequência, sabe-se então que a ordem dos elementos escolhidos importa, ou seja, a sequência (*a*, *b*, *c*) é diferente da sequência (*c*, *b*, *a*) e assim por diante.

A partir disso, estabelece-se uma fórmula para calcular a quantidade de arranjos simples a partir do número *n* de elementos escolhidos e do valor *p* de termos para a sequência.

$$A_{n,p} = \frac{n!}{(n-p)!}$$

Para melhor compreensão dessa fórmula, Hazzan (2004) parte de um número inteiro não negativo *m* para definir o fatorial de *m* (indicado por *m!*) da seguinte forma:

- $m! = m * (m-1) * (m-2) * \ldots * 3 * 2 * 1$, para $m \geq 2$.
- $1! = 1$.
- $0! = 1$.

Portanto, a fórmula estabelecida inicialmente pode ser lida como “O número de arranjos simples de *n* elementos, tomados *p* a *p*, é igual ao fatorial de *n*, dividido pelo fatorial de *n* – *p*.”

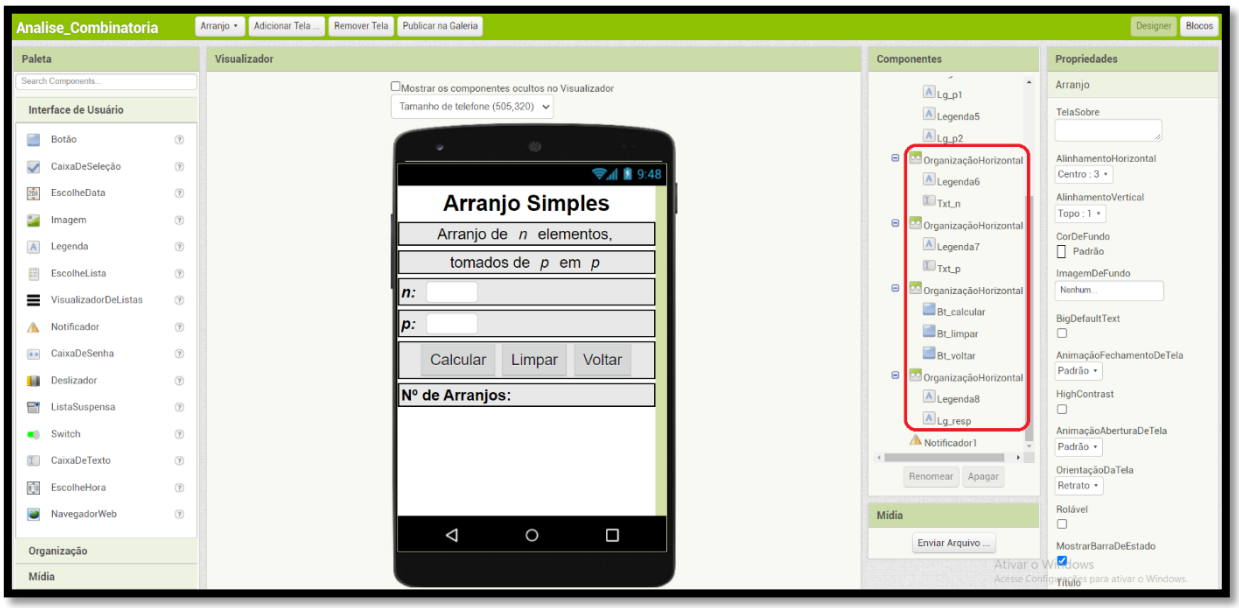
Com isso, propõe-se uma situação problema que trabalhe em conjunto o conhecimento do conteúdo de Arranjo Simples com o pensamento computacional através do App Inventor 2.

Situação problema: Para acessar um site, cada usuário cria uma senha com 4 algarismos, todos distintos entre si. Então, qual o número de senhas possíveis que esse site admite?

Resolução: como uma senha formada com 4 algarismos distintos pode ser vista como uma sequência de 4 termos diferentes, dentre os 10 algarismos existentes, então é possível resolver essa problemática através do cálculo de um Arranjo Simples. Dessa forma, será construída uma calculadora de Arranjo Simples, com auxílio do App Inventor 2, para resolver esta e qualquer outra situação que envolva o cálculo de Arranjos Simples.

Nessa calculadora, o usuário deverá inserir o valor do número total de elementos disponíveis para serem utilizados e o valor da quantidade de termos que ele deseja em sua sequência. Para isso, como visto na Figura 10, o design dessa tela deve conter os seguintes componentes: as “legendas” com os textos que podemos observar da tela do smartphone; as “caixas de texto” nas quais o usuário irá inserir os valores de *n* e de *p*; os “botões” que serão utilizados para realizar o cálculo, a limpeza da calculadora e para voltar ao menu inicial; e um “notificador”, que não é visível na tela, mas será importante durante a programação.

Figura 10: design da calculadora de Arranjo Simples.



Fonte: Acervo do autor (2022).

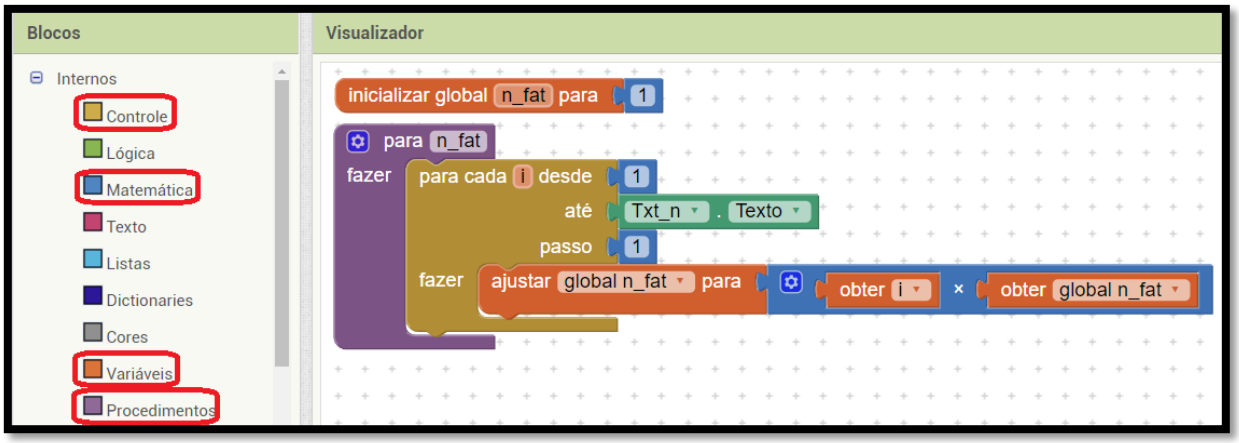
Vale salientar a recomendação de renomear tanto as caixas de textos quanto os botões presentes nos componentes, pois eles serão utilizados durante a programação. Além disso, pode-se observar na Figura 10 que um dos últimos componentes é uma legenda nomeada como "Lg_resp", que corresponde a uma legenda em branco, colocada ao lado da legenda "Nº de Arranjos:" que será o local onde aparecerá a resposta desejada após inserir os dados e pressionar o botão "Calcular".

Feito o design da calculadora, inicia-se a programação em blocos, que se encontra dividida nos passos que seguem:

PASSO 1: criar um procedimento para calcular o *n!*.

Inicialmente deve-se construir uma maneira de calcular o fatorial de um número, pois essa não é uma operação presente no programa. Para isso, como visto na Figura 11, deve-se primeiramente iniciar uma "variável", que foi chamada de "n_fat" e, em seguida, criar o "procedimento" nomeado também como "n_fat", o qual consiste em realizar sucessivas multiplicações, nas quais os valores sempre variam em uma unidade, que vão desde o *1* até o *n* inserido pelo usuário e então armazenar o valor dessa operação como o novo valor da variável "n_fat".

Figura 11: programação do procedimento *n!*.



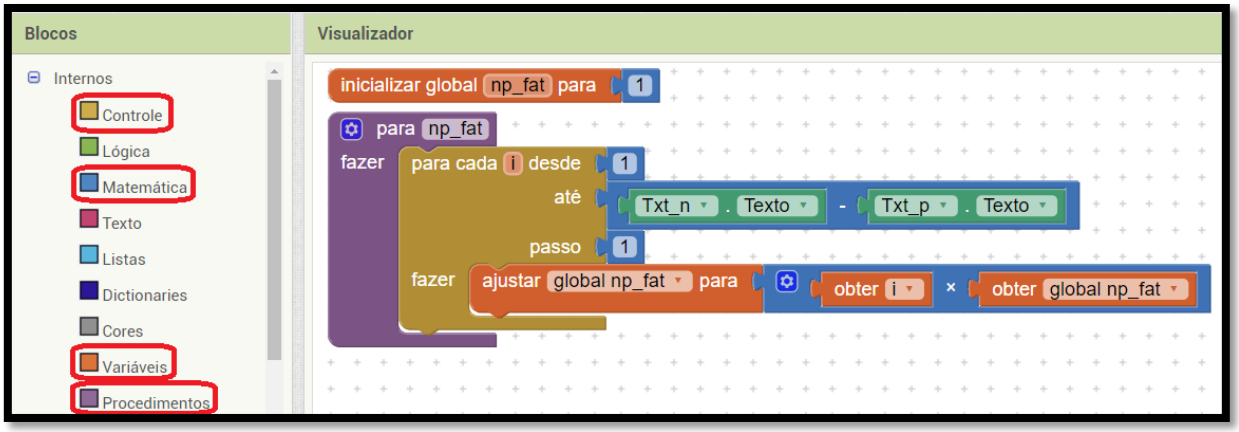
Fonte: Acervo do autor (2022).

Com isso, está feita a programação da operação *n!*, que não existia previamente no App Inventor 2. Uma explicação para o que ocorre nesse procedimento é, quando ele for solicitado ele irá ajustar o valor da variável "n_fat" para o valor obtido da multiplicação de *i* (que varia desde *1* até o valor escolhido de *n*, ao passo de uma unidade) pelo valor da própria variável "n_fat", ou seja, faz-se o cálculo $1 * 2 * 3 * ... * (n - 1) * n$, que consiste na definição de *n!*.

PASSO 2: criar um procedimento para calcular o *(n-p)!*.

De forma análoga ao que foi feito para calcular o *n!*, deverás ser feito para o *(n-p)!*. Assim, como observa-se na Figura 12, deve-se iniciar uma "variável", chamada de "np_fat" e, em seguida, criar o "procedimento" nomeado também como "np_fat", que irá realizar sucessivas multiplicações, nas quais os valores sempre variam em uma unidade, que vão desde o *1* até o *n-p*, onde tanto o *n* quanto o *p* foram inseridos pelo usuário, e então será armazenado o valor dessa operação como o novo valor da variável "np_fat".

Figura 12: programação do procedimento *(n-p)!*.



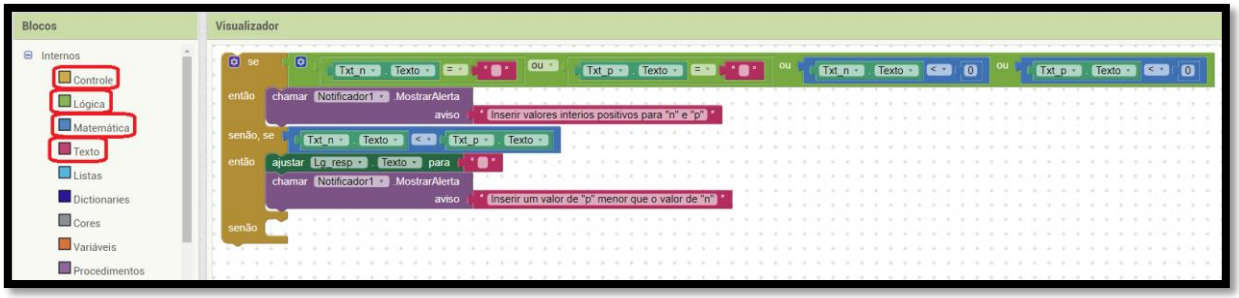
Fonte: Acervo do autor (2022).

Dessa forma, está feita a programação da operação *(n-p)!*, que é muito similar a anterior, apenas com a diferença de que os valores que serão multiplicados vão de *1* até *n-p*.

PASSO 3: Estabelecer as restrições para os valores de *n* e *p*.

Como visto na definição do Arranjo, *n* e *p* devem ser valores inteiros e positivos, com $1 \leq p \leq n$, portanto deve-se estabelecer essa restrição no aplicativo, para não permitir que valores fora dessas condições sejam inseridos pelos usuários. Deste modo, ao observar a Figura 13 nota-se o uso de um bloco de "Controle" com as condicionais "se ... então ... senão, se ... então ... senão" que será utilizado para programar essas ressalvas aos valores de *n* e *p*. Utiliza-se também o "Notificador" que foi inserido no design, pois ele enviará uma mensagem para o usuário referente à restrição que ele não cumpriu ao escolher os valores para *n* e *p*.

Figura 13: programação das restrições de *n* e *p*.



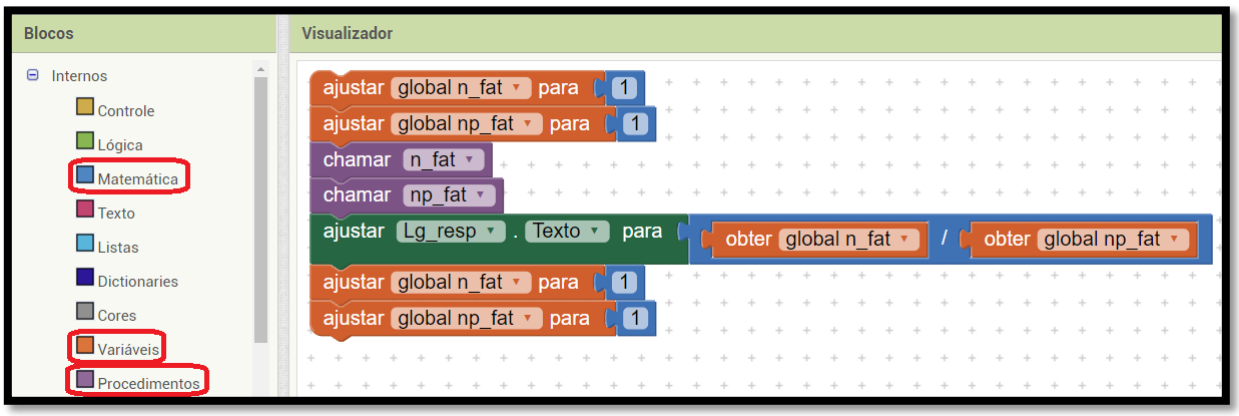
Fonte: Acervo do autor (2022).

Com isso, torna-se vetada a possibilidade de o usuário tentar calcular o Arranjo Simples sem inserir valores para *n* e *p*, bem como tentar entrar com números negativos para essas variáveis, uma vez que caso isso aconteça o aviso “Inserir valores inteiros positivos para *n* e *p*” será exibido em sua tela. Ademais a isso, caso ocorra a tentativa de realizar o cálculo com $p > n$ o alerta “Inserir um valor de *p* menor que o valor de *n*" será mostrado. Para concluir, não é possível impossibilitar a inserção de valores não inteiros, como *3,4* ou *7,9*, entretanto o aplicativo não faz a leitura desses valores, ajustando-os para *3* e *7*, respectivamente.

PASSO 4: programar o cálculo do Arranjo.

Quando o usuário finalmente inserir valores válidos para *n* e *p*, então o programa fará o cálculo do Arranjo Simples de *n* elementos, tomados *p* a *p*. Para isso, deve-se realizar a programação observada na Figura 14. Os blocos utilizados consistem em ajustar as “variáveis” sempre para o valor neutro multiplicativo, tanto no início quanto ao final do comando, bem como solicitar que os “procedimentos” criados anteriormente sejam realizados para, enfim, aplicar a fórmula do Arranjo Simples.

Figura 14: programação do cálculo do Arranjo.



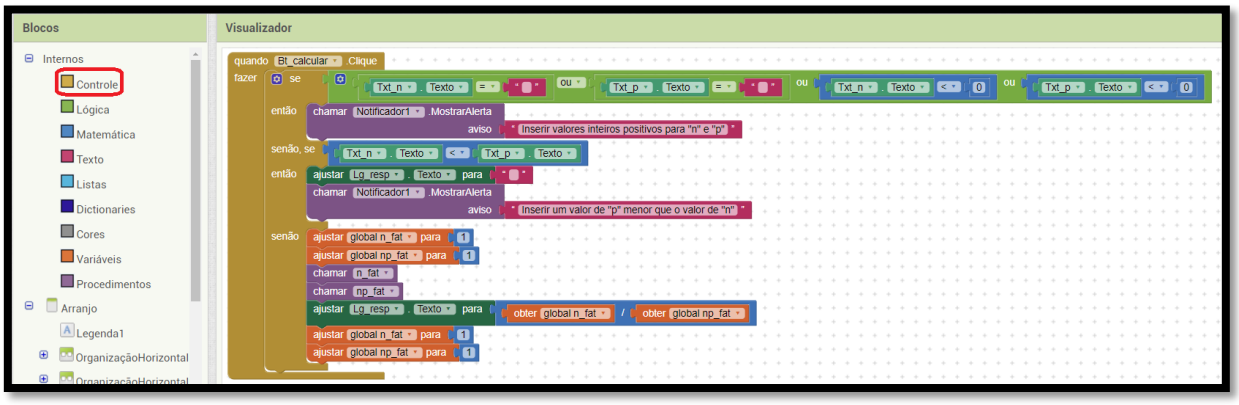
Fonte: Acervo do autor (2022).

Observa-se ainda que será feito o ajuste do texto da “legenda” intitulada “Lg_resp” que, quando colocada na tela de design se encontrava em branco, mas agora assumirá o valor da resposta à conta que é solicitada.

PASSO 5: programar o botão “Calcular”.

Finalmente, para que todos os procedimentos listados acima sejam realizados, faz-se necessário programar o botão “Calcular”, para que quando ele for pressionado, os comandos estabelecidos sejam realizados. Sua programação pode ser encontrada na Figura 15, e consiste em juntar os blocos “Quando Bt_calcular .Clique fazer”, com as programações realizadas nos passos 3 e 4.

Figura 15: programação do botão “Calcular”.



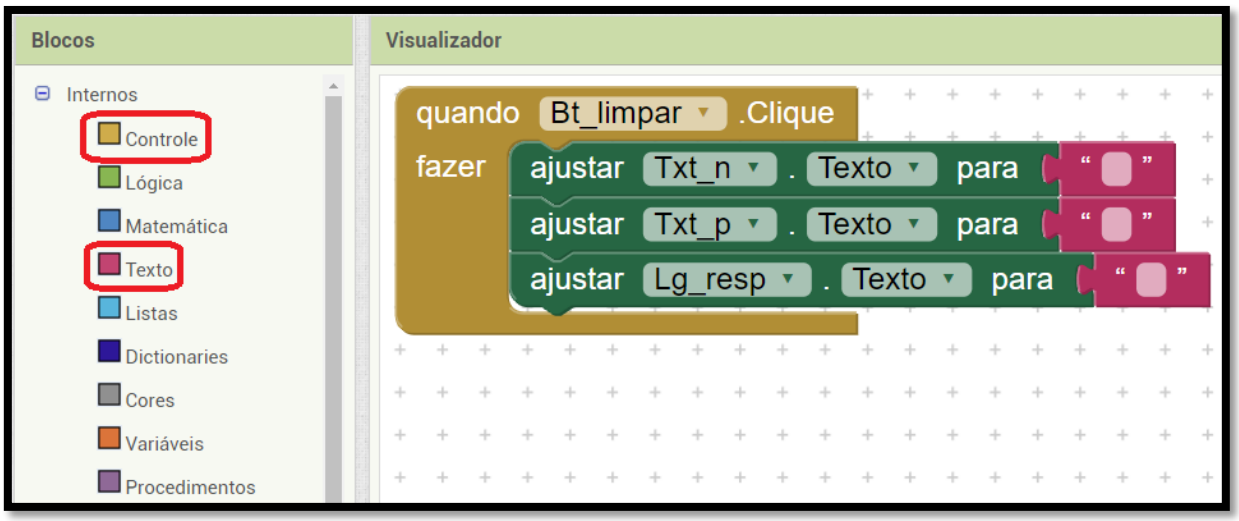
Fonte: Acervo do autor (2022).

Dessa forma, quando o usuário pressionar o botão “Calcular”, o aplicativo primeiramente verificará se os valores podem ser operacionados, de acordo com as restrições já mostradas, para então realizar o cálculo e fornecer o seu valor no lugar da “Lg_resp”.

PASSO 6: programar o botão “Limpar”.

Após programar o primeiro botão, parte-se para os comandos a serem realizados quando pressionado o botão “Limpar”. Essas instruções podem ser observadas na Figura 16 e consistem apenas em esvaziar as “caixas de texto” das variáveis *n* e *p*, e da “legenda” na qual a resposta aparece.

Figura 16: programação do botão “Limpar”.

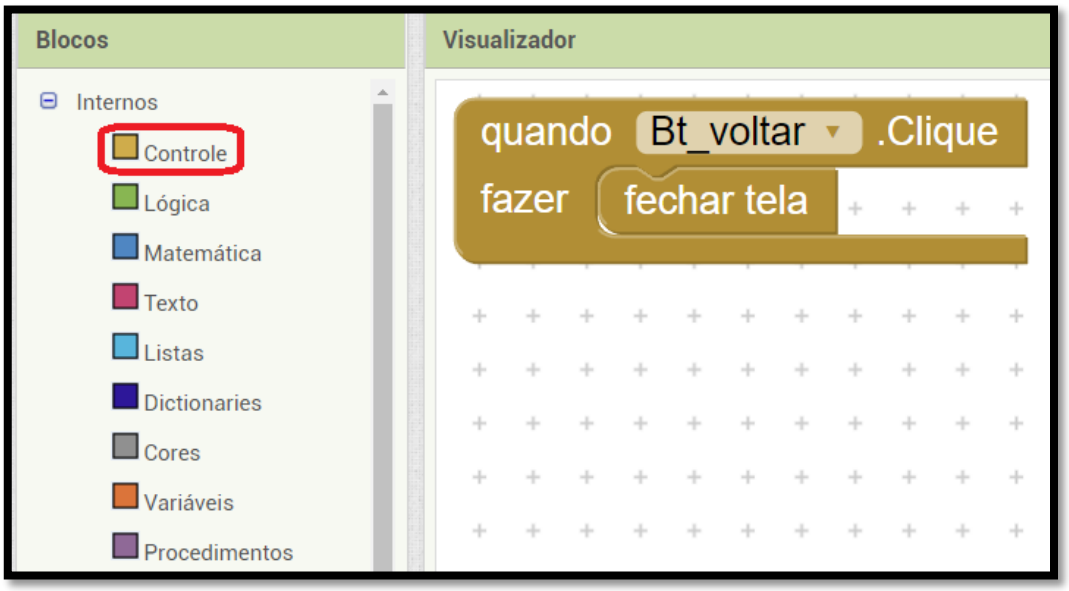


Fonte: Acervo do autor (2022).

PASSO 7: programar o botão “Voltar”.

Por fim, deve-se programar o último botão, que será utilizado para retornar ao menu inicial do aplicativo. A Figura 17 indica sua programação e essa fará com que, assim que o usuário pressione o botão “Voltar”, ele será redirecionado para a tela de escolha das calculadoras.

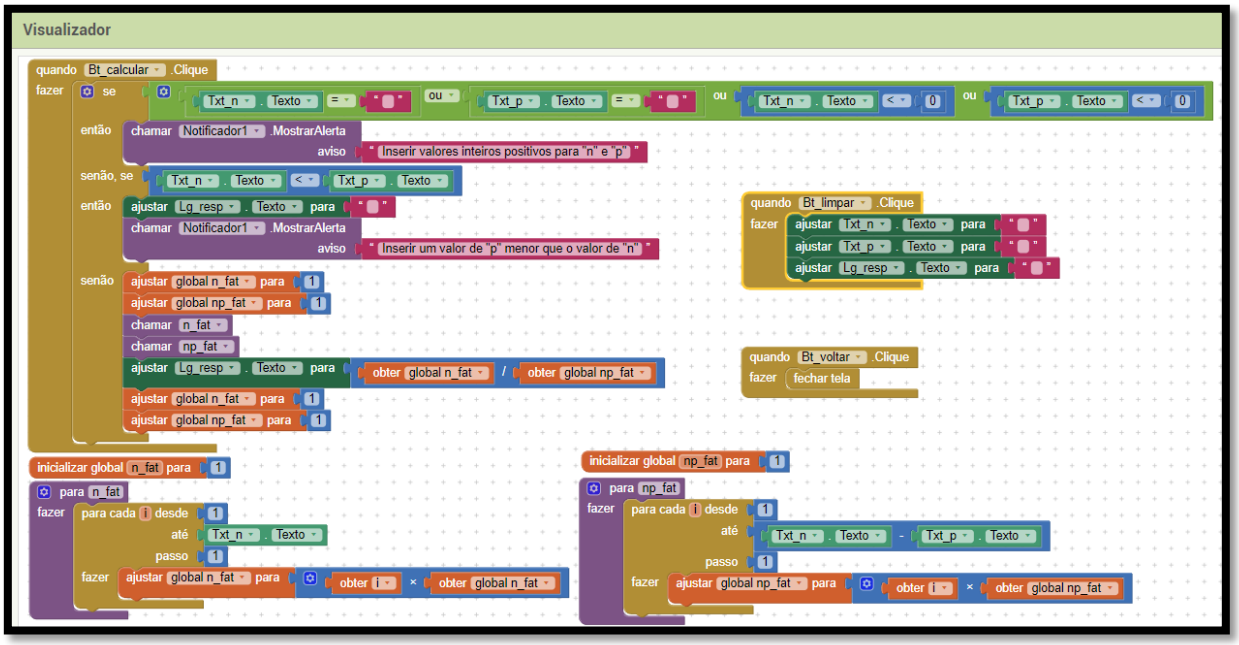
Figura 17: programação do botão “Voltar”.



Fonte: Acervo do autor (2022).

Dessa forma, conclui-se toda a programação da calculadora de Arranjos Simples. Na Figura 18 observa-se a programação completa.

Figura 18: programação completa do Arranjo Simples.



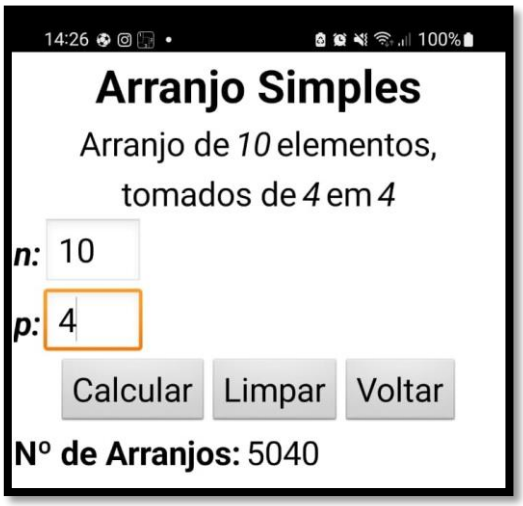
Fonte: Acervo do autor (2022).

PASSO 8: resolver a situação problema com auxílio da calculadora.

Finalizada a construção da calculadora, e após instalar ela em seu smartphone, o aluno pode retornar à situação problema apresentada anteriormente: “Para acessar um site, cada usuário cria uma senha com 4 algarismos, todos distintos entre si. Então, qual o número de senhas possíveis que esse site admite?”.

Como já discutido, trata-se de um problema que envolve o cálculo do Arranjo Simples (por se tratar do número de sequências com elementos distintos existentes) de 10 elementos (os 10 algarismos existentes no sistema decimal) tomados 4 a 4 (a quantidade de algarismo que terá cada senha). Na Figura 19 observa-se a tela da calculadora, já em funcionamento em um celular, após a realização do cálculo em questão.

Figura 19: calculadora de Arranjos Simples em funcionamento.



Fonte: Acervo do autor (2022).

3.2. Combinação Simples

De acordo com Hazzan (2004) seja N um conjunto com *n* elementos, chamamos de combinação simples dos *n* elementos tomados *p* a *p* ($1 \leq p \leq n$) a qualquer subconjunto de *p* termos formada com elementos de N, todos distintos. Por se tratar de um subconjunto de N, sabe-se então que a ordem dos elementos escolhidos não importa, ou seja, a subconjunto (*a*, *b*, *c*) é equivalente ao subconjunto (*c*, *b*, *a*) e assim por diante. Dessa forma, observa-se a clara diferença entre o Arranjo Simples e a Combinação Simples, uma vez que no primeiro a ordem dos elementos importa e no segundo, não importa.

A partir disso, estabelece-se uma fórmula para calcular a quantidade de combinações simples a partir do número *n* de elementos escolhidos e do valor *p* de termos para a sequência.

$$C_{n,p} = \frac{n!}{p!\,(n-p)!}$$

Portanto, a fórmula acima pode ser lida como "O número de combinações simples de *n* elementos, tomados *p* a *p*, é igual ao fatorial de *n*, dividido pelo fatorial de *p* multiplicado pelo fatorial de *n* – *p*."

Com isso, propõe-se uma situação problema que trabalhe em conjunto o conhecimento do conteúdo de Combinação Simples com o pensamento computacional através do App Inventor 2.

Situação problema: Para apostar na Mega-Sena, faz-se necessário escolher 6 números distintos, dentre os 60 disponíveis para escolha, para assim ter "um jogo" válido para aposta. A partir disso, qual o total de "jogos" possíveis que existem na Mega-Sena?

Resolução: como um jogo é formado a partir da escolha de 6 números distintos, em um universo de 60 disponíveis, basta-se estabelecer um subconjunto com 6 elementos a partir do conjunto inicial de 60 elementos, ou seja, é necessário e suficiente calcular o número de Combinações Simples de 60, tomados 6 a 6. Dessa forma, será construída uma calculadora de Combinação Simples, com auxílio do App Inventor 2, para resolver esta e qualquer outra situação que envolva o cálculo de Combinações Simples.

Nessa calculadora, o usuário deverá inserir o valor do número total de elementos disponíveis para serem utilizados e o valor da quantidade de termos que ele deseja em seu subconjunto. Para isso, como visto na Figura 20, o design dessa tela deve conter os seguintes componentes: as "legendas" com os textos que podemos observar da tela do smartphone; as "caixas de texto" nas quais o usuário irá inserir os valores de *n* e de *p*; os "botões" que serão utilizados para realizar o cálculo, a limpeza da calculadora e para voltar ao menu inicial; e um "notificador", que não é visível na tela, mas será importante durante a programação.

Figura 20: design da calculadora de Combinação Simples.

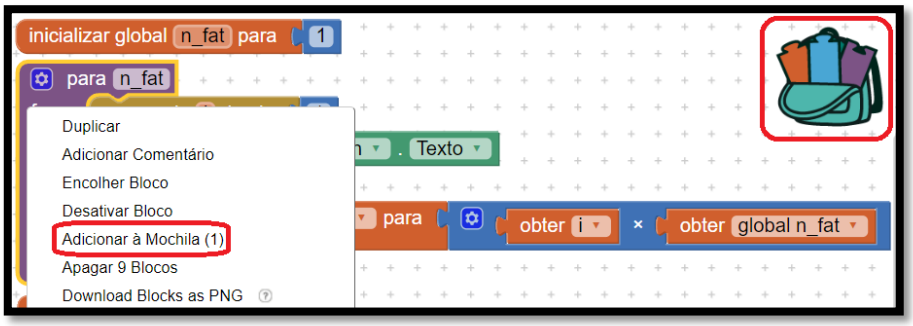
Fonte: Acervo do autor (2022).

Vale salientar a recomendação de renomear tanto as caixas de textos quanto os botões presentes nos componentes, pois eles serão utilizados durante a programação. Além disso, pode-se observar na Figura 20 que um dos últimos componentes é uma legenda nomeada como

“Lg_resp”, que corresponde a uma legenda em branco, colocada ao lado da legenda “Nº de Combinações:” que será o local onde aparecerá a resposta desejada após inserir os dados e pressionar o botão “Calcular”.

Feito o design da calculadora, inicia-se a programação em blocos. Nota-se que muitos dos blocos que serão utilizados nessa etapa são semelhantes ou até mesmo exatamente iguais aos já construídos anteriormente, então é interessante utilizar a opção “Adicionar à Mochila” para salvar esse conjunto de blocos da tela anterior e apenas copiá-los na próxima tela. A Figura 21 destaca essa funcionalidade e, em seguida, temos os passos da programação.

Figura 21: salvar blocos já construídos em outra tela.

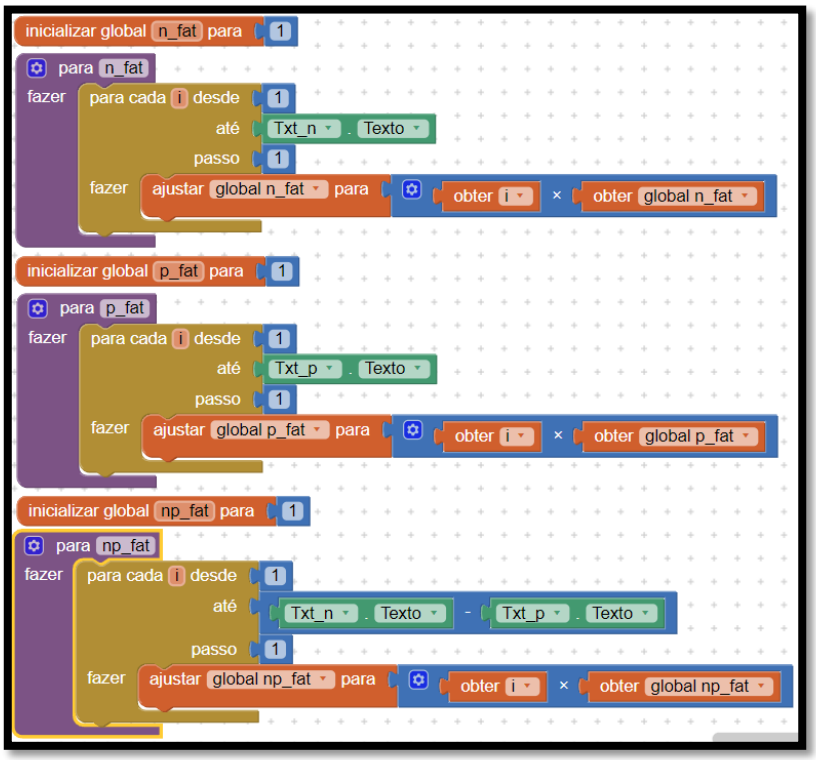


Fonte: Acervo do autor (2022).

PASSO 1: criar os procedimento para calcular o *n!*, o *p!* e o *(n-p)!*.

A construção desses procedimentos será rigorosamente igual aos procedimentos criados na calculadora de Arranjos Simples, como observado na Figura 22.

Figura 22: programação dos procedimentos *n!*, *p!* e *(n-p)!*.

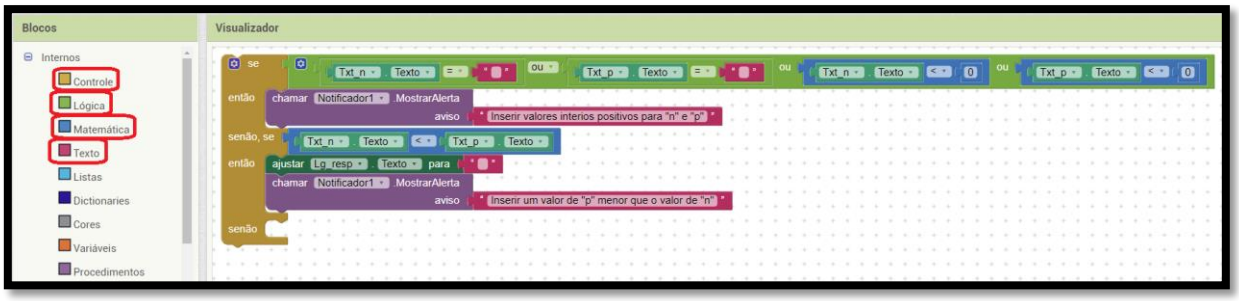


Fonte: Acervo do autor (2022).

PASSO 2: Estabelecer as restrições para os valores de *n* e *p*.

As restrições aqui estabelecidas serão exatamente as mesmas que aquelas programadas na calculadora anterior, assim como é visto na Figura 23.

Figura 23: programação das restrições de *n* e *p*.

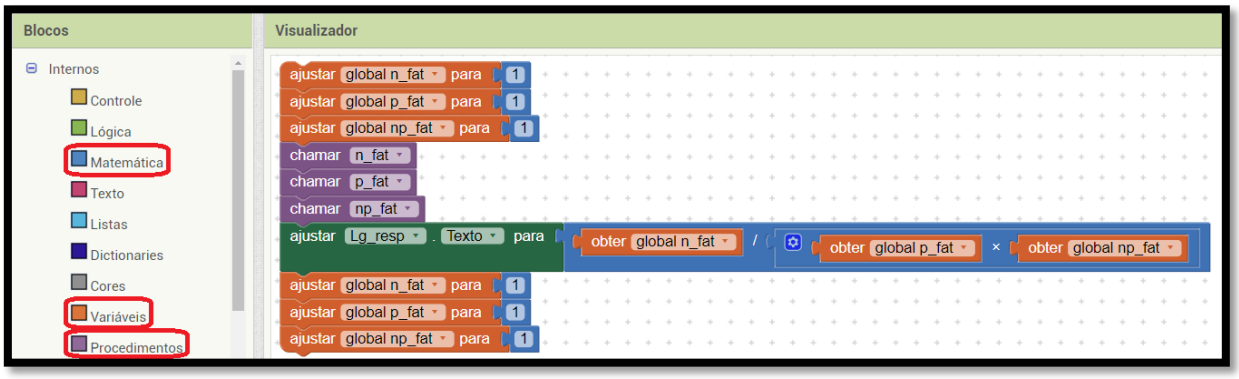


Fonte: Acervo do autor (2022).

PASSO 3: programar o cálculo da Combinação Simples.

Quando o usuário finalmente inserir valores válidos para *n* e *p*, então o programa fará o cálculo da Combinação Simples de *n* elementos, tomados *p* a *p*. Para isso, deve-se realizar a programação observada na Figura 24. Os blocos utilizados consistem em ajustar as "variáveis" sempre para o valor neutro multiplicativo, tanto no início quanto ao final do comando, bem como solicitar que os "procedimentos" criados anteriormente sejam realizados para, enfim, aplicar a fórmula da Combinação Simples.

Figura 24: programação do cálculo da Combinação Simples.



Fonte: Acervo do autor (2022).

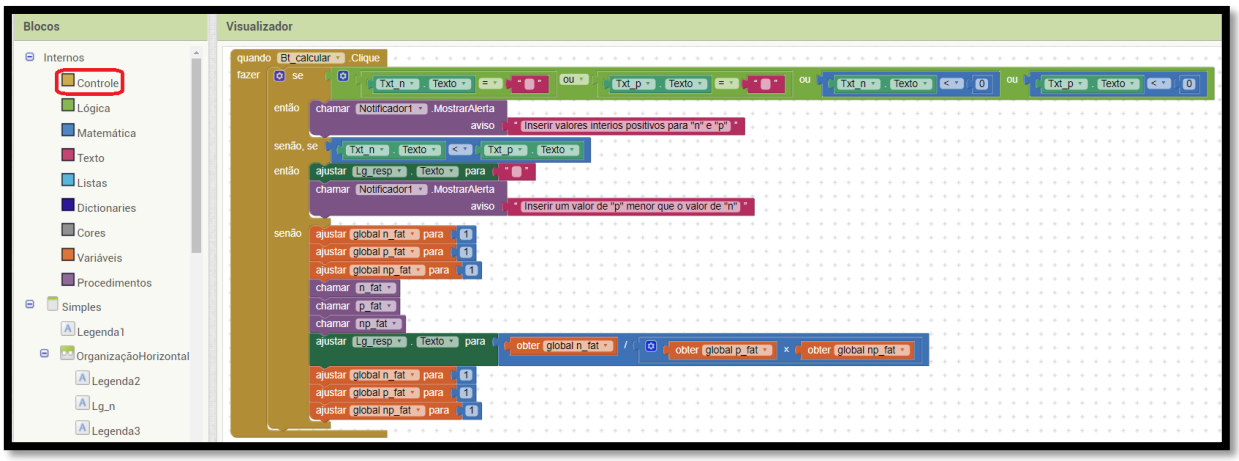
Observa-se ainda que será feito o ajuste do texto da "legenda" intitulada "Lg_resp" que, quando colocada na tela de design se encontrava em branco, mas agora assumirá o valor da resposta à conta que é solicitada.

PASSO 4: programar o botão "Calcular".

Finalmente, para que todos os procedimentos listados acima sejam realizados, faz-se necessário programar o botão "Calcular", para que quando ele for pressionado, os comandos estabelecidos sejam realizados. Sua programação pode ser encontrada na Figura 25, e consiste

em juntar os blocos “Quando Bt_calcular .Clique fazer”, com as programações realizadas nos passos 2 e 3.

Figura 25: programação do botão “Calcular”.



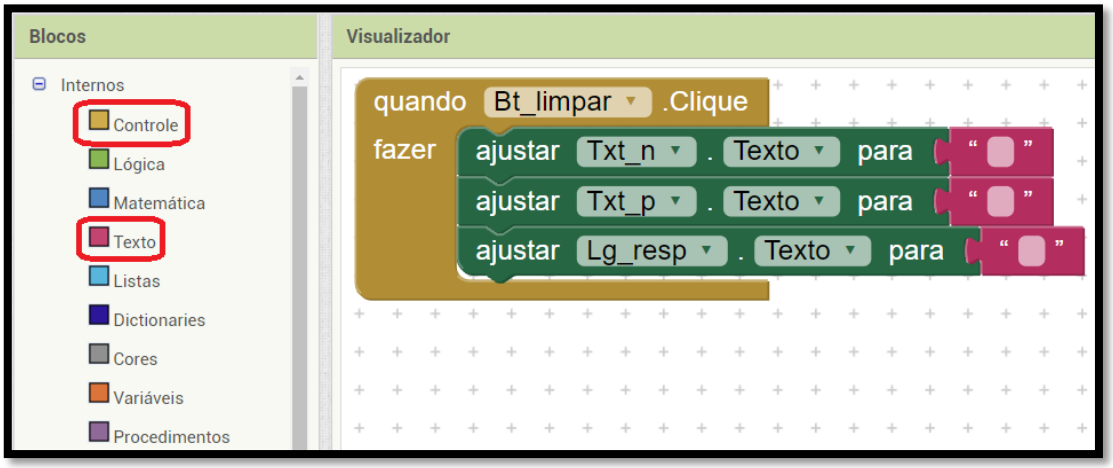
Fonte: Acervo do autor (2022).

Dessa forma, quando o usuário pressionar o botão “Calcular”, o aplicativo primeiramente verificará se os valores podem ser operacionados, de acordo com as restrições já mostradas, para então realizar o cálculo e fornecer o seu valor no lugar da “Lg_resp”.

PASSO 5: programar o botão “Limpar”.

Após programar o primeiro botão, parte-se para os comandos a serem realizados quando pressionado o botão “Limpar”. Essas instruções podem são as mesmas da calculadora de Arranjo Simples, como mostrado na Figura 26 e consistem apenas em esvaziar as “caixas de texto” das variáveis *n* e *p*, e da “legenda” na qual a resposta aparece.

Figura 26: programação do botão “Limpar”.



Fonte: Acervo do autor (2022).

PASSO 6: programar o botão “Voltar”.

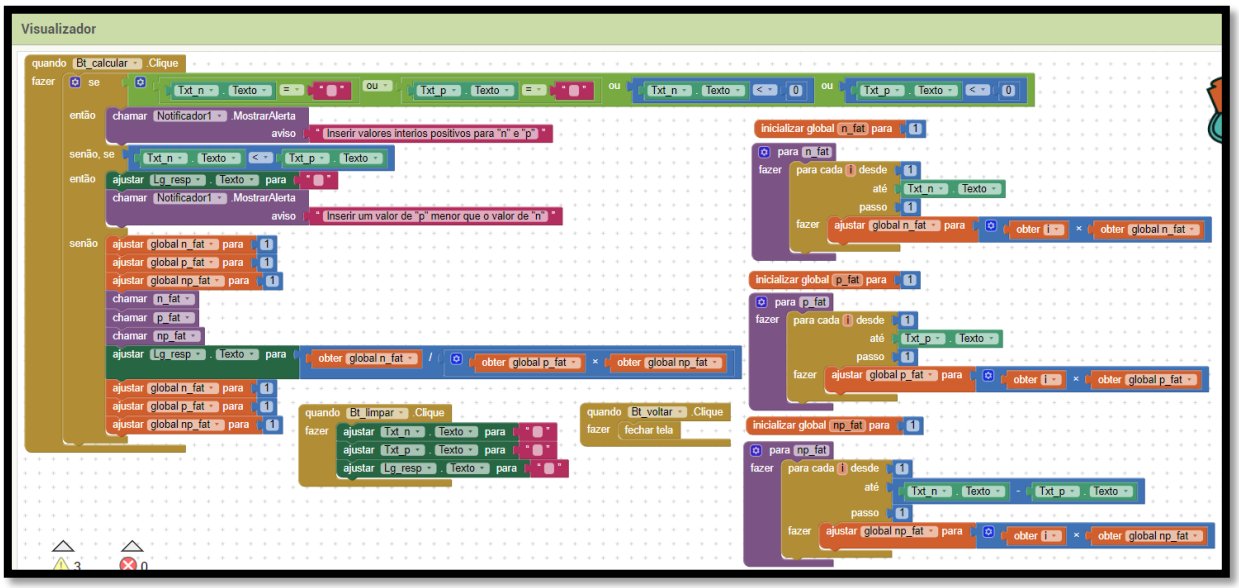
Por fim, deve-se programar o último botão, que será utilizado para retornar ao menu inicial do aplicativo. Mais uma programação idêntica a utilizada na calculadora anterior, como visto na Figura 27.

Figura 27: programação do botão “Voltar”.

Fonte: Acervo do autor (2022).

Dessa forma, conclui-se toda a programação da calculadora de Combinações Simples. Na Figura 28 observa-se a programação completa.

Figura 28: programação completa da Combinação Simples.



Fonte: Acervo do autor (2022).

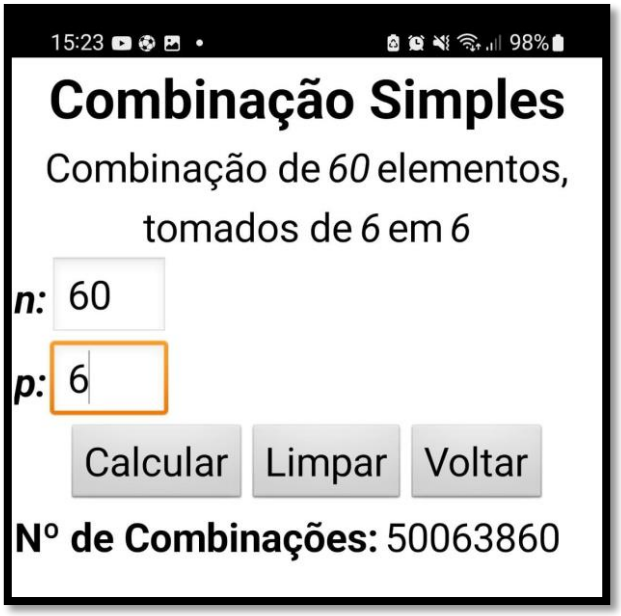
PASSO 7: resolver a situação problema com auxílio da calculadora.

Finalizada a construção da calculadora, e após instalar ela em seu smartphone, o aluno pode retornar à situação problema apresentada anteriormente: “Para apostar na Mega-Sena, faz-se necessário escolher 6 números distintos, dentre os 60 disponíveis para escolha, para assim ter “um jogo” válido para aposta. A partir disso, qual o total de “jogos” possíveis que existem na Mega-Sena?”.

Como já discutido, trata-se de um problema que envolve o cálculo da Combinação Simples (por se tratar do número de subconjuntos com elementos distintos existentes) de 60 elementos (os 60 números disponíveis para escolher) tomados 6 a 6 (a quantidade de números

necessários para se ter um jogo). Na Figura 29 observa-se a tela da calculadora, já em funcionamento em um celular, após a realização do cálculo em questão.

Figura 29: calculadora de Combinações Simples em funcionamento.



Fonte: Acervo do autor (2022).

3.3. Combinação Completa

Segundo Morgado *et al* (2016) a combinação completa de *n* tomados *p* a *p* é o número de modos de escolher *p* objetos distintos ou não, entre *n* objetos distintos dados. Essa combinação se diferencia da combinação simples de *n* tomados *p* a *p* pois esta, de acordo com Morgado *et al* (2016) consiste apenas no número de modos de escolher *p* objetos distintos ou não, entre *n* objetos distintos dados.

A partir disso, estabelece-se uma fórmula para calcular a quantidade de combinações completas a partir do número *n* de elementos escolhidos e do valor *p* de termos para a sequência.

$$CR_{n,p} = \frac{(n+p-1)!}{p!\,(n-1)!} = C_{n+p-1,p}$$

Portanto, a fórmula acima pode ser lida como "O número de combinações completas de *n* elementos, tomados *p* a *p*, é igual ao fatorial de *n + p – 1*, dividido pelo fatorial de *p* multiplicado pelo fatorial de *n – 1*", ou simplesmente "O número de combinações completas de *n* elementos, tomados *p* a *p*, é igual ao número de combinações simples de *n + p – 1* elementos, tomados *p* a *p.*"

Com isso, propõe-se uma situação problema que trabalhe em conjunto o conhecimento do conteúdo de Combinação Completa com o pensamento computacional através do App Inventor 2.

Situação problema: De quantas maneiras distintas podemos comprar 8 sucos em uma loja onde há 5 sabores disponíveis para venda?

Resolução: como não foi estabelecido que os sucos precisam ser de sabores diferentes, então deve-se considerar que eles podem ser distintos ou não, ou seja, faz-se necessário resolver uma Combinação Completa para resolver tal problemática. Dessa forma, será construída uma calculadora de Combinação Completa, com auxílio do App Inventor 2, para resolver esta e qualquer outra situação que envolva o cálculo de Combinações Completa.

Nessa calculadora, o usuário deverá inserir o valor do número total de elementos disponíveis para serem utilizados e o valor da quantidade de termos que ele deseja em seu subconjunto. Para isso, como visto na Figura 30, o design dessa tela deve conter os seguintes componentes: as "legendas" com os textos que podemos observar da tela do smartphone; as "caixas de texto" nas quais o usuário irá inserir os valores de *n* e de *p*; os "botões" que serão utilizados para realizar o cálculo, a limpeza da calculadora e para voltar ao menu inicial; e um "notificador", que não é visível na tela, mas será importante durante a programação.

Figura 30: design da calculadora de Combinação Completa.

Fonte: Acervo do autor (2022).

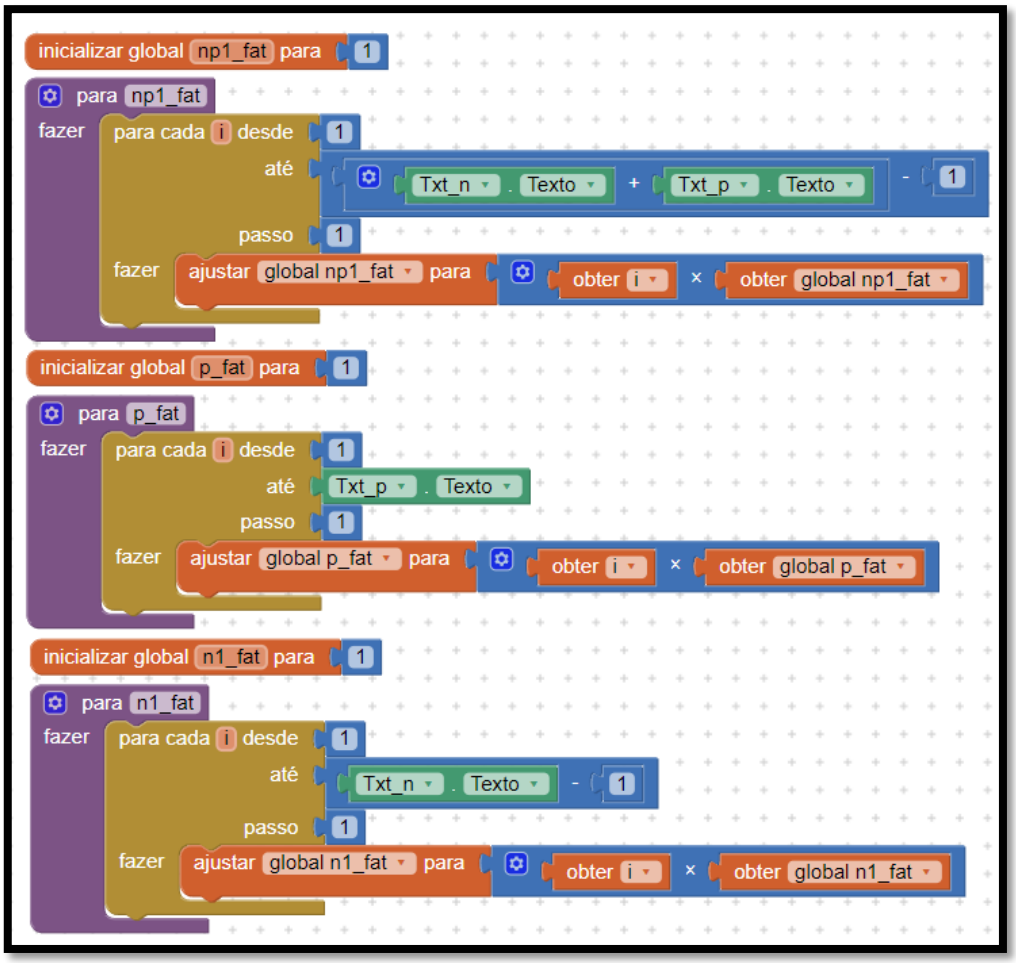
Vale salientar a recomendação de renomear tanto as caixas de textos quanto os botões presentes nos componentes, pois eles serão utilizados durante a programação. Além disso, pode-se observar na Figura 30 que um dos últimos componentes é uma legenda nomeada como "Lg_resp", que corresponde a uma legenda em branco, colocada ao lado da legenda "Nº de Combinações:" que será o local onde aparecerá a resposta desejada após inserir os dados e pressionar o botão "Calcular".

Feito o design da calculadora, inicia-se a programação em blocos, que se encontra dividida nos passos que seguem:

PASSO 1: criar os procedimento para calcular o *(n+p-1)!*, o *p!* e o *(n-1)!*.

A construção desses procedimentos será rigorosamente igual aos procedimentos criados na calculadora de Combinação Simples, como observado na Figura 31.

Figura 31: programação dos procedimentos *(n+p-1)!*, *p!* e *(n-1)!*.

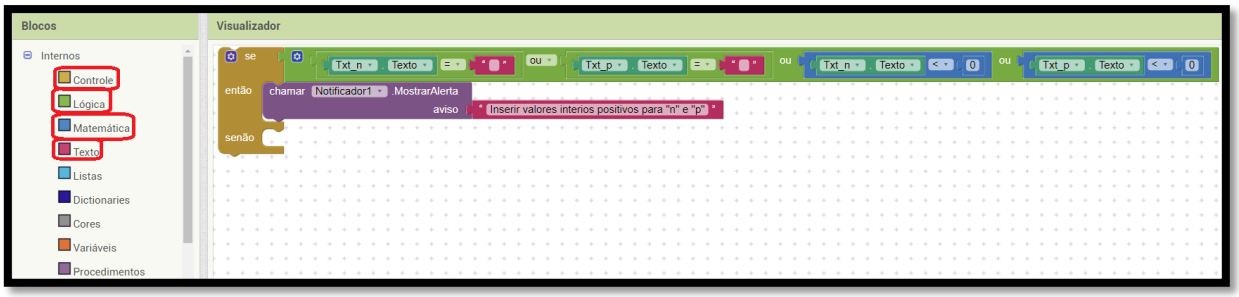


Fonte: Acervo do autor (2022).

PASSO 2: Estabelecer as restrições para os valores de *n* e *p*.

As restrições aqui estabelecidas serão mais simples que aquelas programadas na calculadora anterior, em virtude de a Combinação Completa não apresentar a necessidade de $p \leq n$, então apenas uma parte dos comandos já vistos serão utilizados nessa etapa, como pode ser observado na Figura 32.

Figura 32: programação das restrições de *n* e *p*.

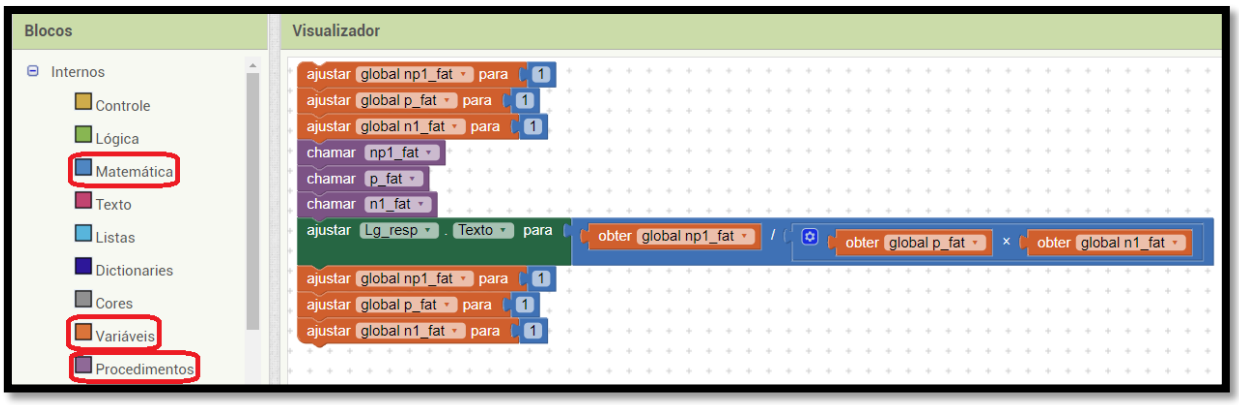


Fonte: Acervo do autor (2022).

PASSO 3: programar o cálculo da Combinação Simples.

Quando o usuário finalmente inserir valores válidos para *n* e *p*, então o programa fará o cálculo da Combinação Completa de *n* elementos, tomados *p* a *p*. Para isso, deve-se realizar a programação observada na Figura 33. Os blocos utilizados consistem em ajustar as "variáveis" sempre para o valor neutro multiplicativo, tanto no início quanto ao final do comando, bem como solicitar que os "procedimentos" criados anteriormente sejam realizados para, enfim, aplicar a fórmula da Combinação Completa.

Figura 33: programação do cálculo da Combinação Completa.



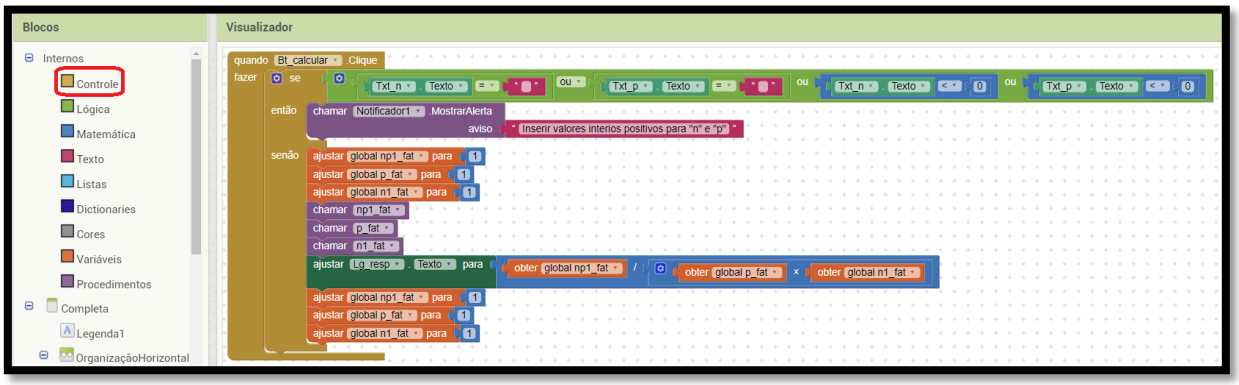
Fonte: Acervo do autor (2022).

Observa-se ainda que será feito o ajuste do texto da "legenda" intitulada "Lg_resp" que, quando colocada na tela de design se encontrava em branco, mas agora assumirá o valor da resposta à conta que é solicitada.

PASSO 4: programar o botão "Calcular".

Finalmente, para que todos os procedimentos listados acima sejam realizados, faz-se necessário programar o botão "Calcular", para que quando ele for pressionado, os comandos estabelecidos sejam realizados. Sua programação pode ser encontrada na Figura 34, e consiste em juntar os blocos "Quando Bt_calcular .Clique fazer", com as programações realizadas nos passos 2 e 3.

Figura 34: programação do botão "Calcular".



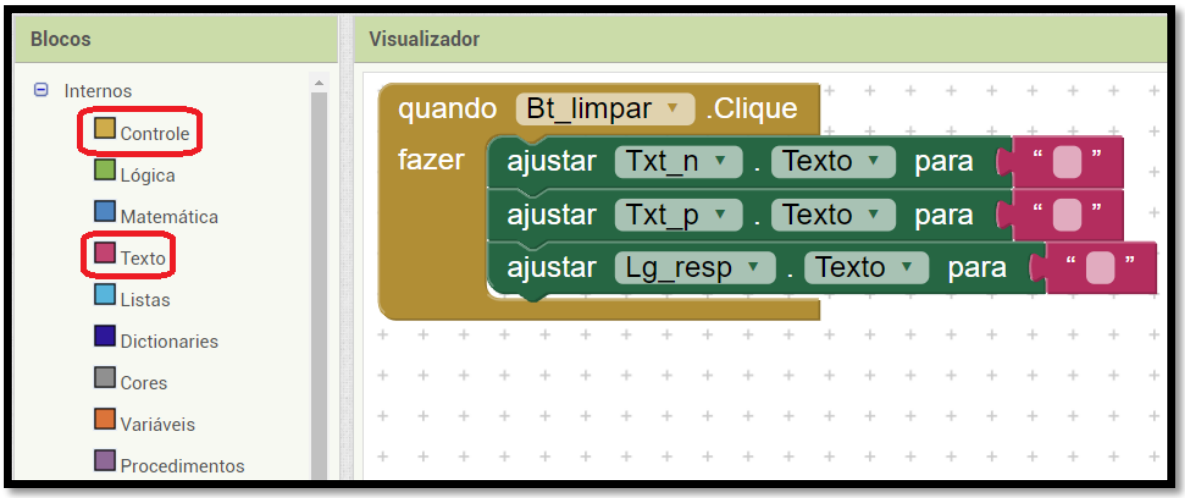
Fonte: Acervo do autor (2022).

Dessa forma, quando o usuário pressionar o botão “Calcular”, o aplicativo primeiramente verificará se os valores podem ser operacionados, de acordo com as restrições já mostradas, para então realizar o cálculo e fornecer o seu valor no lugar da “Lg_resp”.

PASSO 5: programar o botão “Limpar”.

Após programar o primeiro botão, parte-se para os comandos a serem realizados quando pressionado o botão “Limpar”. Essas instruções podem são as mesmas da calculadora de Combinação Simples, como mostrado na Figura 35 e consistem apenas em esvaziar as “caixas de texto” das variáveis *n* e *p*, e da “legenda” na qual a resposta aparece.

Figura 35: programação do botão “Limpar”.

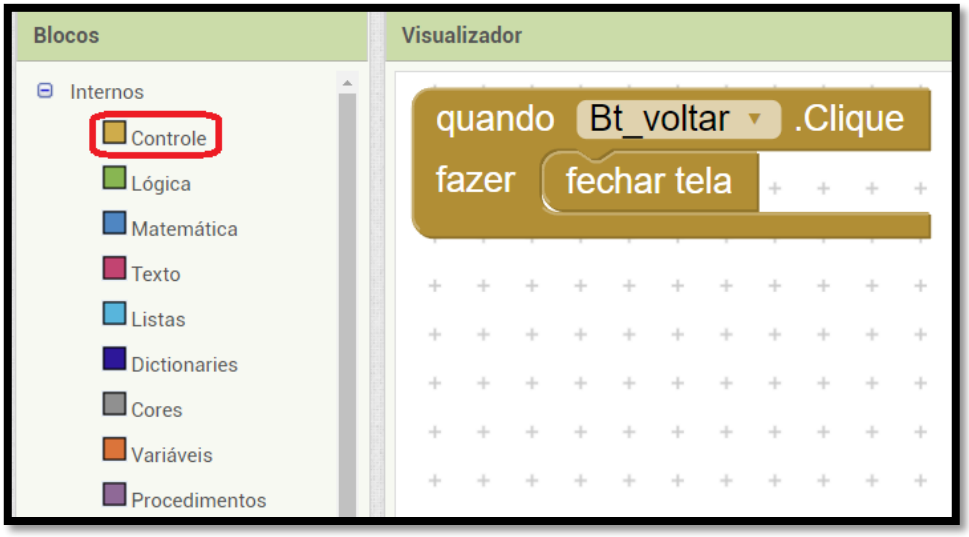


Fonte: Acervo do autor (2022).

PASSO 6: programar o botão “Voltar”.

Por fim, deve-se programar o último botão, que será utilizado para retornar ao menu inicial do aplicativo. Mais uma programação idêntica a utilizada na calculadora anterior, como visto na Figura 36.

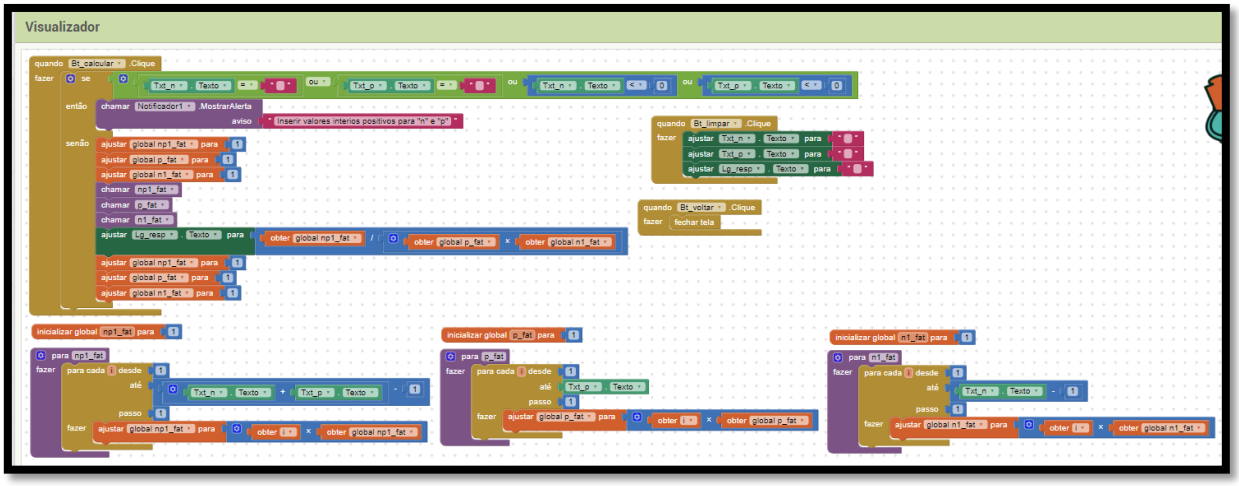
Figura 36: programação do botão “Voltar”.



Fonte: Acervo do autor (2022).

Dessa forma, conclui-se toda a programação da calculadora de Combinações Completa. Na Figura 37 observa-se a programação completa.

Figura 37: programação completa da Combinação Completa.



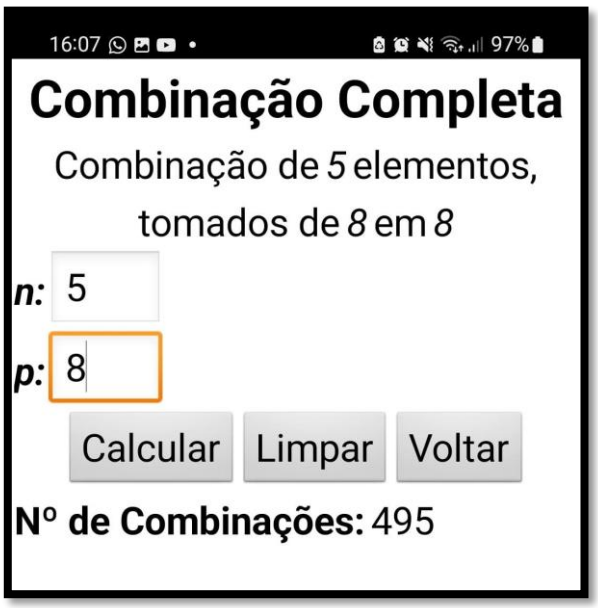
Fonte: Acervo do autor (2022).

PASSO 7: resolver a situação problema com auxílio da calculadora.

Finalizada a construção da calculadora, e após instalar ela em seu smartphone, o aluno pode retornar à situação problema apresentada anteriormente: “De quantas maneiras distintas podemos comprar 8 sucos em uma loja onde há 5 sabores disponíveis para venda?”.

Como já discutido, trata-se de um problema que envolve o cálculo da Combinação Completa (por se tratar do número de modos de escolher *p* objetos distintos ou não, entre *n* objetos distintos dados) de 5 elementos (os 5 sabores de suco disponíveis) tomados 8 a 8 (a quantidade de sucos que se deseja obter). Na Figura 38 observa-se a tela da calculadora, já em funcionamento em um celular, após a realização do cálculo em questão.

Figura 38: calculadora de Combinações Completa em funcionamento.



Fonte: Acervo do autor (2022).

4. CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo apresentar os resultados no nosso estudo, cuja questão de pesquisa foi “É possível trabalhar o pensamento computacional em conjunto com o ensino de Análise Combinatória?”. Para isso realizamos a construção de três calculadoras distintas, por meio do programa App Inventor 2, incentivando o uso dessa metodologia por demais docentes em sala de aula.

Para realizar essa construção, iniciamos apresentando o App Inventor 2, e dando alguns direcionamentos para o seu uso, uma vez que ele não é um software educacional conhecido por todos aqueles que atuam na área da docência. Após apresentar o programa iniciamos a etapa da união dos conceitos por trás do conteúdo matemático com as funcionalidades presentes no aplicativo, que se relacionam diretamente ao pensamento computacional.

Dessa forma, foi feita uma apresentação formal do tópico que seria trabalhado na calculadora a seguir construída a seguir, para então partimos para a construção de seu design e, por fim, realizar a programação em blocos da mesma. Os tópicos trabalhados foram Arranjo Simples, Combinação Simples e Combinação Completa, todos referentes ao conteúdo de Análise Combinatória. Para auxiliar tanto os alunos quanto os professores, foi feito um passo a passo de cada etapa da construção e da programação da calculadora, para que assim todos consigam realizar sua montagem. Por fim, era apresentado uma situação problema em cada um dos tópicos, para que os alunos utilizassem usas próprias calculadoras para resolver o que lhes foi solicitado.

Vale ressaltar que essa metodologia pode se mostrar mais eficiente quando utilizada em turmas nas quais os alunos já tenham tido contato com o conteúdo de Análise Combinatória, e não como uma forma de introduzir esse novo assunto para os estudantes.

Por fim, destacamos que essa metodologia ainda não foi aplicada em pelos autores, mas que projetos futuros estão direcionados para a utilização desse trabalho em sala de aula, para que assim possamos coletar dados do relacionados a potencialização desse método no processo de ensino e aprendizagem de Análise Combinatória, bem como no desenvolvimento do pensamento computacional dos estudantes.

5. REFERÊNCIAS

AZEVEDO, Greiton Toledo de; MALTEMPI, Marcus Vinicius. **Processo de Aprendizagem de Matemática à luz das Metodologias Ativas e do Pensamento Computacional**. Ciência & Educação (Bauru), v. 26, 2020.

BARCELOS, Thiago Schumacher; SILVEIRA, Ismar Frango. **Pensamento computacional e educação matemática: Relações para o ensino de computação na educação básica**. In: XX Workshop sobre Educação em Computação, Curitiba. Anais do XXXII CSBC. 2012. p. 23.

BRASIL. **Base Nacional Comum Curricular: Educação Infantil, Ensino Fundamental e Ensino Médio**. Brasília: MEC/Secretaria de Educação Básica, 2018. Disponível em: <http://basenacionalcomum.mec.gov.br/>.

HAZZAN, Samuel. **Fundamentos de matemática elementar, 5: combinatória, probabilidade**: 43 exercícios resolvidos, 4399 exercícios propostos com resposta, 155 testes de vestibulares com resposta. 7. ed. – São Paulo: Atual, 2004.

MORGADO, Augusto César. CARVALHO, João Bosco Pitombeira de. CARVALHO, Paulo Cezar Pinto. FERNANDEZ, Pedro. **Análise Combinatória e Probabilidade**. – 10 ed. – Rio de Janeiro: SBM, 2016. 338 p. (Coleção Professor de Matemática; 2)

PRIETO, Fernando Posada. **Creando aplicaciones para móviles Android con MIT App Inventor 2.** Ministerio de Educación y Formación Professional. Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF). Recursos Educativos Digitales, Abril 2019. Disponível em: <http://hdl.handle.net/11162/197146>.

CURRÍCULO DOS AUTORES

Lucas Valois Leal de Oliveira

Graduado em Licenciatura em Matemática pela Universidade do Estado do Pará, campus de Ciências Sociais e Educação. Atualmente participante do Programa de Pós-Graduação em Ensino de Matemática da Universidade do Estado do Pará intitulado Mestrado Profissional em Ensino de Matemática.

Cinthia Cunha Maradei Pereira

Possui graduação em Licenciatura em Matemática e em Tecnologia em Processamento de Dados, especialização em Informática Médica, Mestrado em Ciências da Computação e Doutorado em Genética e Biologia Molecular (Bioinformática). Atualmente é Professora da Universidade do Estado do Pará, Docente do Mestrado Profissional em Ensino de Matemática/UEPA e vice-líder do Grupo de Pesquisa em Ensino de Matemática e Tecnologias. Participa do desenvolvimento de tecnologias aplicadas ao ensino de Matemática

Fábio José da Costa Alves

Possui graduação em Licenciatura em Matemática pela União das Escolas Superiores do Pará, Licenciatura em Ciências de 1º Grau pela União das Escolas Superiores do Pará, graduação em Engenharia Civil pela Universidade Federal do Pará. Possui Doutorado e Mestrado em Geofísica pela Universidade Federal do Pará e Pós-Doutorado pelo Programa de Pós-Graduação em Ensino de Ciências e Matemática da Universidade Federal do Rio Grande do Norte. Docente do Mestrado em Educação/UEPA e Docente do Mestrado Profissional em Ensino de Matemática/UEPA. Líder do Grupo de Pesquisa em Ensino de Matemática e Tecnologias. Experiência em desenvolvimento de software educativo para o ensino de matemática.