MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE RONDÔNIA

***CAMPUS* CACOAL**

LIDIOMAR CASTELUBER DA SILVA

SEQUENCECALC:
UM APLICATIVO PARA CÁLCULO DO TERMO GERAL DE SEQUÊNCIAS NUMÉRICAS

Cacoal
2021

LIDIOMAR CASTELUBER DA SILVA

SEQUENCECALC:
UM APLICATIVO PARA CÁLCULO DO TERMO GERAL DE SEQUÊNCIAS NUMÉRICAS

Trabalho de conclusão de curso na modalidade monografia apresentado à Coordenação de Curso de Licenciatura em Matemática do Instituto Federal de Educação, Ciência e Tecnologia de Rondônia – IFRO, *Campus* Cacoal, como requisito para obtenção de aprovação no curso de Licenciatura em Matemática, sob a orientação do Professor Me. Claudemir Miranda Barboza.

Cacoal
2021

FICHA CATALOGRÁFICA

S586s

Silva, Lidiomar Casteluber da.

Sequencecalc : um aplicativo para cálculo de termo geral de sequências numéricas. /Lidiomar Casteluber da Silva. Cacoal, 2021.

53 f.; 30 cm. il.
Inclui bibliografia

Monografia. Licenciatura em Matemática, Instituto Federal de Rondônia – IFRO, Campus Cacoal, 2021.

Orientador: Prof Ms. Claudemir Miranda Barboza.

1. Matemática - ensino 2. Tecnologia educacional 3. Sequência numérica. Interpolação Nominal. I. Lidiomar Casteluber da Silva II. Instituto Federal de Rondônia – IFRO. III. Título.

CDD 512.9434

Bibliotecária responsável: Gizele de Melo Viana – CRB11/914

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE RONDÔNIA

Cacoal - Código INEP: 11109815

Rodovia BR 364, Lote 2A, CEP 76960-970, Cacoal (RO)CNPJ:
10.817.343/0008-73 - Telefone: (69) 2182-9641

ATA DE DEFESA DE MONOGRAFIA

Na data 08/12/2021 realizou-se a sessão pública de defesa da Monografia intitulada **SequenceCalc:Um aplicativo para Cálculo do termo geral de Sequências Numéricas** apresentada pelo aluno **Lidiomar Casteluber da Silva (2016106043024-6)** do Curso **Licenciatura em Matemática (Cacoal)**. Os trabalhos foram iniciados às **19:30** pelo Professor presidente da banca examinadora, constituída pelos seguintes membros:

- **Claudemir Miranda Barboza** (Orientador)
- **Maily Marques Pereira** (Examinadora Interna)
- **Adriana Aparecida Rigolon** (Examinadora Externa)

A banca examinadora, tendo terminado a apresentação do conetúdo da monografia, passou a arguição do candidato. Em seguida, os examinadores reuniram-se para avaliação e deram o parecer final sobre o trabalho apresentado pelo aluno, tendo sido atribúido o seguinte resultado:

[X] APROVADO **Nota: 100**

Proclamados os resultados pelo Presidente da banca examinadora, foram encerrados os trabalhos e, para constar eu, Claudemir Miranda Barboza lavrei a presente Ata que assino juntamente com os demais membros da banca examinadora.

CACOAL/RO, 08/12/21021

Documento assinado eletronicamente por **Lidiomar Casteluber da Silva**, Discente, em 09/12/2021, às 17:45, conforme horário oficial deRondônia, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Claudemir Miranda Barboza**, Presidente, em 09/12/2021, às 17:56, conforme horário oficial deRondônia, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Claudemir Miranda Barboza**, Orientador, em 09/12/2021, às 17:56, conforme horário oficial deRondônia, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Maily Marques Pereira**, Examinador Interno, em 09/12/2021, às 20:16, conforme horário oficial deRondônia, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Adriana Aparecida Rigolon**, Examinador Externo, em 09/12/2021, às 18:46, conforme horáriooficial de Rondônia, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

AGRADECIMENTOS

Agradeço a minha família, pelo apoio e compreensão durante a graduação e desenvolvimento deste trabalho, em especial, a meus pais, Antonio Donizete da Silva e Elizete Castelubre da Silva, por desde sempre me apoiarem incondicionalmente em meus estudos, sendo incentivadores, pacientes e compreensivos.

Agradeço muito também a todos os meus amigos e colegas de curso, pela companhia na caminhada da graduação, pelo apoio e incentivos mútuos. Em especial, lembro aqui dos meus amigos Josias Zeferino dos Reis Junior, Claudiana Vinhatti Parteli, Glaucio Giordanni Moreira Montes e João Paulo Araújo Santos, companheiros de estrada mais próximos na maior parte do curso, estando presentes comigo diariamente na ida e volta ao campus, nos ajudando e compartilhando os muitos momentos felizes e divertidos, mas também os tristes e sérios. Obrigado por aturarem meu gosto musical.

Não posso deixar também de agradecer aos professores do curso que se dedicaram, me incentivaram e instruíram, ensinando verdadeiros valores e não apenas conteúdos matemáticos e pedagógicos. Em especial, gostaria de agradecer aos professores: Jorge da Silva Werneck, cujas aulas me ensinaram muito e tiveram grande contribuição para a ideia de origem desse trabalho; Maily Marques Pereira, por quem tenho grande admiração como professora e matemática e cujas aulas, cheias de teoremas e demonstrações, sempre me encantavam e faziam gostar ainda mais de matemática; Samanta Margarida Milani que, além de professora, foi uma grande amiga e proporcionou a mim e meus colegas de curso algumas das melhores experiências que poderíamos ter em um curso de licenciatura, através dos projetos desenvolvidos; e ao meu orientador, professor e amigo, Claudemir Miranda Barboza, pelas orientações e compartilhamento de ideias, essenciais em diversos momentos do curso e do desenvolvimento deste trabalho, pelo apoio, pela dedicação e principalmente pela paciência.

Por fim – mas não menos importante – quero agradecer muito à minha namorada Priscila Miranda Engelhardt, pelo grande incentivo e motivação durante o desenvolvimento e conclusão desse trabalho, por compartilhar comigo as pequenas e grandes conquistas e por me apoiar nos dias difíceis.

Todos vocês contribuíram para que eu conseguisse alcançar essa conquista e, por isso, gostaria de dizer, “Muito Obrigado!”.

"A história revela que a ciência e a tecnologia se transformam em ritmo acelerado por meio das ideias matemáticas; mesmo as teorias matemáticas que são inicialmente vistas como abstratas e esotéricas se tornam posteriormente indispensáveis para aplicações."

Edward Frenkel

RESUMO

O presente trabalho trata do processo de desenvolvimento, funcionalidade e utilização do aplicativo de smartphones Android, intitulado SequenceCalc. A proposta desse aplicativo é calcular uma expressão polinomial para representar o termo geral de qualquer sequência finita de números racionais, limitando-se, dentro do estabelecido, apenas pela capacidade computacional do aparelho utilizado para executá-lo. Considerando isso, o objetivo desse trabalho é apresentar o SequenceCalc, a lógica matemática envolvida em seu funcionamento e mostrar sua utilidade na resolução de problemas que necessitem da expressão do termo geral de uma sequência numérica ou que auxilie, de alguma forma, na resolução desses problemas. Além disso, constatou-se a falta de outros aplicativos de smartphone com funcionalidade equivalente, sendo esse, um dos fatores que motivaram o interesse em desenvolvê-lo. A fim de mostrar a utilidade do aplicativo, apresenta-se no trabalho a resolução de dois problemas utilizando-o. A partir dessas resoluções, verificou-se que o SequenceCalc traz benefícios para a solução de problemas que, de forma direta, envolvem o termo geral de sequências numéricas e também em outros que possam, apenas em parte, necessitar dessa expressão do termo geral, agilizando um processo que seria normalmente muito demorado se feito manualmente. Ele mostrou-se útil, inclusive, na interação com outros softwares matemáticos, a partir de suas funções de troca de variável e compartilhamento da expressão do termo geral. Apesar do aplicativo já se mostrar funcional, pretende-se, futuramente, adicionar a ele novas funcionalidades como: a identificação de tipos de sequência e de algumas sequências específicas conhecidas, a apresentação de mais de uma expressão de termo geral para dadas sequências, entre outros recursos.

Palavras-chave: SequenceCalc; Sequência Numérica; Aplicativo; Termo Geral;

ABSTRACT

This work deals with the development process, functionality and use of the Android smartphone application, entitled SequenceCalc. The purpose of this application is to calculate a polynomial expression to represent the general term of any finite sequence of rational numbers, limited, within the established, only by the computational capacity of the device used to execute it. Considering this, the objective of this work is to present SequenceCalc, the mathematical logic involved in its operation and to show its usefulness in solving problems that require the expression of the general term of a numerical sequence or that help, in some way, in solving these problems. In addition, there was a lack of other smartphone apps with equivalent functionality, which is one of the factors that motivated the interest in developing it. In order to show the usefulness of the application, the work presents the resolution of two problems using it. From these resolutions, it was found that SequenceCalc brings benefits for solving problems that directly involve the general term of numerical sequences and also in others that may, only partially, need this expression of the general term, streamlining a process that would normally be very time-consuming if done manually. It also proved to be useful in the interaction with other mathematical software, based on its functions of changing the variable and sharing the expression of the general term. Although the application is already functional, it is intended, in the future, to add new features to it, such as: the identification of sequence types and some specific known sequences, the presentation of more than one general term expression for given sequences, among others resources.

Keywords: SequenceCalc; Number Sequences; Application; Keyword 4; General Term.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

Identificar padrões em uma sequência numérica e conseguir expressá-los por meio da linguagem algébrica é uma das habilidades que constam na Base Nacional Comum Curricular Brasileira (BNCC) para a área da matemática logo na etapa do Ensino Fundamental. O desenvolvimento dessa habilidade, apontado como “essencial para utilizar modelos matemáticos na compreensão, representação e análise de relações quantitativas de grandezas e, também, de situações e estruturas matemáticas, fazendo uso de letras e outros símbolos” (BRASIL, 2018, p. 270), possui a finalidade de desenvolver o pensamento algébrico dos estudantes.

No entanto, esse não é o único motivo para o estudo das sequências numéricas. Elas costumam aparecer em diversas situações e a descoberta de um padrão que as define, ou seja, uma relação entre os elementos conhecidos de início, permite determinar outros elementos inicialmente não presentes podendo permitir a previsão de eventos. Além disso, identificar a mesma sequência ou padrão em lugares diferentes pode revelar conexões inesperadas entre áreas distintas.

O parágrafo anterior sugere que um ponto chave no estudo das sequências numéricas é o padrão existente nelas, ou seja, a relação entre seus elementos. Matematicamente, é comum representar esse padrão através de uma função algébrica. Vários autores, como Guidorizzi (2018) e Lima (2006), inclusive, definem sequências de números reais como funções $n \mapsto a_n$[1] com domínio em um subconjunto de $\mathbb{N}$ (conjunto dos números naturais). A notação a_n é usada para indicar o valor que a sequência assume no natural n. Diz-se que a_n é o *termo geral* da sequência. Definições mais aprofundadas sobre sequências numéricas serão exploradas no capítulo 2 deste trabalho. É importante, neste ponto, atentar-se à possibilidade de representar o padrão existente em uma sequência numérica através de uma função algébrica.

Posto isso, é preciso considerar que, muitas vezes, os padrões existentes não são facilmente perceptíveis, podem aparentar não existir ou até são identificados, mas se torna complexo representa-los através de uma função algébrica. Para esses casos, existem técnicas matemáticas que permitem calcular uma função que defina tais sequências. No entanto, essas técnicas podem exigir cálculos muito demorados e difíceis caso sejam feitos manualmente e, embora seja importante que técnicas com este propósito sejam estudadas e praticadas dessa

---

[1] O símbolo $\mapsto$ sigifica “mapeia para” e indica os elementos de entrada e saída de uma função.

forma, para domínio ou fixação do conhecimento, após essa etapa de prática, o cálculo manual se torna apenas maçante e demorado, especialmente nos casos em que desejamos apenas o resultado desses cálculos para utilizá-los na solução de problemas.

Em situações como essas, a utilização de computadores se mostra uma excelente opção. Computadores são conhecidos por serem ótimos na realização de cálculos matemáticos de forma rápida e precisa. Para realizar esses cálculos, os computadores fazem uso de programas (ou aplicativos), que são as instruções a serem realizadas por eles a partir de dados iniciais para que sejam produzidos os resultados desejados. Programas que facilitam o cálculo ou permitem a manipulação de expressões algébricas, como as expressões que definem o termo geral de sequências numéricas, são denominados Sistemas de Álgebra Computacional ou CAS (do termo em inglês "Computer Algebra System"). Consolidando as afirmações anteriores, sobre a eficácia dos computadores no auxílio e realização de cálculos complexos, von zur Gathen e Gerhard afirmam que:

> "Os sistemas de álgebra computacional têm uma ampla variedade de aplicações em campos que exigem cálculos tediosos, demorados e difíceis de acertar manualmente. Na física, os sistemas de álgebra computacionais são usados em física de alta energia, para eletrodinâmica quântica, cromodinâmica quântica, órbita de satélite e cálculos de trajetória de foguetes e mecânica celeste em geral. Como exemplo, Delaunay calculou a órbita da lua sob a influência do sol e uma terra não esférica com uma eclíptica inclinada. Este trabalho levou vinte anos para ser concluído e foi publicado em 1867. Foi mostrado, em 20 horas em um pequeno computador em 1970, que estava correto com nove casas decimais."[2] (2013, p. 20, tradução nossa).

Diante disso, decidiu-se desenvolver um programa computacional com o propósito específico de calcular e apresentar o termo geral de qualquer sequência numérica finita de números racionais. E diante das escolhas possíveis de plataforma[3] para execução deste programa, deu-se preferência pelos smartphones com o sistema operacional Android, da Google. A escolha pelos smartphones motiva-se pela atual grande presença deles no cotidiano das pessoas e pelo sistema operacional Android, por ele estar presente em aproximadamente 86% dos smartphones utilizados no Brasil no período de 2019 a 2021 (STATCOUNTER, 2021). Salientando a presença dos smartphones no cotidiano das pessoas, Guidini afirma que:

---

[2] No original: Computer algebra systems have a wide variety of applications in fields that require computations that are tedious, lengthy and difficult to get right when done by hand. In physics, computer algebra systems are used in high energy physics, for quantum electrodynamics, quantum chromodynamics, satellite orbit and rocket trajectory computations and celestial mechanics in general. As an example, Delaunay calculated the orbit of the moon under the influence of the sun and a nonspherical earth with a tilted ecliptic. This work took twenty years to complete and was published in 1867. It was shown, in 20 hours on a small computer in 1970, to be correct to nine decimal places.

[3] Combinação do hardware de um computador (parte física) e um sistema operacional (parte lógica que permite a execução de programas em um computador).

> Os smartphones são mais íntimos do que muitos familiares. Colados ao corpo do indivíduo, os telefones inteligentes concentram as atividades do dia a dia, os contatos e as comunicações. A quantidade e a qualidade da informação que esses aparelhos retêm fazem deles um acessório pessoal de total intimidade e dependência. Se desvincular dessa tecnologia parece inimaginável. (2017, p. 42)

Outro fator relevante para essa escolha de plataforma foi a relativa falta de apps (aplicativos móveis) com funcionalidade equivalente à do proposto neste trabalho. Ao buscar por apps relacionados a sequências numéricas ou solucionadores de sequências na Play Store[4], loja de aplicativos da Google para aparelhos Android, três dos apps apresentados se propõe a apresentar o termo geral de uma dada sequência finita de números racionais, porém, testes razoáveis foram realizados com os três e nenhum deles conseguiu cumprir com este propósito para todas as sequências informadas.

O trabalho propõe então, o desenvolvimento de um aplicativo para suprir essas deficiências, de forma a calcular e apresentar o termo geral para qualquer sequência finita de números racionais, limitando-se apenas à capacidade computacional do aparelho utilizado para executá-lo. Considerando isso, o objetivo desse trabalho é apresentar este aplicativo, nomeado SequenceCalc, como uma ferramenta para este propósito, a fim de facilitar a resolução de problemas que necessitem da expressão do termo geral de uma sequência numérica ou que, de alguma forma, essa expressão auxilie na resolução desses problemas.

A ideia de desenvolvimento deste app surgiu a partir da necessidade de se conhecer o termo geral de algumas sequências numéricas em situações específicas e recorrentes, onde se podia ou precisava utilizá-lo para verificar ou chegar à solução de problemas. Um algoritmo para isso já havia sido produzido em computador utilizando o software matemático Maple, mas haviam algumas desvantagens em seu uso, tais quais, a indisponibilidade de acesso a um computador com o algoritmo sempre que se desejasse usá-lo e a pouca praticidade em sua manipulação. Esses fatores foram outros que incentivaram o desenvolvimento do SequenceCalc.

Quanto à estrutura deste trabalho: no capítulo 2 são abordadas as ideias e conceitos matemáticos relacionados ao aplicativo; no capítulo 3 apresenta-se o SequenceCalc, as ferramentas utilizadas no seu processo de desenvolvimento e sua interface e comportamento; no capítulo 4 são apresentados dois problemas matemáticos e soluções para eles utilizando o SequenceCalc; no capítulo 5 expõe-se algumas considerações finais sobre o trabalho.

---

[4] Acessível através do site https://play.google.com/store/apps ou do app Play Sore, nativo do sistema Android.

2 SEQUÊNCIAS NUMÉRICAS

Lima, (2006) define uma sequência de números reais como uma função $f : \mathbb{N} \to \mathbb{R}$, que associa a cada número natural *n* um número real a_n, chamado *o* ***n*-ésimo termo** da sequência.

Iezzi e Hazzan (2013, p. 1) classificam as sequências numéricas entre finitas e infinitas. De acordo com os autores, "chama-se **sequência finita** ou **ênupla** toda aplicação *f* do conjunto $\mathbb{N}_n^* = \{1,2,3,\dots,n\}$ em $\mathbb{R}$ [...] [em que] cada número natural i $(1 \leq i \leq n)$ está associado um número real a_i" e "chama-se **sequência infinita** toda aplicação f de $\mathbb{N}^*$ em $\mathbb{R}$ [...] [em que] cada $i \in \mathbb{N}^*$ está associado um $a_i \in \mathbb{R}$".

Pode-se representar uma sequência *f* anotando os pares ordenados que associam cada natural *i* à sua imagem a_i, dessa maneira: $f = \{(1,a_1),(2,a_2),(3,a_3),\dots,(i,a_i),\dots\}$; ou ainda, anotando apenas a imagem de *f*, do seguinte modo: $f = (a_1,a_2,a_3,\dots,a_i,\dots)$. Em ambas as representações, a ordem dos elementos é indicada pelo índice $i \in \mathbb{N}$ (da esquerda para a direita, do menor ao maior índice).

2.1 Termo geral de uma Sequência Numérica

Normalmente, interessam à Matemática, as sequências em que os termos se sucedem obedecendo a certa regra, isto é, aquelas que têm uma lei de formação. Iezzi e Hazzan (2013), afirmam que esta lei de formação pode ser apresentada de três maneiras: por fórmula de recorrência; expressando cada termo em função de sua posição e; por propriedade dos termos.

As sequências numéricas apresentadas por fórmula de recorrência são definidas por duas regras: uma para identificar o primeiro termo (a_1) e outra para calcular cada termo (a_n) a partir do antecedente (a_{n-1}). Seja, por exemplo, uma sequência finita *f* em que $a_1 = 2$ e $a_n = a_{n-1} + 3, \forall\, n \in \{2,3,4,5,6\}$, temos:

$n = 2 \Rightarrow a_2 = a_1 + 3 = 2 + 3 = 5$

$n = 3 \Rightarrow a_3 = a_2 + 3 = 5 + 3 = 8$

$n = 4 \Rightarrow a_4 = a_3 + 3 = 8 + 3 = 11$

$n = 5 \Rightarrow a_5 = a_4 + 3 = 11 + 3 = 14$

$n = 6 \Rightarrow a_6 = a_5 + 3 = 14 + 3 = 17$

Dessa forma, $f = (2, 5, 8, 11, 14, 17)$.

Quando se apresenta uma sequência expressando cada termo em função de sua posição, é dada uma fórmula que expresse a_n em função de *n*. Por exemplo, sendo g uma sequência finita cujos termos obedecem à lei $a_n = 2^n, n \in \{1,2,3,4\}$, temos:

$a_1 = 2^1 = 2, a_2 = 2^2 = 4, a_3 = 2^3 = 8$ e $a_4 = 2^4 = 16$

Assim, $a = (2, 4, 8, 16)$.

Por fim, quando uma sequência é definida a partir de uma propriedade de seus termos, é dada uma propriedade que os termos da sequência devem apresentar. Uma sequência ordenada de números primos, por exemplo, possui este tipo de lei de formação. Seja *p* a sequência infinita formada pelos números primos positivos colocados em ordem crescente, temos que $p = (2, 3, 5, 7, 11, \dots)$, neste caso, com os cinco primeiros termos sendo mostrados.

Notemos que essa sequência não pode ser dada por uma fórmula de recorrência, bem como não existe uma fórmula para calcular o *n*-ésimo número primo positivo a partir de *n*, considerando a sequência completa, com todos os seus infinitos termos.

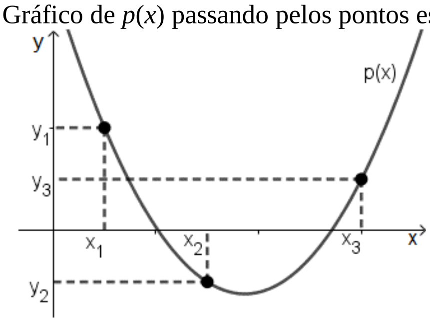
No entanto, caso consideremos uma parte finita desta sequência numérica (ou de qualquer outra), é possível encontrar uma lei de formação – ou **termo geral** – para a mesma. E isto pode ser realizado através de um processo matemático denominado **interpolação polinomial**.

2.2 Interpolação Polinomial

As definições e conceitos apresentados neste tópico são baseados em Barroso e Barroso *et al*. (1987) e Freitas, Corrêa e Vaz (2019).

Consideremos o problema de determinar um polinômio quadrático *p*(*x*) cujo gráfico passa pelos pontos (x_1, y_1), (x_2, y_2) e (x_3, y_3), sendo x_1, x_2 e x_3 todos diferentes entre si (Figura 1).

Figura 1 – Gráfico de *p*(*x*) passando pelos pontos especificados



Fonte: O autor.

Um polinômio quadrático é escrito, em sua forma geral, como $p(x) = ax^2 + bx + c, a \neq 0$ e $a, b, c \in \mathbb{R}$. Como o objetivo nesse problema é determinar o polinômio, as incógnitas cujos valores devem ser encontrados são os três coeficientes *a*, *b* e *c* do polinômio na forma geral. A partir dos três pontos pelos quais o gráfico do polinômio deve passar, podemos definir três equações $p(x_1) = y_1, p(x_2) = y_2$ e $p(x_3) = y_3$.

Explicitando essas equações, teremos:

$$ax_1^2 + bx_1 + c = y_1$$
$$ax_2^2 + bx_2 + c = y_2$$
$$ax_3^2 + bx_3 + c = y_3$$

Reescrevendo essas equações, fica evidente o caráter de sistema linear do problema:

$$x_1^2 \cdot a + x_1 \cdot b + c = y_1$$
$$x_2^2 \cdot a + x_2 \cdot b + c = y_2$$
$$x_3^2 \cdot a + x_3 \cdot b + c = y_3$$

Problemas desse tipo podem ser generalizados para um número qualquer n de pontos. Ou seja: dados os pares (x_1, y_1), …, (x_n, y_n), com os x_i's distintos dois a dois, podemos encontrar um polinômio $p(x)$ cujo gráfico passe pelos pontos dados, de forma que: $p(x_1) = y_1, p(x_2) = y_2, \dots, p(x_n) = y_n$. E essa é justamente a definição de Interpolação Polinomial: o processo de encontrar um polinômio $p(x)$ cujo gráfico passe por um dado conjunto de pontos.

Uma das formas de encontrar esse polinômio é através da solução de um sistema linear como apresentado anteriormente, de modo que, com a solução deste sistema, se obterão os coeficientes de $p(x)$. Nesse contexto, se procurarmos por um polinômio de grau k teremos $k + 1$ coeficientes a determinar. Para n pontos dados, temos que satisfazer n equações e, dessa forma, convém fixar $k = n - 1$. Assim, temos o sistema de n equações a seguir, onde os coeficientes são as n incógnitas a_k, $k \in [0, n]$:

$$\begin{array}{ccccccccccc}
a_0 & + & x_1 \cdot a_1 & + & x_1^2 \cdot a_2 & + & \cdots & + & x_1^{n-1} \cdot a_{n-1} & = & y_1 \\
a_0 & + & x_2 \cdot a_1 & + & x_2^2 \cdot a_2 & + & \cdots & + & x_2^{n-1} \cdot a_{n-1} & = & y_2 \\
\vdots & & \vdots & & \vdots & & \ddots & & \vdots & = & \vdots \\
a_0 & + & x_n \cdot a_1 & + & x_n^2 \cdot a_2 & + & \cdots & + & x_n^{n-1} \cdot a_{n-1} & = & y_n
\end{array}$$

Pode-se também representar este sistema na forma de um produto linear de matrizes, da seguinte maneira:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ 1 & x_3 & x_3^2 & \cdots & x_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}$$

Em uma sequência numérica qualquer com n elementos ($n \in \mathbb{N}$) podemos associar, a cada um de seus elementos, um par ordenado $(i, y_i), i \in [1, n]$ de forma que y_i corresponda ao valor numérico do elemento na sequência e i corresponda à sua posição. Fazendo isto para todos os elementos da sequência podemos interpolar os pares ordenados obtidos e obter o seguinte sistema linear:

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2^2 & \cdots & 2^{\square-1} \\ 1 & 3 & 3^2 & \cdots & 3^{\square-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \square & \square^2 & \cdots & \square^{\square-1} \end{bmatrix} \cdot \begin{bmatrix} \square_0 \\ \square_1 \\ \square_2 \\ \vdots \\ \square_{\square-1} \end{bmatrix} = \begin{bmatrix} \square_1 \\ \square_2 \\ \square_3 \\ \vdots \\ \square_{\square} \end{bmatrix}$$

em que os termos $\square_{\square}, \square \in [0, \square - 1]$ serão as incógnitas do sistema, coeficientes da lei de formação da sequência em questão.

Desta forma, para obter a lei de formação de uma sequência numérica qualquer, basta que solucionemos este sistema de equações e escrevamos o resultado na forma de um polinômio.

Vejamos um exemplo: Dada a sequência numérica $\square = (2, 3, 9, 16, 25)$, vamos obter uma possível lei de formação através do método de interpolação polinomial.

Solução:

Descrevamos o sistema linear associado à sequência. Como a sequência possui 5 elementos, o sistema linear associado possuirá 5 linhas e 5 incógnitas $(\square_0, \square_1, \square_2, \square_3, \square_4)$ e sua lei de formação será um polinômio do 4º grau.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2^2 & 2^3 & 2^4 \\ 1 & 3 & 3^2 & 3^3 & 3^4 \\ 1 & 4 & 4^2 & 4^3 & 4^4 \\ 1 & 5 & 5^2 & 5^3 & 5^4 \end{bmatrix} \cdot \begin{bmatrix} \square_0 \\ \square_1 \\ \square_2 \\ \square_3 \\ \square_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 9 \\ 16 \\ 25 \end{bmatrix}$$

Para resolver este sistema linear através do método da eliminação Gauss (escalonamento), toma-se a seguinte matriz a ser escalonada:

$$\begin{array}{c} \square_1 \\ \square_2 \\ \square_3 \\ \square_4 \\ \square_5 \end{array} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 \\ 1 & 2 & 2^2 & 2^3 & 2^4 & 3 \\ 1 & 3 & 3^2 & 3^3 & 3^4 & 9 \\ 1 & 4 & 4^2 & 4^3 & 4^4 & 16 \\ 1 & 5 & 5^2 & 5^3 & 5^4 & 25 \end{bmatrix}$$

Resolvendo as potências e escalonando:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 \\ 1 & 2 & 4 & 8 & 16 & 3 \\ 1 & 3 & 9 & 27 & 81 & 9 \\ 1 & 4 & 16 & 64 & 256 & 16 \\ 1 & 5 & 25 & 125 & 625 & 25 \end{bmatrix} \begin{array}{c} \longrightarrow \\ \square_2 - \square_1 \\ \square_3 - \square_1 \\ \square_4 - \square_1 \\ \square_5 - \square_1 \end{array} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 \\ 0 & 1 & 3 & 7 & 15 & 1 \\ 0 & 2 & 8 & 26 & 80 & 7 \\ 0 & 3 & 15 & 63 & 255 & 14 \\ 0 & 4 & 24 & 124 & 624 & 23 \end{bmatrix} \begin{array}{c} \longrightarrow \\ \longrightarrow \\ \square_3 - 2\square_2 \\ \square_4 - 3\square_2 \\ \square_5 - 4\square_2 \end{array}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 \\ 0 & 1 & 3 & 7 & 15 & 1 \\ 0 & 2 & 8 & 26 & 80 & 7 \\ 0 & 3 & 15 & 63 & 255 & 14 \\ 0 & 4 & 24 & 124 & 624 & 23 \end{bmatrix} \begin{array}{c} \longrightarrow \\ \square_2 - \square_1 \\ \square_3 - \square_1 \\ \square_4 - \square_1 \\ \square_5 - \square_1 \end{array} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 \\ 0 & 1 & 3 & 7 & 15 & 1 \\ 0 & 2 & 8 & 26 & 80 & 7 \\ 0 & 3 & 15 & 63 & 255 & 14 \\ 0 & 4 & 24 & 124 & 624 & 23 \end{bmatrix} \begin{array}{c} \longrightarrow \\ \longrightarrow \\ \square_3 - 2\square_2 \\ \square_4 - 3\square_2 \\ \square_5 - 4\square_2 \end{array}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 2 \\ 0 & 1 & 2 & 3 & 4 & 1 \\ 0 & 0 & 2 & 6 & 12 & 4 \\ 0 & 0 & 0 & 6 & 24 & -17 \\ 0 & 0 & 0 & 0 & 57 & 78 \end{bmatrix}$$

Terminado o processo de escalonamento, podemos solucionar o sistema. Teremos:

$$57\square_4 = 78 \rightarrow \square_4 = \frac{78}{57} = \frac{26}{19}$$

$$6\square_3 + 24\square_4 = -17 \rightarrow 6\square_3 + 24\left(\frac{26}{19}\right) = -17 \rightarrow 6\square_3 = -17 - \frac{624}{19} \rightarrow \square_3 = -\frac{947}{114}$$

Fazendo $\square_3 - \square_4$, teremos:

$$2\square_2 - 12\square_4 = 21 \rightarrow 2\square_2 - 12\left(\frac{26}{19}\right) = 21 \rightarrow 2\square_2 = 21 + \frac{312}{19} \rightarrow \square_2 = \frac{711}{38}$$

Fazendo $\square_2 - \square_3$, teremos:

$$\square_1 - 3\square_3 - 8\square_4 = -3 \rightarrow \square_1 - 3\left(-\frac{947}{114}\right) - 8\left(\frac{26}{19}\right) = -3 \rightarrow$$

$$\square_1 = -3 - \frac{947}{38} + \frac{208}{19} \rightarrow \square_1 = -3 - \frac{531}{38} \rightarrow \square_1 = -\frac{645}{38}$$

Fazendo $\square_1 - \square_2$, teremos:

$$\square_0 - \square_2 - 2\square_3 - 3\square_4 = 1 \rightarrow \square_0 - \frac{711}{38} - 2\left(-\frac{947}{114}\right) - 3\left(\frac{26}{19}\right) = 1 \rightarrow$$

$$\square_0 = 1 + \frac{711}{38} - \frac{947}{57} + \frac{78}{19} \rightarrow \square_0 = \frac{114 + 2133 - 1894 + 468}{114} \rightarrow \square_0 = \frac{821}{114}$$

Dessa forma, uma possível lei de formação para a sequência $\square = (2, 3, 9, 16, 25)$ será o seguinte polinômio do 4º grau:

$$\square_{\square} = \square_0 + \square_1\square + \square_2\square^2 + \square_3\square^3 + \square_4\square^4$$

Organizando os expoentes do maior para o menor e substituindo os valores dos coeficientes, teremos:

$$\square_{\square} = \frac{26}{19}\square^4 - \frac{947}{114}\square^3 + \frac{711}{38}\square^2 - \frac{645}{38}\square + \frac{821}{114}$$

sendo esta a lei de formação obtida pelo método de escalonamento para a sequência dada inicialmente.

É perceptível que a aplicação deste método envolve cálculos extensos quando feito manualmente, principalmente para sequências com uma grande quantidade de termos. No entanto, é importante observar que ele pode ser generalizado para qualquer outra sequência finita de números racionais, utilizando apenas os termos da própria sequência como dados necessários para aplicar a generalização. Isso é importante pois permite que se crie um algoritmo (uma espécie de “receita”) de como obter o termo geral de uma dada sequência a partir dos termos dela. Esse algoritmo pode ser definido, resumidamente, da seguinte maneira:

1. Considera-se uma sequência finita de números racionais com $\square$ termos;
2. Cria-se a matriz $\square_{\square\times\square}$ em que cada elemento $\square_{\square\square}$ dessa matriz é dado por $\square^{\square}$, sendo $\square$ o índice numérico que indica a linha da matriz a qual pertence o elemento, e $\square$ o índice que indica a coluna;
3. Cria-se a matriz $\square$ de $\square$ linhas e 1 coluna cujo elemento da $\square$-ésima linha será ocupado pelo $\square$-ésimo termo da sequência considerada.
4. Encontra-se a matriz $\square$ a partir da solução do sistema $\square \times \square = \square$. A matriz $\square$ irá conter os coeficientes do termo geral da sequência considerada inicialmente.
5. Apresenta-se o termo geral $\square_{\square}$ da sequência inicial na forma da seguinte soma:

$$\square_{\square} = \sum_{\square=1}^{\square} \square_{\square} \cdot \square^{\square}$$

em que $\square_{\square}$ representa o elemento da $\square$-ésima linha da matriz $\square$.

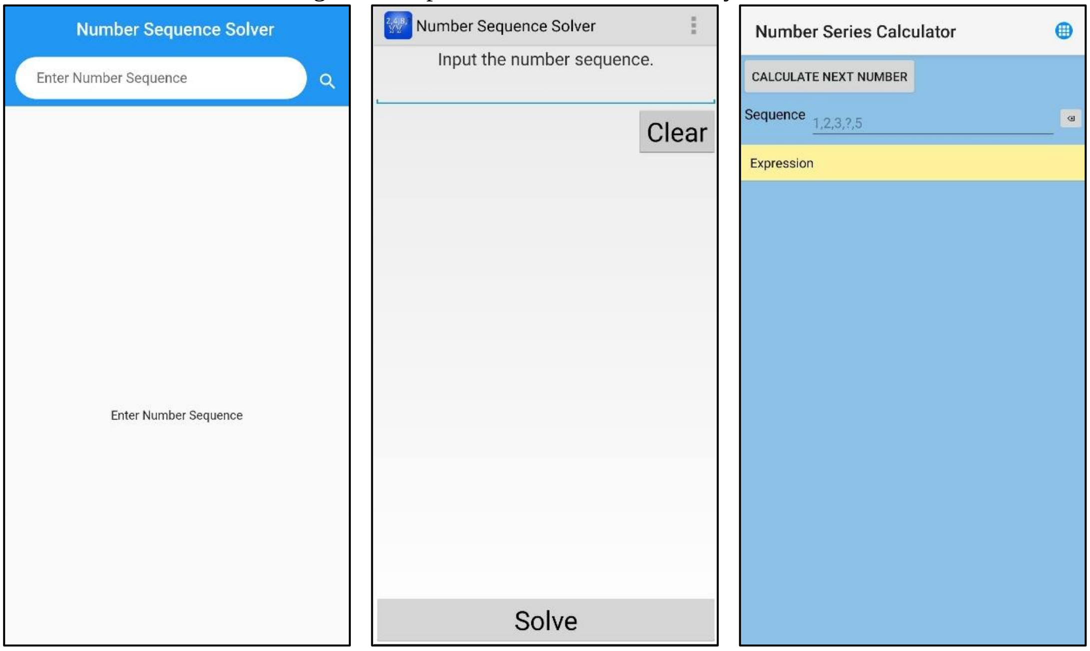
A possibilidade de generalização desse algoritmo permite que ele possa ser codificado, através de uma linguagem de programação, em instruções a serem seguidas pelo computador. E essas instruções serão as que vão permitir ao SequenceCalc obter e apresentar o termo geral da sequência informada a ele.

O próximo capítulo apresentará as ferramentas utilizadas para o desenvolvimento do SequenceCalc, além de sua interface e comportamento, mostrando como esse algoritmo funciona na prática em sua implementação.

3 O APLICATIVO SEQUENCECALC

Uma das motivações que levaram ao desenvolvimento do SequenceCalc foi o levantamento realizado na Play Store com os descritores de busca “termo geral de uma sequência” e “calculador de sequências”, resultando em três aplicativos com a mesma temática do SequenceCalc: dois deles, nomeados *Number Sequence Solver*[5] e outro intitulado *Number Series Calculator*[6] (Figura 2).

Figura 2 – Aplicativos levantados na Play Store



Fonte: Captura de tela dos aplicativos.

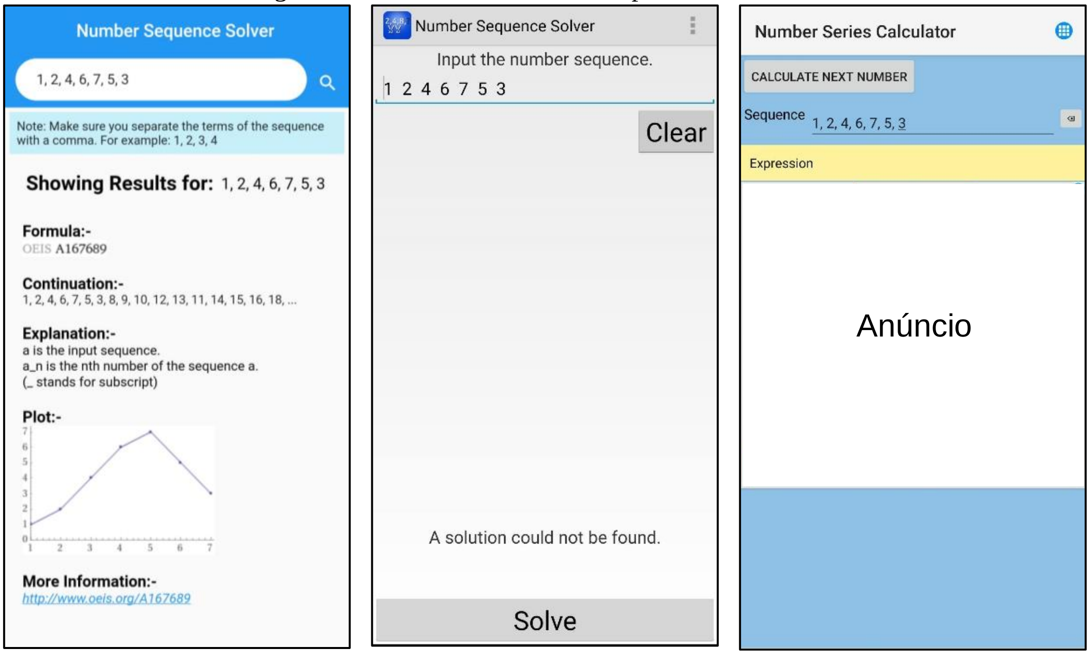
Testou-se os três aplicativos informando algumas sequências numéricas e constatou-se que nenhum dos três conseguia calcular uma expressão para o termo geral de algumas delas – por exemplo, a sequência (1, 2, 4, 6, 7, 5, 3) (Figura 3).

---

[5] Downloads disponíveis em: https://play.google.com/store/apps/details?id=com.sequence / https://play.google.com/store/apps/details?id=com.shauryamehta1804.pattern_solver

[6] Download disponível em: https://play.google.com/store/apps/details?id=com.expressionfinder

Figura 3 – Teste realizado com os aplicativos similares



Fonte: Capturas de tela dos aplicativos.

O aplicativo *Number Sequence Solver* – primeiro, da esquerda para a direita na Figura 3 – conseguiu identificar um padrão na sequência informada (números naturais com primos gêmeos trocados e não primos adjacentes), mas não apresentou uma expressão de termo geral para ela. Os outros dois aplicativos, além de não identificarem um padrão, também não conseguiram calcular uma expressão para o termo geral da sequência informada. Considerando isso, o SequenceCalc foi planejado e desenvolvido para suprir essas limitações, de forma a calcular e apresentar uma expressão de termo geral para qualquer sequência finita de números racionais (computacionalmente calculáveis). É importante destacar que, a expressão calculada pelo SequenceCalc não será a única e nem sempre considerada a mais simples possível para representar determinada sequência. O principal propósito dele, no entanto, é conseguir apresentar uma expressão para qualquer caso. Na próxima seção serão apresentados o seu processo de desenvolvimento, as ferramentas utilizadas nesse processo e um pouco da lógica de programação.

3.1 Desenvolvimento do aplicativo

O SequenceCalc foi projetado para funcionar em smartphones com o sistema operacional Android e programado a partir do software Android Studio para Windows utilizando a linguagem de programação Java.

O software Android Studio é tido como o “ambiente de desenvolvimento integrado [...] oficial para o desenvolvimento de apps Android” (ANDROID DEVELOPERS, 2021). A escolha dele como plataforma de desenvolvimento para o SequenceCalc se deu por ele atender a todas as necessidades do projeto e por interesse e possibilidades do autor deste trabalho.

A linguagem de programação Java é uma das linguagens oficiais para o desenvolvimento de aplicativos Android, em conjunto com as linguagens Kotlin e C++. A escolha por utilizá-la para programar o SequenceCalc se deu por ela ser uma das disponíveis nativamente no Android Studio e por interesse do autor.

No Android Studio foram realizadas tanto a codificação do algoritmo para o cálculo e apresentação do termo geral das sequências numéricas (utilizando a linguagem Java), quanto a criação da interface do SequenceCalc, que são os elementos visuais dele, que aparecem na tela do smartphone.

Inicialmente preparou-se uma interface básica para o aplicativo, de forma a contar apenas com um campo de entrada para digitação da sequência, um botão para acionar o cálculo do termo geral e um espaço textual para apresenta-lo após calculado. Posteriormente, outros elementos foram adicionados para funcionalidade e informações adicionais. Partindo para a parte de implementação do algoritmo de cálculo do termo geral, percebeu-se que alguns complementos se fariam necessários, neste caso, algumas bibliotecas de código.

Certas linguagens de programação, incluindo a linguagem Java, possibilitam aos programadores o uso de conjuntos de código prontos chamados bibliotecas de código. Normalmente, em cada biblioteca são implementadas várias funções com propósitos relacionados uns com os outros. Essas funções são partes do código de uma biblioteca que permitem realizar alguma operação (por exemplo, validar dados em um formulário ou apresentar a raiz quadrada de um número). Algumas bibliotecas são nativas da linguagem (internas) enquanto outras são desenvolvidas pela comunidade de programadores e compartilhada para ser utilizada por outros (externas). Para facilitar a implementação tanto do algoritmo de cálculo quanto da apresentação do termo geral das sequências numéricas no SequenceCalc, utilizou-se de duas bibliotecas externas de código Java: a biblioteca *Jama*[7],

---

[7] Página do projeto: https://math.nist.gov/javanumerics/jama/

para manipulação e cálculo de matrizes e; a biblioteca *JLatexMath*[8] para apresentação do termo geral no formato *LaTeX*[9].

Como mostrado ao final do capítulo 2, o algoritmo utilizado para calcular o termo geral de uma dada sequência numérica de $\square$ elementos envolve a criação de duas matrizes: uma matriz quadrada $\square_{\square\times\square}$ cujos elementos são potências da forma $\square^{\square}$ (sendo $\square$ o índice indicativo das linhas da matriz e $\square$ o índice indicativo das colunas) e uma matriz coluna $\square$ composta pelos $\square$ termos da sequência. Uma terceira matriz $\square$, dos coeficientes do termo geral da sequência inicial, é também obtida a partir da solução do sistema $\square \times \square = \square$. A biblioteca *Jama* foi utilizada para implementar no código Java a criação das matrizes $\square$ e $\square$ e também para solucionar esse sistema, armazenando a solução na matriz $\square$. Essa é a parte crucial do algoritmo, já que, calculados os elementos da matriz $\square$, bastaria "montar" a expressão do termo geral.

Um problema nesse ponto do desenvolvimento foi que os coeficientes do termo geral armazenados em $\square$ eram todos obtidos na forma decimal, e desejava-se apresenta-los na forma de fração. Fez-se então necessário converter os números da forma decimal para a fracionária. Para este caso decidiu-se desenvolver um algoritmo próprio para conversão de decimais em fração, em vez de procurar e utilizar alguma biblioteca de código que implementasse essa funcionalidade. Esse algoritmo, além de ser utilizado para a apresentação dos coeficientes do termo geral das sequências na forma fracionária, acabou sendo implementado como uma funcionalidade extra do SequenceCalc, tendo ele agora uma interface própria para esse propósito. Tanto essa interface, nomeada Módulo **Conversor**, quanto o algoritmo utilizado, são apresentados no subtópico 3.2.2 deste trabalho.

Após resolver este problema, outros detalhes relacionados ao termo geral da sequência foram adicionados ao aplicativo, como:

- A adição de um espaço textual para exibição da lei de formação da sequência em *LaTeX* utilizando funções e componentes da biblioteca *JLatexMath* – antes disto, a expressão do termo geral era apresentada toda em uma única linha e muitas vezes, para correta apresentação, contendo muitos parênteses;
- A informação da quantidade de termos da sequência digitada e qual o próximo termo de acordo com a expressão do termo geral calculada;

[8] Página do projeto: https://github.com/opencollab/jlatexmath
[9] Um sistema ou programa de marcação para a editoração de documentos de alta qualidade tipográfica, específico para a elaboração de textos científicos. Muito útil para apresentar fórmulas e símbolos matemáticos.

- Um campo de entrada para permitir que seja solicitado o n-ésimo termo da sequência de acordo com a expressão do termo geral calculada;
- Um botão para troca da variável que aparece na expressão do termo geral. A variável padrão é n e o aplicativo permite apenas a troca para uma letra minúscula do alfabeto latino.
- Um botão para compartilhar, via aplicativos do smartphone, o texto que expressa o termo geral obtido (via WhatsApp ou e-mail, por exemplo).

Apenas por capricho, adicionou-se também uma função no código para identificar a sequência de Fibonacci e apresentar sua expressão de termo geral exponencial clássica:

$$F(n) = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n} - \left(\frac{1-\sqrt{5}}{2}\right)^{n}\right]$$

Os cinco primeiros termos dessa sequência, em ordem, foram também implementados como sequência padrão do aplicativo, de forma que, caso o campo de inserção da sequência esteja vazio e acione-se o cálculo do termo geral de alguma sequência, o campo de inserção será preenchido com os referidos cinco termos e a expressão acima mencionada será apresentada.

Além desses detalhes, aplicados à interface principal do aplicativo – que será referenciada neste trabalho como Módulo **Principal** –, foram adicionados também ao aplicativo o já mencionado Módulo **Conversor**, para converter valores na forma decimal para a forma de fração irredutível, e um Módulo **Sobre**, para apresentar as instruções de uso do aplicativo e outras informações de créditos e contato. Cada um destes Módulos será apresentado na próxima seção.

3.2 Interface e comportamentos

Nesta seção será apresentada a interface[10] visual do SequenceCalc, como utilizá-lo e o comportamento dele de acordo com as ações realizadas pelo usuário (pessoa que o utilizará).

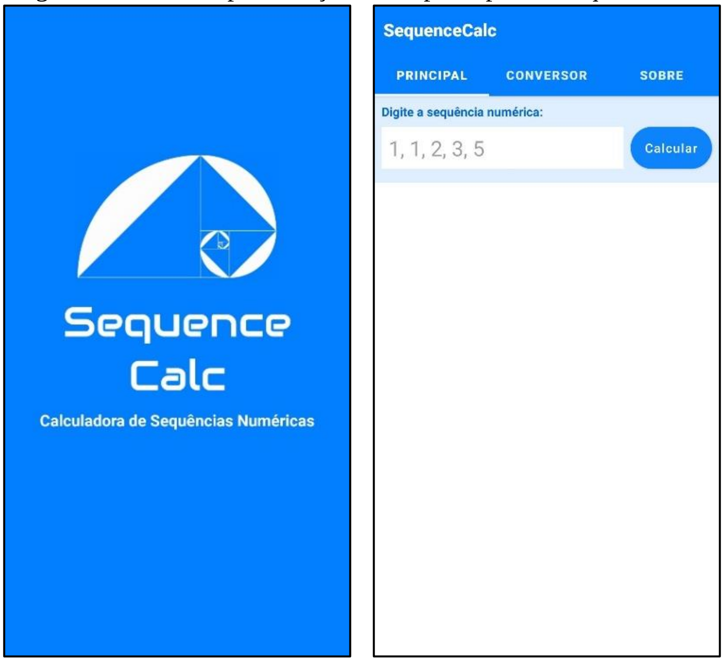
Ao iniciar o SequenceCalc, uma tela de apresentação não-interativa aparece por 3 segundos. Nesta tela, mostra-se uma animação contendo uma imagem, o nome do aplicativo e a caracterização de sua principal funcionalidade – Calculadora de Sequências Numéricas. Em

---

[10] Capturas de tela do aplicativo serão utilizadas para ilustração nesta seção e partes delas poderão estar suprimidas para melhor ajuste no texto. Nestes casos, as partes suprimidas não são significativas. Diferenças sutis na exibição da interface do aplicativo também podem ocorrer a depender do dispositivo utilizado para executá-lo.

seguida, aparece a tela principal dele, contendo, na parte superior, o nome do aplicativo e três guias de conteúdo (Figura 4).

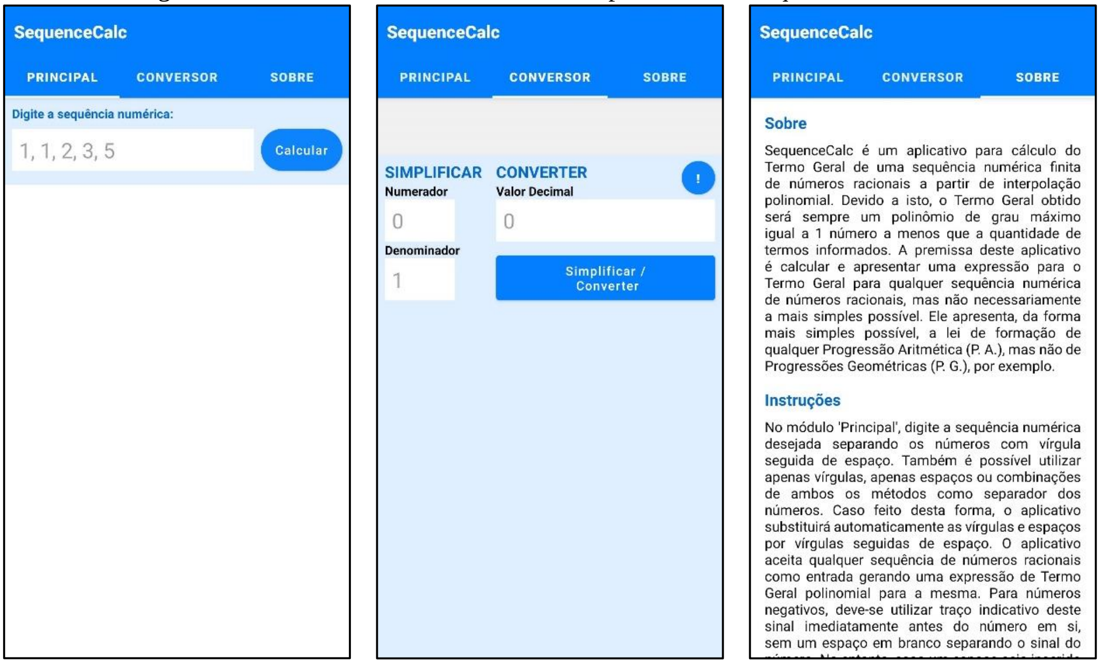
Figura 4 – Tela de apresentação e tela principal do SequenceCalc



Fonte: Capturas de tela do aplicativo SequenceCalc.

As três guias são alternáveis e cada uma delas, ao ser acionada com o toque, muda a interface e o contexto da parte logo abaixo delas. Estas guias e seus contexto serão referenciados neste trabalho como **Módulos**, denominando-se, da esquerda para a direita: Módulo **Principal**, Módulo **Conversor** e Módulo **Sobre**. A Figura 5 apresenta capturas de tela do aplicativo estando ativos cada um desses Módulos.

Figura 5 – Conteúdos dos três Módulos do aplicativo assim que ele é iniciado



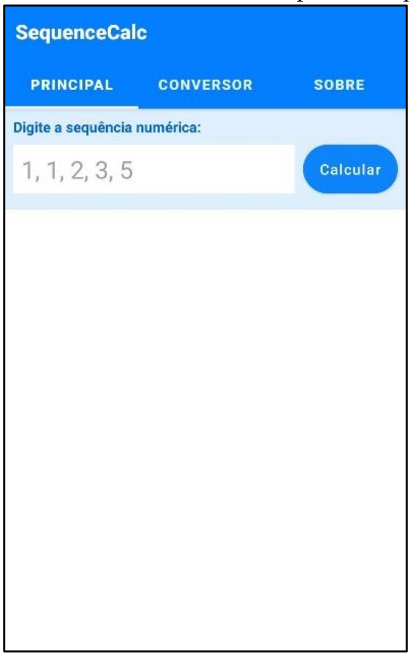
Fonte: Capturas de tela do aplicativo SequenceCalc.

As possibilidades de utilização do aplicativo se resumem ao que ele permite que seja feito a partir destes três módulos. Nos próximos tópicos serão apresentados os elementos de interface e funcionalidades de cada um deles.

3.2.1 Módulo Principal

O Módulo Principal do SequenceCalc será sempre o módulo mostrado após o fechamento da tela de apresentação. Por ele se tem acesso à principal funcionalidade do SequenceCalc: a inserção da sequência numérica para que seja calculado e apresentado o seu termo geral. A interface deste Módulo conta, inicialmente, apenas com um texto para orientação, um campo de entrada para inserção da sequência e um botão “Calcular” (Figura 6).

Figura 6 – Interface do Módulo Principal do SequenceCalc



Fonte: Captura de tela do aplicativo SequenceCalc.

No campo de entrada de sequência aparece, em cores claras, a sequência "1, 1, 2, 3, 5" – cinco primeiros termos da sequência de Fibonacci –, tomada, subjetivamente, como uma sequência de exemplo para esse campo. Qualquer outra sequência pode ser digitada para que o aplicativo calcule seu termo geral, mas, caso nenhuma seja informada e o botão "Calcular" seja acionado, o aplicativo apresentará o termo geral correspondente a essa sequência de exemplo.

Algo importante a se considerar ao digitar a sequência numérica no campo de entrada de sequência é a sintaxe de inserção e de formato de número. O termo "sintaxe" refere-se à forma ou estrutura de representação textual das sequências e dos números requeridas pelo aplicativo. Sobre isso, resumidamente, temos o seguinte:

- os elementos das sequências devem ser números racionais – positivos ou negativos – escritos na forma de números inteiros (quando inteiros) ou na forma decimal (quando fracionários) e devem ser separados por vírgula seguida de espaço;
- quando na forma decimal, o separador decimal utilizado deve ser o ponto decimal (".");
- o indicador de número negativo (sinal de menos), deve ser inserido imediatamente à esquerda do número a ser negativado, sem nenhum espaço entre o sinal e o número;
- não há, na versão atual do SequenceCalc, uma forma não ambígua de indicar que um número é uma dízima periódica. O que se orienta, quando desejado representa-las, é repetir o período da dízima por, pelo menos, 8 vezes. Desta forma, o aplicativo irá interpretar o número inserido como uma dízima periódica. Tecnicamente, isso torna

impossível a precisa conversão em fração de números com uma grande – porém, finita – quantidade de dígitos decimais que aparentem ser dízimas periódicas (possuam repetição de dígitos decimais), influenciando a representação do termo geral de sequências que os contenham.

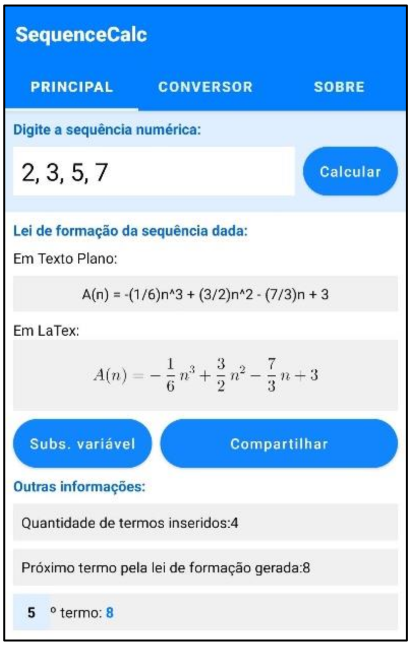
No aplicativo, as informações sobre essas sintaxes podem ser acessadas no Módulo Sobre do SequenceCalc, apresentado no tópico 3.2.3 deste trabalho.

Considerando essas sintaxes e partindo da interface inicial do Módulo Principal do SequenceCalc, três situações podem ocorrer ao ser acionado o botão “Calcular”: o campo de entrada de sequência estar ou não preenchido e, caso preenchido, ter sido ou não utilizada a sintaxe correta de preenchimento. As subseções seguintes irão abordar essas situações.

3.2.1.1 Campo de entrada preenchido utilizando a sintaxe correta

Se o campo de entrada de sequência estiver preenchido com alguma sequência utilizando a sintaxe correta, o aplicativo irá calcular e apresentar (de duas maneiras), na parte inferior da tela – antes, totalmente vazia – o termo geral da sequência digitada e outras informações relacionadas a ela, como: a quantidade de termos inseridos, qual o próximo termo e qual o *n*-ésimo termo da sequência pela expressão do termo geral obtida (Figura 7). Para a apresentação do *n*-ésimo termo, é necessário informar qual o valor de *n* (posição do termo desejado na sequência).

Figura 7 – Sequência com sintaxe correta e acionando o botão “Calcular”

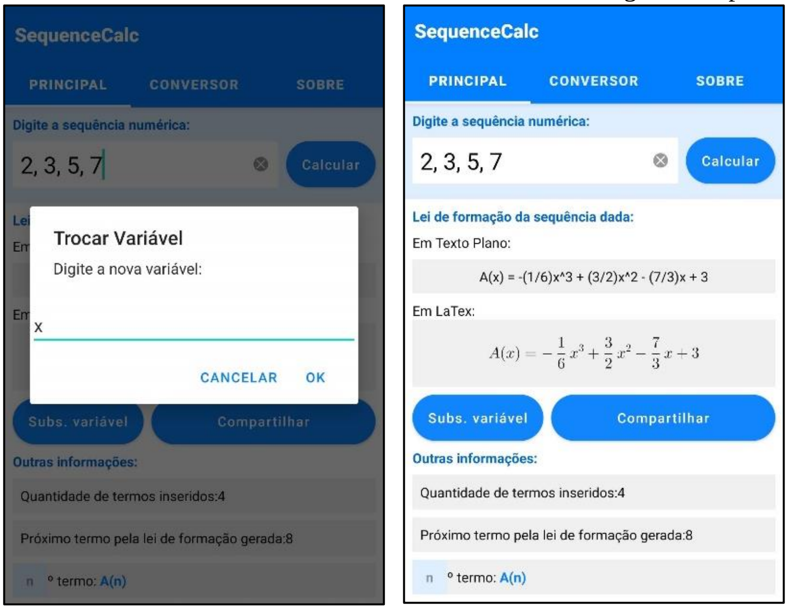


Fonte: Captura de tela do aplicativo SequenceCalc.

Além dos elementos informativos, também aparecem o botão “Subs. variável”, para permitir a substituição da variável padrão *n* na expressão do termo geral por alguma outra, e o botão “Compartilhar”, para permitir o envio das expressões do termo geral para algum aplicativo de mensagens instalado no smartphone.

A substituição de variável foi implementada pensando principalmente no compartilhamento da expressão do termo geral. Pode-se desejar utilizar a expressão do termo geral em algum software matemático – para, por exemplo, plotar o gráfico da função representada por essa expressão – e acontecer de o referido software apenas reconhecer determinadas variáveis – como *x*, *y* e *z*, por exemplo. A troca de variável permite que a expressão do termo geral seja compartilhada já com a variável adequada, não necessitando, posteriormente, editá-la para o uso. Para permitir a escolha da nova variável – que poderá ser, limitadamente, uma única letra do alfabeto latino – o aplicativo apresenta uma interface extra com um campo de entrada para que se digite a nova variável. Após digitá-la e confirmar, o aplicativo mostrará novamente a interface do Módulo Principal com a expressão do termo geral já utilizando a nova variável (Figura 8).

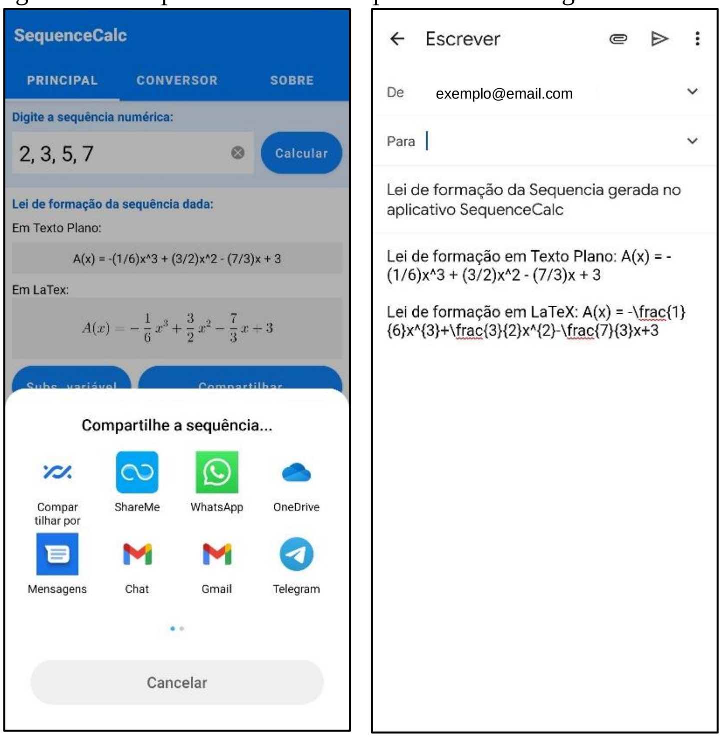
Figura 8 – Interface e resultado da troca de variável do termo geral no aplicativo



Fonte: Capturas de tela do aplicativo SequenceCalc.

O botão "Compartilhar" permite que se escolha um dos softwares de mensagens disponíveis entre os instalados no smartphone para o compartilhamento da expressão do termo geral da sequência. A Figura 9 mostra a interface de escolha de aplicativo para compartilhamento (esquerda) e a forma como aparece a mensagem a ser compartilhada quando escolhido um aplicativo de e-mail (direita).

Figura 9 – Compartilhamento da expressão do termo geral via e-mail



Fonte: Capturas de tela do aplicativo SequenceCalc.

Da mesma forma que na interface do Módulo Principal, a expressão do termo geral compartilhada apresenta-se de duas maneiras: em texto plano e em *LaTeX*. Referiu-se como texto plano, a forma de escrita de expressões matemáticas sem o uso de um editor especializado, com a representação de frações, por exemplo, ocupando uma linha única de texto. Essa forma de escrita é a indicada para ser utilizada na maioria dos casos, considerando não ser algo comum para a maioria das pessoas a utilização de editores de texto matemático especializados. Já o texto em *LaTeX* é mais específico, sendo útil apenas quando utilizado em um editor de texto *LaTeX*. Incluiu-se também a exibição e compartilhamento da expressão do termo geral das sequências nesse formato pela alta qualidade tipográfica possibilitada por ele na escrita de textos matemáticos.

3.2.1.2 Campo de entrada preenchido utilizando a sintaxe incorreta

Se o campo de entrada de sequência estiver preenchido com alguma sequência numérica utilizando a sintaxe incorreta, o aplicativo apresentará na tela, por alguns segundos, uma mensagem de erro solicitando verificar a sequência inserida, conforme mostrado na Figura 10.

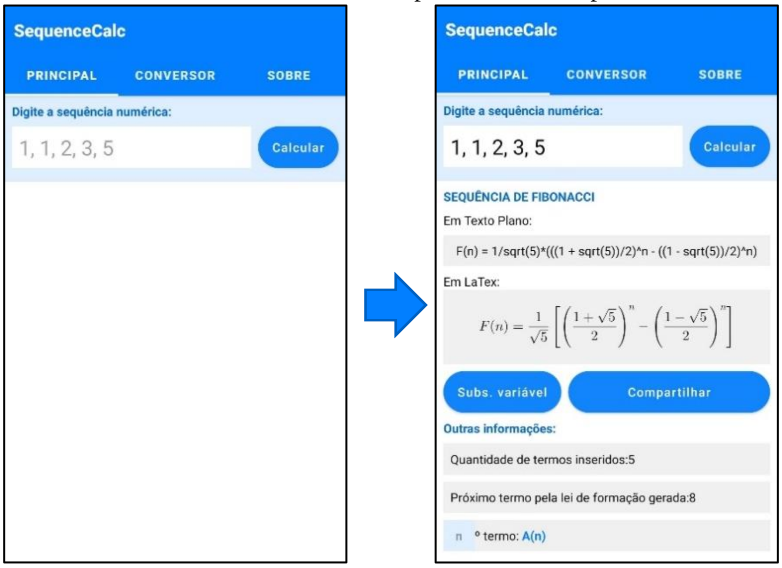
Figura 10 – Sequência com sintaxe incorreta e acionando o botão “Calcular”

Fonte: Captura de tela do aplicativo SequenceCalc.

3.2.1.3 Campo de entrada vazio (não preenchido)

Se o campo de entrada de sequência estiver vazio (nenhuma sequência for digitada), o aplicativo calculará o termo geral da sequência de exemplo “1, 1, 2, 3, 5” (Figura 11).

Figura 11 – Acionando o botão "Calcular" sem preencher o campo de entrada de sequência



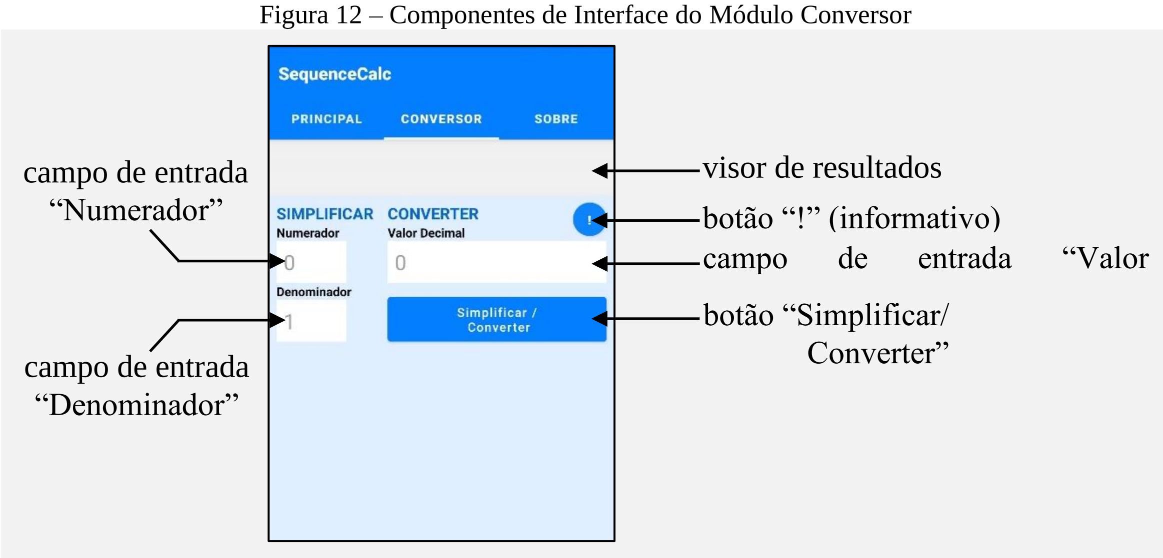
Fonte: Captura de tela do aplicativo SequenceCalc.

3.2.2 Módulo Conversor

No Módulo Conversor foi implementada a interface de um conversor e simplificador de frações. Essa funcionalidade foi adicionada aproveitando o código utilizado para converter em fração os coeficientes do termo geral das sequências inseridas no Módulo Principal.

A interface deste Módulo é composta por um **visor** cinza na parte superior (para apresentação de resultados), três **campos de entrada** indicados por "numerador", "denominador" e "valor decimal", onde é possível digitar valores numéricos, o **botão** "Simplificar/Converter" e o **botão** informativo "!" (Figura 12).

Figura 12 – Componentes de Interface do Módulo Conversor



Fonte: Captura de tela do aplicativo SequenceCalc (montagem, pelo autor).

A partir destes componentes é possível simplificar frações e converter números racionais entre as formas fracionária e decimal. Para isto, deve-se informar valores numéricos nos campos de entrada disponíveis e acionar o botão “Simplificar/Converter”. O aplicativo não permite que valores sejam informados nos 3 campos em uma mesma operação, de forma que, caso informe-se o campo “Valor Decimal” e os campos “Numerador” ou “Denominador” forem posteriormente editados, o aplicativo apagará o valor decimal informado, ocorrendo o mesmo com os campos “Numerador” e “Denominador” caso algum valor tenha sido informado neles e posteriormente seja editado o campo “Valor Decimal”.

Para converter números na forma fracionária para a forma decimal, deve-se realizar os seguintes passos:

1) Informa-se, nos campos correspondentes, o numerador e o denominador da fração que se deseja converter para decimal;
2) Aciona-se o botão “Simplificar/Converter”;
3) O aplicativo divide o valor do numerador pelo denominador obtendo como resultado um número na forma decimal e apresenta este resultado no campo “Valor Decimal”. O aplicativo também irá mostrar no visor a forma irredutível da fração dada como entrada. Essa forma irredutível é calculada através do mesmo processo de simplificação de frações, que será apresentado a seguir.

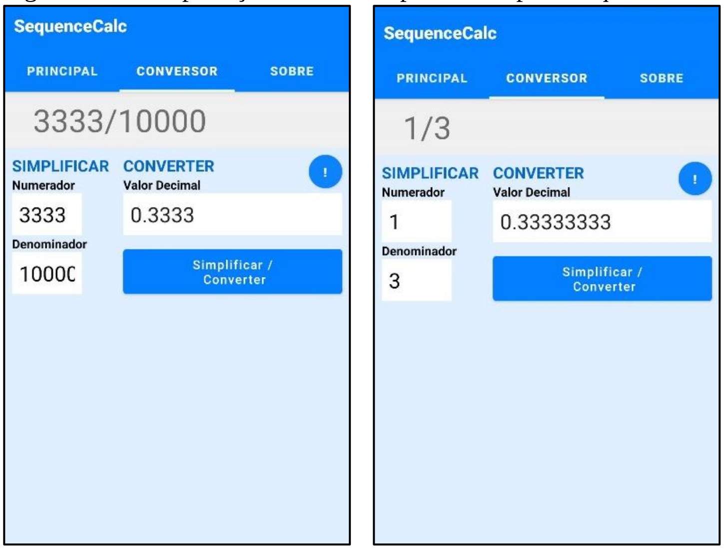
O processo de simplificação da fração é realizado automaticamente ao acionar o botão “Simplificar/Converter” e utiliza como entrada o número contido no campo “Valor Decimal” do Módulo Conversor, que poderá ser preenchido pelo usuário ou ser calculado a partir dos valores contidos nos campos “Numerador” e “Denominador”, que poderão também ser preenchidos pelo usuário ou utilizar valores padrão pré-estabelecidos. Os valores padrão são: 0 para o campo “Numerador” e 1 para o campo “Denominador”. A seguir são apresentados os passos realizados pelo aplicativo para a simplificação da fração inserida:

1) O aplicativo considera uma fração $k = a/b$ cujo numerador a será o número contido no campo “Valor Decimal” (não necessariamente inteiro) e o denominador b será igual a “1”.
2) O aplicativo multiplica a por b e verifica se o resultado é um número inteiro (utilizando um comando da linguagem de programação Java). Nisto, há duas possibilidades:
   - O resultado ser inteiro: Neste caso, o aplicativo atribui este resultado a a e, com isto, tanto a quanto b serão números inteiros. O processo de simplificação então se encerra e o aplicativo exibe no visor de resultados a fração a/b.
   - O resultado não ser inteiro: Neste caso, o aplicativo soma 1 unidade a b e repete o passo 2, até que o resultado da multiplicação seja inteiro.

Este método é o mesmo utilizado para converter em fração os coeficientes do termo geral das sequências numéricas calculados no Módulo Principal do aplicativo.

O botão informativo “!”, ao ser acionado, apresenta a seguinte mensagem: “Para a correta conversão de dízimas, repita o período por 8 ou mais casas decimais”. Isso se faz necessário porque, caso poucas casas decimais sejam informadas, o aplicativo pode não interpretar o número inserido como uma dízima periódica. Por exemplo, caso queiramos converter a dízima “0,3333...” em fração e utilizarmos apenas 4 casas decimais, o aplicativo apresentará como resultado a fração 3333/10000 e não 1/3 (Figura 13).

Figura 13 – Interpretação de dízimas periódicas pelo SequenceCalc



Fonte: Captura de tela do aplicativo SequenceCalc.

A partir de 5 casas decimais o aplicativo já interpretará o número decimal periódico como uma dízima e fará a conversão em fração levando isso em consideração. No entanto, por questões de arredondamento de valores, indica-se a repetição do período por 8 ou mais casas decimais, principalmente para dízimas com período composto por 2 ou mais dígitos.

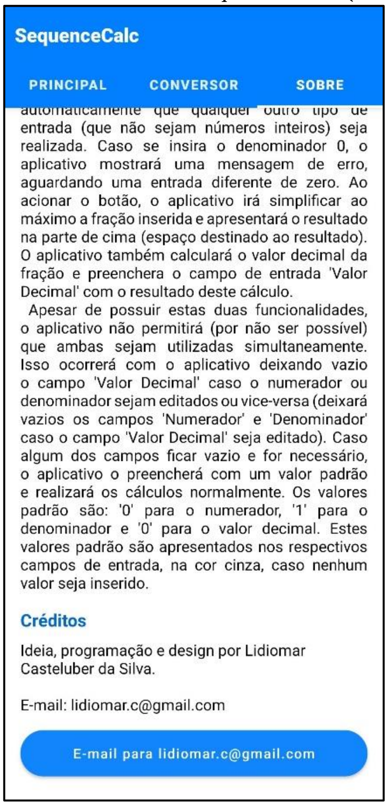
No momento, indica-se apenas a repetição do período da dízima por 8 ou mais casas decimais para contornar este problema, mas pretende-se, futuramente, implementar uma forma específica de representar dízimas periódicas para que problemas como este não venham a acontecer. A repetição do período por várias casas decimais funciona pois o aplicativo trabalha com arredondamento de certos resultados quando eles se aproximam muito de um número inteiro.

Testes informais, não documentados, foram realizados para verificar a eficácia aparente do aplicativo com relação às conversões envolvendo dízimas. Os resultados foram satisfatórios, mas a verificação das margens de erro reais dessas conversões ainda precisa ser realizada. Assim como a implementação da forma específica de representação das dízimas periódicas, testes documentados também ficarão para trabalhos futuros.

3.2.3 Módulo Sobre

O Módulo Sobre apresenta basicamente textos informativos sobre o SequenceCalc: apresentação, instruções de uso (sintaxe de inserção das sequências e de formato de números), créditos pela criação e e-mail para contato com o desenvolvedor. Além dessa parte textual, na parte mais inferior da interface deste módulo, há um botão para acionar a edição e envio de sugestões ou comentários para o desenvolvedor por e-mail (Figura 14).

Figura 14 – Módulo *Sobre* do SequenceCalc (final da página)



Fonte: Captura de tela do aplicativo SequenceCalc.

A transcrição dos textos apresentados nesse Módulo consta no APÊNDICE A – Transcrição do texto do Módulo Sobre do Sequencecalc.

4 RESOLVENDO PROBLEMAS COM O SEQUENCECALC

Neste capítulo serão apresentados dois problemas matemáticos que serão solucionados utilizando a funcionalidade principal do SequenceCalc: a obtenção do termo geral (ou lei de formação) de uma sequência finita de números racionais. Os problemas descritos originam-se de experiências do autor deste trabalho.

O primeiro problema consiste em, dados os primeiros termos de uma sequência, encontrar o próximo termo. É um tipo de problema clássico e muito recorrente em redes sociais. O diferencial deste, em específico, são seus primeiros elementos, que seguem uma lei de formação bastante clara, mas que em determinado ponto foge totalmente da regra identificada. A expressão do termo geral obtida para ela é também consideravelmente interessante.

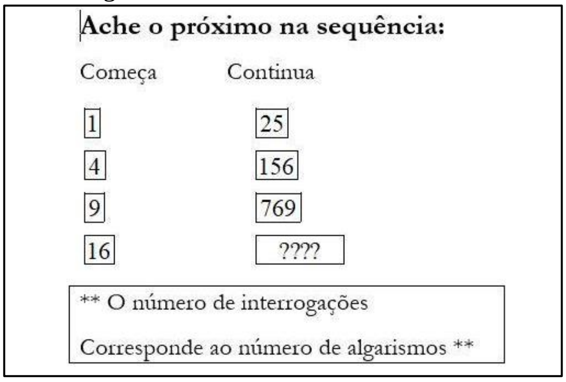
No segundo problema, o SequenceCalc atua em conjunto com outro software matemático: o GeoGebra. Sendo um software muito popular entre matemáticos e professores de matemática, o GeoGebra permite construções geométricas, apresenta as expressões algébricas correspondentes e possibilita que diversas manipulações sejam realizadas com esses elementos. Uma das formas de realizar construções e manipular os elementos no GeoGebra é através de comandos de texto e o problema em questão reside em realizar no Geogebra uma construção que envolve sequências utilizando um único comando. O SequenceCalc irá ter papel importantíssimo na etapa final da solução deste problema.

Todo o contexto, apresentação e resolução de ambos os problemas serão apresentados nas seções seguintes e o SequenceCalc terá papel de destaque nesse processo, seja sendo um facilitador de soluções e possibilitando novas descobertas, seja permitindo o acesso a elementos cruciais para essas soluções.

4.1 Encontrando o próximo termo de uma sequência numérica

Problemas que pedem o próximo termo de uma sequência numérica são relativamente comuns. Sejam problemas presentes no estudo da matemática em contexto escolar (em livros didáticos, ao estudar sequências, por exemplo) ou encarados de forma recreativa (compartilhados em alguma rede social como o Facebook ou WhatsApp), não é difícil já não termos nos deparado com um problema deste tipo. O problema seguinte (Figura 15), por exemplo, foi retirado do grupo de discussão “Loucos por Matemática” no Facebook[11]:

Figura 15 – Enunciado do Problema 1


Ache o próximo na sequência:

Começa	Continua
1	25
4	156
9	769
16	????

** O número de interrogações

Corresponde ao número de algarismos **

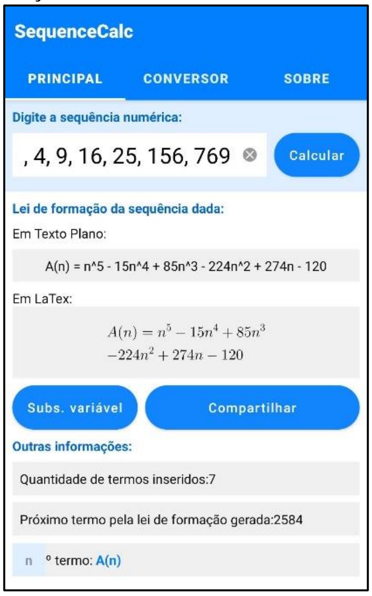
Fonte: Grupo "Loucos por Matemática" no Facebook.

O problema apresenta uma sequência de oito termos em que, até o quinto, representa os quadrados das posições destes termos na sequência. A partir do sexto termo, no entanto, este padrão é descontinuado, de forma a dificultar a percepção de uma regra que permita encontrar o termo solicitado (oitavo).

Resolver este problema utilizando o SequenceCalc é extremamente simples. Basta digitar a sequência dos números na ordem em que ela foi apresentada no enunciado e acionar o botão “Calcular”. O aplicativo apresentará, então, uma lei de formação polinomial para esta sequência, a quantidade de termos inseridos, qual o próximo termo da sequência e um campo para verificar um outro termo qualquer (*n*-ésimo termo), considerando a expressão de termo geral calculada. Para este problema, o aplicativo apresenta o valor 2584 como o próximo termo da sequência, como visto na Figura 16.

11 Disponível em: https://www.facebook.com/groups/loucospormatematicaoficial/permalink/1644071528971155/ (grupo privado). Acesso em: 17 nov. 2021.

Figura 16 – Solução do Problema 1 utilizando o SequenceCalc



Fonte: Aplicativo SequenceCalc.

A mérito de comparação, serão apresentadas, a seguir, duas formas de solucionar este problema sem a utilização do aplicativo.

A primeira forma de solução pode ser aplicada em sequências classificadas como Progressões Aritméticas (P.A.) de Ordem Superior. Segundo Iezzi e Hazzan (2013, p. 6), "uma P.A. é uma sequência em que cada termo, a partir do segundo, é a soma do anterior com uma constante r dada", $r \in \mathbb{R}$. Esta constante r pode, portanto, ser obtida a partir da diferença entre um termo da sequência (a partir do segundo) e seu antecessor. Nobre e Rocha (2018) definem, para uma dada sequência, o chamado "operador diferença" $\left(\Delta^1 a_n\right)_{n \in \mathbb{N}} = (a_{n+1} - a_n)_{n \in \mathbb{N}}$, que constitui uma nova sequência. Como $\left(\Delta^1 a_n\right)_{n \in \mathbb{N}}$ forma uma nova sequência, pode-se novamente obter o operador diferença, isto é, $\left(\Delta^1\left[\Delta^1 a_n\right]\right)_{n \in \mathbb{N}} = \left(\Delta^2 a_n\right)_{n \in \mathbb{N}}$, e assim recursivamente, $\left(\Delta^k a_n\right)_{n \in \mathbb{N}}$, para $k \geq 3$. A partir desta definição, segue que "uma sequência $(a_n)_{n \in \mathbb{N}}$ será uma P.A. de ordem k se for necessário aplicar o operador diferença k vezes para se chegar a uma sequência constante" (NOBRE; ROCHA, 2018, p. 37).

Seja $(a_n)_{n \in \mathbb{N}} = (1, 4, 9, 16, 25, 156, 769, \ldots)$ a sequência apresentada no problema, aplicando-a o operador diferença, teremos o seguinte:

$$(a_n)_{n \in \mathbb{N}} = (1, 4, 9, 16, 25, 156, 769, \ldots)$$

$$\left(\Delta^1 a_n\right)_{n \in \mathbb{N}} = (3, 5, 7, 9, 131, 613, \ldots)$$

$$\left(\Delta^2 a_n\right)_{n \in \mathbb{N}} = (2, 2, 2, 122, 482, \ldots)$$

$$\left(\Delta^3 a_n\right)_{n \in \mathbb{N}} = (0, 0, 120, 360, \ldots)$$

$$\left(\Delta^4 a_n\right)_{n \in \mathbb{N}} = (0, 120, 240, \ldots)$$

$\left(\Delta^5 a_n\right)_{n\in\mathbb{N}} = (120, 120, \dots)$

Como $\left(\Delta^5 a_n\right)_{n\in\mathbb{N}}$ é uma sequência constante, encerra-se a aplicação do operador diferença. A partir daqui, para obter o próximo termo da sequência inicial, basta realizarmos as seguintes operações:

- Somamos o termo constante de $\left(\Delta^5 a_n\right)_{n\in\mathbb{N}}$ ao último termos de $\left(\Delta^4 a_n\right)_{n\in\mathbb{N}}$ e incluímos o resultado a $\left(\Delta^4 a_n\right)_{n\in\mathbb{N}}$:

  $120 + 240 = 360$

  $\left(\Delta^4 a_n\right)_{n\in\mathbb{N}} = (0, 120, 240, 360 \dots)$

- Somamos o novo último termo de $\left(\Delta^4 a_n\right)_{n\in\mathbb{N}}$ ao último termo de $\left(\Delta^3 a_n\right)_{n\in\mathbb{N}}$ e incluímos o resultado $\left(\Delta^3 a_n\right)_{n\in\mathbb{N}}$. Repetimos o processo para $\left(\Delta^2 a_n\right)_{n\in\mathbb{N}}$, $\left(\Delta^1 a_n\right)_{n\in\mathbb{N}}$ e finalmente à $(a_n)_{n\in\mathbb{N}}$, obtendo seu próximo termo:

  $360 + 360 = 720 \Rightarrow \left(\Delta^3 a_n\right)_{n\in\mathbb{N}} = (0, 0, 120, 360, 720 \dots)$

  $720 + 482 = 1202 \Rightarrow \left(\Delta^2 a_n\right)_{n\in\mathbb{N}} = (2, 2, 2, 122, 482, 1202 \dots)$

  $1202 + 613 = 1815 \Rightarrow \left(\Delta^1 a_n\right)_{n\in\mathbb{N}} = (3, 5, 7, 9, 131, 613, 1815 \dots)$

  $1815 + 769 = 2584 \Rightarrow (a_n)_{n\in\mathbb{N}} = (1, 4, 9, 16, 25, 156, 769, 2584 \dots)$

Desta forma, o próximo termo da sequência será 2584.

Algo perceptível quanto a este método de solução é que ele não apresenta a lei de formação da sequência apresentada no problema, servindo apenas para encontrar os próximos termos da sequência desde que saibamos os termos anteriores. Se o problema solicitasse um termo específico da sequência, o vigésimo, por exemplo, precisaríamos calcular todos os termos anteriores, até o décimo nono, para que, enfim, pudéssemos calcular o valor do vigésimo termo. Ter em mãos a lei de formação (ou termo geral) da sequência poderia ser muito útil neste tipo de situação.

Outro método de solução do problema sem o aplicativo seria, então, encontrar a lei de formação da sequência através da interpolação polinomial e, a partir dela, calcular o valor do oitavo termo solicitado. A vantagem deste método seria, como já comentado, a possibilidade de calcular com facilidade o valor de qualquer outro termo desejado da sequência. Em contrapartida, a interpolação manual dos sete termos desta sequência para encontrar sua lei de formação necessitaria da solução de um sistema de equações com sete equações e sete variáveis, o que certamente seria muito trabalhoso. Apresentaremos a seguir quais seriam os passos para a realização deste cálculo sem, no entanto, mostrar o desenvolvimento dos mesmos, focando na apresentação dos resultados.

Temos a seguinte sequência:

$$(a_n)_{n\in\mathbb{N}} = (1, 4, 9, 16, 25, 156, 769, \ldots)$$

Como esta sequência possui sete termos, sua lei de formação será da forma

$$(a_n)_{n\in\mathbb{N}} = a_0 + a_1 n + a_2 n^2 + a_3 n^3 + a_4 n^4 + a_5 n^5 + a_6 n^6$$

Para interpolar os termos desta sequência, podemos associá-la à seguinte matriz:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 & 32 & 64 & 4 \\ 1 & 3 & 9 & 27 & 81 & 243 & 729 & 9 \\ 1 & 4 & 16 & 64 & 256 & 1024 & 4096 & 16 \\ 1 & 5 & 25 & 125 & 625 & 3125 & 15625 & 25 \\ 1 & 6 & 36 & 216 & 1296 & 7776 & 46656 & 156 \\ 1 & 7 & 49 & 343 & 2401 & 16807 & 114649 & 769 \end{bmatrix}$$

Escalonando essa matriz obteremos o seguinte resultado:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 7 & 15 & 31 & 63 & 3 \\ 0 & 0 & 2 & 12 & 50 & 180 & 602 & 2 \\ 0 & 0 & 0 & 6 & 60 & 390 & 2100 & 0 \\ 0 & 0 & 0 & 0 & 24 & 360 & 3360 & 0 \\ 0 & 0 & 0 & 0 & 0 & 120 & 2520 & 120 \\ 0 & 0 & 0 & 0 & 0 & 0 & 720 & 0 \end{bmatrix}$$

Resolvendo o sistema associado à matriz anterior, obteremos as seguintes soluções:

$$a_6 = 0, a_5 = 1, a_4 = -15, a_3 = 85, a_2 = -224, a_1 = 274 \text{ e } a_0 = -120$$

Dessa forma, a expressão do termo geral da sequência inicial será dada por:

$$(a_n)_{n\in\mathbb{N}} = n^5 - 15n^4 + 85n^3 - 224n^2 + 274n - 120$$

Como esperado, essa expressão é a mesma apresentada pelo SequenceCalc. A partir de algumas manipulações algébricas podemos reescrevê-la da seguinte maneira:

$$(a_n)_{n\in\mathbb{N}} = (n-1)(n-2)(n-3)(n-4)(n-5) + n^2$$

Essa reescrita é interessante pois possibilita que percebamos a lógica por trás do padrão encontrado nos cinco primeiros termos da sequência: quando *n* assume os valores naturais de 1 a 5, o produto $(n-1)(n-2)(n-3)(n-4)(n-5)$ se anula, de forma a restar o quadrado de *n*.

Pode-se definir ainda a expressão condicional abaixo a partir dessa solução:

$$(a_n)_{n\in\mathbb{N}} = \begin{cases} n^2, \text{se } n \leq 5 \\ n^2 + \dfrac{(n-1)!}{(n-6)!}, \text{se } n > 5 \end{cases}$$

$$\square_8 = 8^2 + \frac{7!}{2!} = 64 + \frac{5040}{2} = 64 + 2520 = 2584$$

A sequência numérica apreciada nesse problema é, como pôde-se perceber, bastante interessante. Inicialmente, pela peculiaridade dos elementos que a compõe, mas, principalmente, pela justificativa dessa peculiaridade, vislumbrada através da expressão de seu termo geral. O SequenceCalc teve o papel importante de facilitar o processo de solução do problema, permitindo que se chegasse a uma resposta válida e que posteriormente pôde ser ainda mais refinada.

4.2 Alterando a ordem dos elementos de uma sequência pré-definida

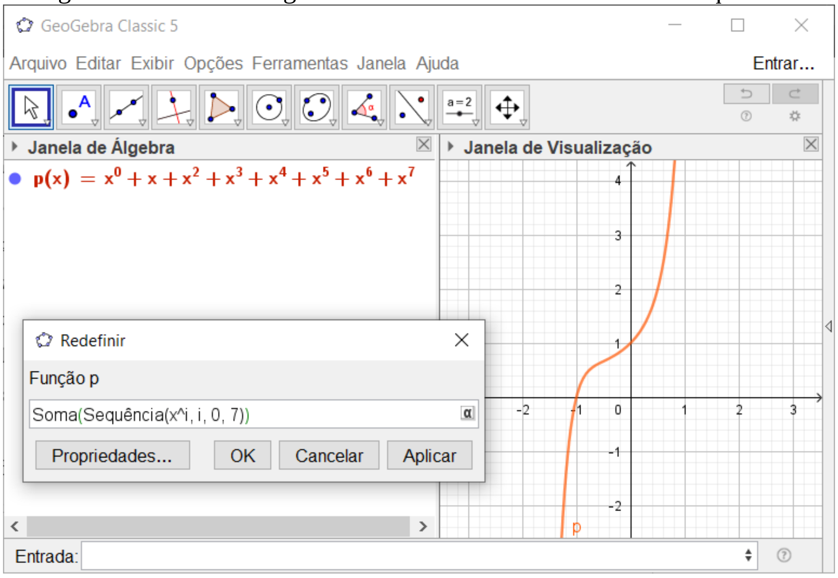
No ano de 2018 aconteceu a 15ª edição do curso online sobre o software de matemática dinâmica GeoGebra, oferecido pelo site “O GeoGebra”[12]. Neste curso, aprendia-se a utilizar as ferramentas do GeoGebra e realizava-se tarefas para praticar. Como uma das tarefas do curso, foi disponibilizada uma lista com enunciados de 30 problemas matemáticos selecionados pelos organizadores. Estes problemas deveriam ser resolvidos através do GeoGebra utilizando os conhecimentos desenvolvidos no decorrer do curso. Um dos problemas era o seguinte: As raízes do polinômio $1 + \square + \square^2 + \square^3 + \square^4 + \square^6 + \square^7$, quando representadas no plano complexo, formam os vértices de um polígono convexo. Qual é a área desse polígono?[13]

O Geogebra possui (por padrão, na parte inferior de sua janela) uma caixa de texto em que podemos digitar comandos para construir objetos, executar transformações, obter medidas, entre outras possibilidades. Uma das formas mais simples de solucionar este problema seria através de comandos inseridos neste Campo de Entrada. Uma das possiblidades é a solução em 3 passos a seguir:

1. Cria-se o polinômio definido no enunciado a partir do Campo de Entrada do Geogebra, utilizando o comando *p(x) = Soma(Sequência(x^i, i, 0, 7))* (Figura 17).

[12] Acessível em: https://ogeogebra.com.br/site/
[13] Enunciado 21 da lista disponível em: https://www.ogeogebra.com.br/arquivos/enunciados15.pdf. Acesso em: 17 nov. 2021.

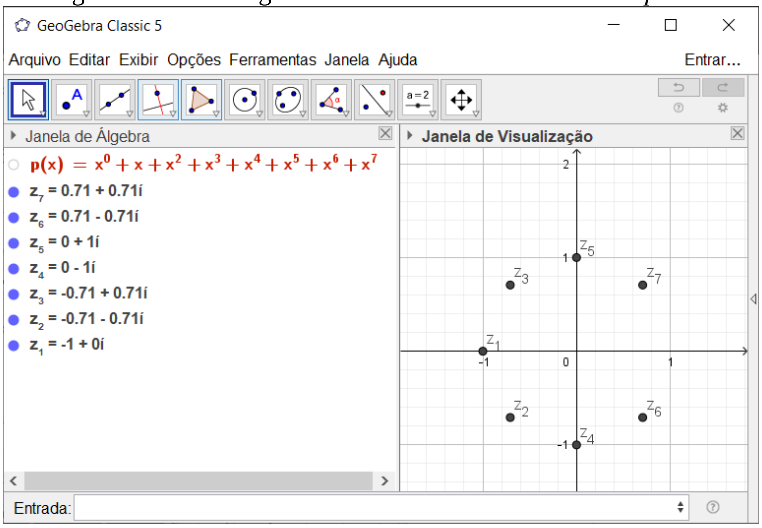
Figura 17 – Polinômio gerado com os comandos "Soma" e "Sequência"



Fonte: Software GeoGebra.

2. Oculta-se o gráfico do polinômio *p(x)* e, para obter e visualizar os pontos correspondentes às suas raízes complexas na janela de visualização, digita-se no Campo de Entrada o comando *RaízesComplexas(p)* (Figura 18).

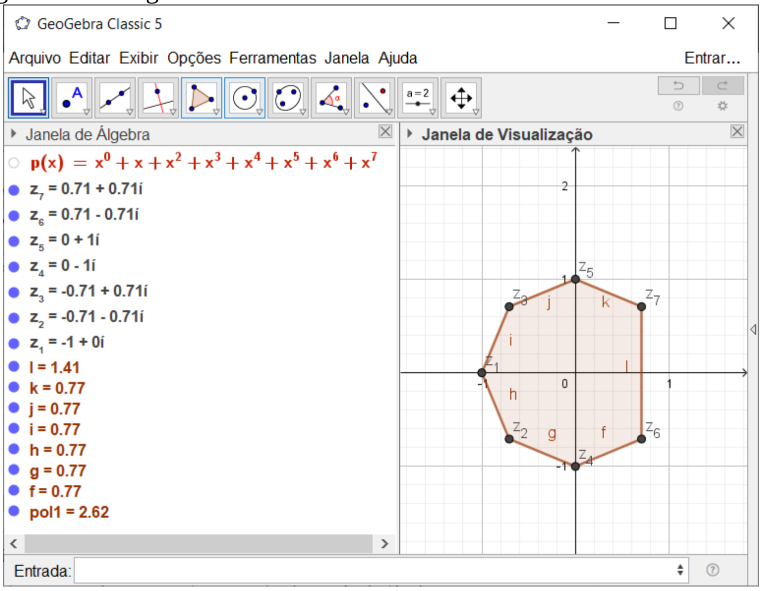
Figura 18 – Pontos gerados com o comando *RaízesComplexas*



Fonte: Software GeoGebra.

3. Com a ferramenta *Polígono*, constrói-se o polígono convexo definido pelos pontos encontrados no passo anterior. O GeoGebra nomeará este polígono automaticamente como *pol1*. Por fim, verifica-se a partir da janela de álgebra que a área corresponde a 2,62 unidades quadradas (Figura 19).

Figura 19 – Polígono convexo construído no GeoGebra e o valor de sua área



Fonte: Software GeoGebra.

Como pôde-se perceber, o problema pode ser resolvido com relativa facilidade utilizando os comandos *Soma*, *Sequência* e *RaízesComplexas*, em conjunto com a ferramenta polígono. Seria possível ainda, realizar toda a construção utilizando apenas comandos inseridos no Campo de Entrada. Para isto, bastaria que, no passo 3, utilizássemos o comando *Polígono(z_1, z_2, z_4, z_6, z_7, z_5, z_3)* ao invés da ferramenta polígono. Neste caso, teríamos realizado a construção necessária à solução do problema com três comandos – sendo o primeiro deles, definido como um comando aninhado, por combinar dois comandos diferentes em um único (os comandos *Soma* e *Sequência*).

A partir deste problema surgiu uma dúvida que acabou se tornando um desafio: seria possível resolver este problema com um único comando aninhado inserido no Campo de entrada? Esta dúvida surgiu a partir das interações entre os participantes do curso e levou alguns deles a tentarem chegar a uma solução.

À primeira vista parecia simples: bastava aninhar todos os comandos utilizados em um único. Mas, ao realizar isto de forma intuitiva, percebia-se um problema. Vejamos o que ocorria.

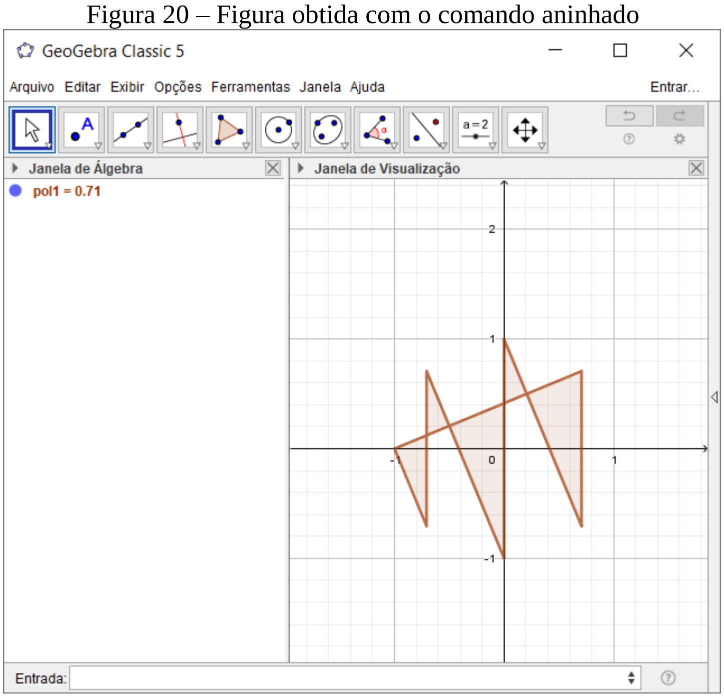
Como a construção que queremos realizar é um polígono, iniciamos o aninhamento com o comando *Polígono*. O comando *Polígono* pode ser implementado de diversas maneiras, sendo uma delas através da identificação de cada um dos pontos que irão compor seus vértices – como sugerido a ser uma alternativa ao passo 3 da resolução inicial – ou através de uma lista de pontos. Vamos implementá-lo utilizando uma lista de pontos com o comando *Polígono(<lista de pontos>)*.

Um comando que gera uma lista de pontos e que foi utilizado na solução apresentada anteriormente é o comando *RaízesComplexas*, que gera uma lista de pontos correspondentes às raízes complexas de um dado polinômio no plano complexo. Aninhando-o ao comando anterior, teremos: *Polígono(RaízesComplexas(<Polinômio>))*.

O polinômio em questão é formado pelo somatório dos elementos *x* elevados à uma variável *i*, de forma que esta variável *i* assume valores inteiros de 0 à 7. Representamos essa soma com o comando aninhado utilizado no passo 1 da primeira solução apresentada. Ficamos então com: *Polígono(RaízesComplexas(Soma(Sequência(x^i, i, 0, 7))))*.

Com isto, finalizamos o comando e, ao confirmarmos sua inserção no Campo de Entrada (apertando a tecla Enter após digitá-lo), obteremos o resultado apresentado na Figura 20.

Figura 20 – Figura obtida com o comando aninhado



Fonte: Software GeoGebra.

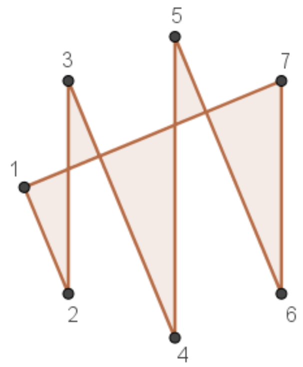
Ao observarmos a figura, percebemos claramente que o polígono gerado não é o mesmo obtido na construção em 3 passos. E este é o problema. Embora tenhamos conseguido realizar uma construção utilizando um único comando aninhado, o resultado foi diferente.

Mas nem tudo fica perdido ao chegarmos neste ponto. Observando o polígono construído é possível perceber que seus vértices se encontram nas mesmas posições em que estavam os pontos que geraram o polígono da solução em 3 passos. Não poderia ser diferente, já que são os pontos correspondentes às raízes complexas do mesmo polinômio.

O problema se encontra na maneira em que os segmentos que formam os lados deste polígono foram construídos. Considerando que os segmentos foram construídos a partir de uma determinada ordem de escolha dos pontos correspondentes às raízes complexas do polinômio, podemos considerar que a ordem estabelecida tomou como parâmetros o valor da abcissa e da ordenada destes pontos, sendo a abcissa o parâmetro primário e a ordenada o parâmetro secundário. Isto pois, observando a Figura 20 e considerando (-1, 0) como ponto de partida, percebemos que os pontos seguintes (da esquerda para a direita) seguem uma ordem crescente nos valores da abcissa e ordenada (para as ordenadas, quando possuem valores de abcissa iguais).

A partir disto, para solucionarmos o problema da construção “incorreta” do polígono, podemos pensar em alterar a ordem em que estes pontos serão escolhidos. Vejamos como fazer isto. Inicialmente, associemos um número correspondente à ordem de escolha dos pontos a cada vértice do polígono (Figura 21):

Figura 21 – Vértices numerados de acordo com a ordem inicial de escolha dos pontos

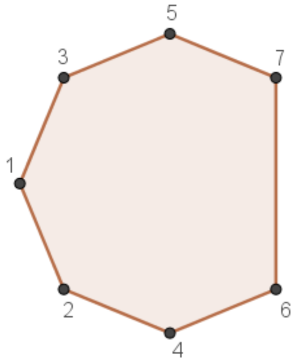


Fonte: Software GeoGebra

Os números associados aos pontos, na ordem em que foram escolhidos, podem ser associados à sequência $a_n = \{1, 2, 3, 4, 5, 6, 7\}, n \in \mathbb{N}^*$, em que cada elemento da sequência corresponde à sua posição nela. Assim, a lei de formação desta sequência (ou termo geral), pode ser dado por $a_n = n$.

Tendo isto em mente, podemos pensar em alterar esta lei de formação para outra que defina uma nova ordem de escolha para os vértices do polígono. Desejamos que o polígono formado seja da seguinte forma (Figura 22):

Figura 22 – Vértices numerados de acordo com a ordem desejada



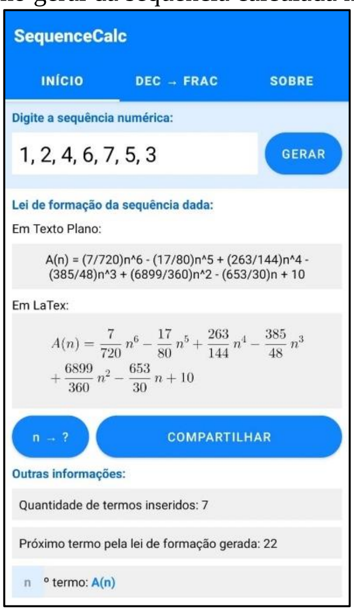
Fonte: Software GeoGebra

Sendo assim, a sequência de pontos escolhida deve ser a seguinte:

$$a_n = \{1, 2, 4, 6, 7, 5, 3\}, n \in \mathbb{N}^*$$

E, para que possamos modificar a ordem de escolha dos vértices do polígono para corresponder à esta sequência, precisaremos de uma expressão de termo geral para essa sequência. Poderíamos tentar encontrar uma expressão satisfatória mentalmente, por tentativas ou utilizando procedimentos matemáticos como a interpolação polinomial manualmente, mas certamente seria um procedimento trabalhoso, considerando que essa sequência não se classifica como progressões aritméticas ou geométricas simples. Considerando isso, o SequenceCalc se torna muito útil, já que podemos simplesmente inserir nele a sequência que queremos e ele calculará uma expressão de termo geral para ela. Fazendo isso, obtemos o resultado apresentado na Figura 23:

Figura 23 – Termo geral da sequência calculada no SequenceCalc



Fonte: Captura de tela do aplicativo SequenceCalc.

Dessa forma, a expressão do termo geral da sequência a_n definida anteriormente será:

$$a_n = \frac{7}{720}n^6 - \frac{17}{80}n^5 + \frac{263}{144}n^4 - \frac{385}{48}n^3 + \frac{6899}{360}n^2 - \frac{653}{30}n + 10$$

Precisamos agora aplicar convenientemente esta lei de formação ao problema e confirmar a solução. Comecemos analisando o comando aninhado que gerou o polígono incorreto (não convexo): *Polígono(RaízesComplexas(Soma(Sequência(x^i, i, 0, 7))))*.

Da mesma forma que nele, devemos, para construir o polígono correto, iniciar o comando aninhado com o comando *Polígono*, por ser a construção que queremos que o GeoGebra nos apresente. O comando *Polígono* requer uma lista de pontos como parâmetro e para o polígono incorreto, essa lista de pontos foi estabelecida utilizando o comando *RaízesComplexas(<Polinômio>)*. No entanto, para construirmos o polígono correto essa abordagem não será efetiva. Ao utilizarmos o comando *RaízesComplexas* o GeoGebra ordena automaticamente os pontos que serão ligados para formar os lados do polígono, seguindo a regra explicada anteriormente. O que precisará ser feito será a criação de uma sequência de pontos com abcissas iguais às partes reais das raízes complexas do polinômio *p(x)* e ordenadas iguais às partes imaginárias destas raízes. Desta forma será possível definir uma ordem de escolha dos vértices do polígono utilizando a lei de formação definida pelo SequenceCalc. Vejamos, passo-a-passo, como isto pode ser feito:

1. Iniciamos o comando aninhado com o comando polígono:
   Polígono(<Lista de Pontos>)

2. A lista de pontos será dada por uma sequência de pontos, então aninharemos ao comando *Polígono* o comando *Sequência*:
   Polígono(Sequência(<Expressão>, <Variável>, <Valor Inicial>, <Valor Final>))

3. Os elementos dessa sequência serão os pontos que formarão o polígono, cujas abcissas e ordenadas serão dadas pelas partes real e imaginária das raízes complexas do polinômio *p(x)*. Para construirmos pontos no GeoGebra através do Campo de Entrada, basta que digitemos as coordenadas do ponto na forma de um par ordenado. Portanto, o parâmetro *<Expressão>* solicitado pelo comando *Sequência* deve estar na forma de um par ordenado. Inicialmente o representaremos apenas como "(*x*, *y*)" e posteriormente substituiremos estes valores. Precisaremos ter 7 pontos identificados conforme definido na Figura 22, ou seja, numerados de 1 a 7. Atribuiremos, então, aos parâmetros *<Valor Inicial>* e *<Valor Final>* os números 1 e 7. O parâmetro *<Variável>* será utilizado na

expressão e assumirá na expressão os valores definidos pelos parâmetros <*Valor Inicial*> e <*Valor Final*>. Definiremos como *n* o identificador desta variável. Assim, o comando aninhado ficará da seguinte forma:

Polígono(Sequência((x, y), n, 1, 7))

4. Vamos identificar agora quais comandos substituirão os valores de *x* e de *y*. *x* deve ser a parte real e *y* a parte imaginária das raízes complexas de *p(x)*. O GeoGebra possui os comandos *ParteReal* e *ParteImaginária*, que podemos utilizar neste momento:

   Polígono(Sequência((ParteReal(<x>), ParteImaginária(<x>)), n, 1, 7))

5. Precisamos agora definir os parâmetros dos comandos *ParteReal* e *ParteImaginária*, ou seja, de quais números complexos extrairemos as partes real e imaginária. Utilizaremos para isto, em ambos os parâmetros, o comando *RaízesComplexas* aninhado com os comandos *Soma* e *Sequência*, que definem o polinômio *p(x)*. Algo a se atentar é que precisaremos dar entrada a esses comandos entre chaves ({}), para que as raízes complexas sejam apresentadas na forma de uma lista:

   Polígono(Sequência((ParteReal({RaízesComplexas(Soma(Sequência(x^i, i, 0, 7)))}), parteImaginária({RaízesComplexas(Soma(Sequência(x^i, i, 0, 7)))})), n, 1, 7))

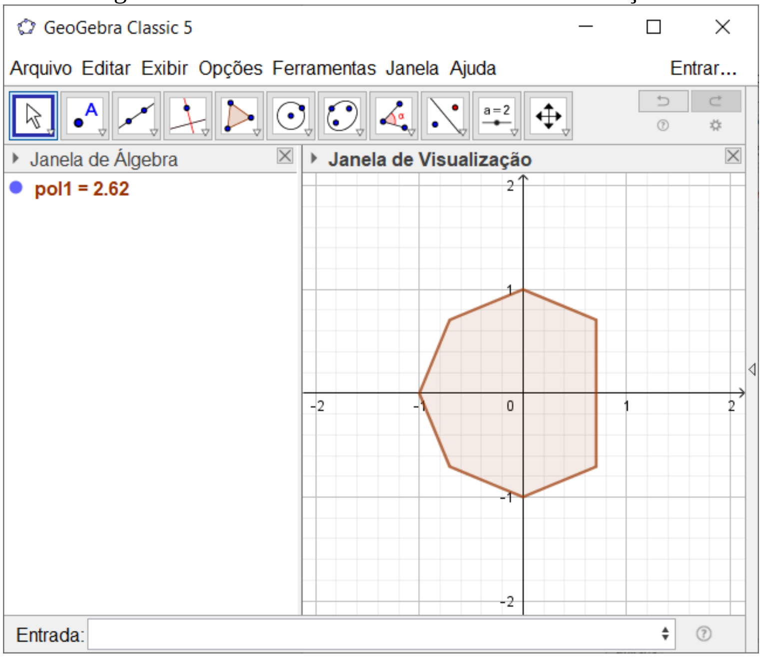
6. Para finalizar, precisamos definir a ordem em que os pontos da lista de pontos que formarão o polígono serão escolhidos, aplicando a lei de formação $?_{?}$ encontrada com o SequenceCalc. Algo a se observar é que aparentemente o comando aninhado está finalizado e poderíamos esperar que o polígono “incorreto” fosse construído na janela de visualização do GeoGebra ao executá-lo. Porém, ao tentarmos executar o comando, isso não ocorre porque o comando *RaízesComplexas* gera uma lista de números complexos e os comandos *ParteReal* e *parteImaginária* precisam utilizar um elemento da lista por vez para formar os pares ordenados que irão compor os pontos correspondentes ao vértice do polígono. Precisamos, então, definir qual elemento da lista de números complexos gerada pelo comando *RaízesComplexas* nós iremos utilizar por vez. Para isto, podemos fazer uso do comando *Elemento*, que possui dois parâmetros: uma lista e uma posição na lista. Tanto para a abcissa do ponto, quanto para a ordenada, o parâmetro lista será dada pela lista das raízes complexas (entre chaves), e a posição na lista será dada, inicialmente, pela variável *n* definida para no passo 3. O comando aninhado ficará, então, da seguinte forma:

Polígono(Sequência((ParteReal(Elemento({RaízesComplexas(Soma(Sequência(x^i, i, 0, 7)))}, n)), parteImaginária(Elemento({RaízesComplexas(Soma(Sequência(x^i, i, 0, 7)))}, n))), n, 1, 7))

7. Ao executarmos o comando do passo anterior no GeoGebra, obteremos o mesmo polígono mostrado na Figura 20, porém com a diferença primordial de que, com o este comando, podemos controlar a ordem em que os pontos que formarão o polígono serão escolhidos, através da utilização do comando *Elemento*. No momento a ordem de escolha dos pontos segue o parâmetro com o valor "*n*" do comando *Elemento* e, desta forma, o polígono é construído tomando os pontos numerados de 1 a 7, em ordem. Como queremos que a ordem de escolha dos pontos seja dada pela sequência $?_{?} = \{1, 2, 4, 6, 7, 5, 3\}$, devemos trocar o parâmetro "*n*" pela lei de formação da sequência $?_{?}$. Pode-se utilizar o recurso "Compartilhar" no SequenceCalc para enviar a expressão do termo geral da sequência em "texto plano" para um e-mail ou algum aplicativo de mensagens que permita o acesso através do smartphone e do computador onde se esteja utilizando o GeoGebra e copiá-la no comando aninhado. Fazendo isso, teremos:
   Polígono(Sequência((ParteReal(Elemento({RaízesComplexas(Soma(Sequência(x^i, i, 0, 7)))}, (7/720)n^6 - (17/80)n^5 + (263/144)n^4 - (385/48)n^3 + (6899/360)n^2 - (653/30)n + 10)), parteImaginária(Elemento({RaízesComplexas(Soma(Sequência(x^i, i, 0, 7)))}, (7/720)n^6 - (17/80)n^5 + (263/144)n^4 - (385/48)n^3 + (6899/360)n^2 - (653/30)n + 10))), n, 1, 7))

   E executando este comando no GeoGebra obteremos, finalmente, o polígono correto, como mostrado na Figura 24.

Figura 24 – Problema 2: resultado final da construção.



Fonte: Software GeoGebra.

É importante salientar que o problema aqui tratado foi o de construir o polígono convexo formado pelas raízes complexas do polinômio enunciado com um único comando aninhado no Geogebra. Esse foi o desafio: utilizar um único comando. Caso isso seja desconsiderado, o uso do SequenceCalc nessa situação não faria sentido. Salientamos isso pois, apesar de o software GeoGebra ter sido o ambiente de resolução do problema, não se deve desprezar o papel fundamental que o SequenceCalc teve em permitir que o problema fosse solucionado – através da expressão do termo geral que alterou a ordem de escolha dos pontos no plano complexo para a correta construção do polígono desejado.

5 CONSIDERAÇÕES FINAIS

As sequências numéricas estão presentes em várias situações do cotidiano e, da mesma forma, em diversos problemas matemáticos. Algo intrínseco a essas sequências é o padrão que as definem, nem sempre identificado de imediato. Esse padrão é comumente expresso através de uma expressão algébrica denominada “termo geral” ou “lei de formação”.

Técnicas matemáticas como a interpolação polinomial, possibilitam a identificação de uma expressão de termo geral para qualquer sequência numérica finita, até mesmo para as que não aparentam ter padrão nenhum. No entanto, aplicar essa técnica manualmente pode ser uma tarefa bem árdua, por exigir cálculos cada vez mais extensos conforme aumenta-se a quantidade de termos da sequência a qual se deseja obter o termo geral. Para contornar esse problema, levantou-se a possibilidade de uso de softwares de computador para realização de cálculos e, considerando a grande presença de aparelhos smartphones Android na vida das pessoas, junto com a falta de um aplicativo para estes dispositivos que possuíam a funcionalidade proposta, decidiu-se por desenvolvê-lo para essa plataforma.

A fundamentação teórica do trabalho englobou a apresentação de conceitos e definições de sequências numéricas, tipos de sequência, termo geral e interpolação polinomial e baseou-se em autores como Lima (2006), Iezzi e Hazzan (2013), Barroso e Barroso *et al.* (1987) e Freitas, Corrêa e Vaz (2019). A partir das definições apresentadas, exemplificou-se o cálculo manual da interpolação polinomial de uma sequência com 5 termos, destacando o quão extenso esse cálculo poderia ser e, com isso, corroborando com a proposta de automatizá-lo utilizando o SequenceCalc.

Os principais pontos do processo de desenvolvimento do aplicativo também foram apresentados: as ferramentas utilizadas, a lógica por trás de seu funcionamento e a implementação dos elementos visuais. Posteriormente, elencaram-se dois problemas para exemplificar as vantagens do uso do SequenceCalc.

No primeiro problema, a possibilidade de uso do aplicativo se fez bem clara, tendo bem definido em seu enunciado a necessidade de se descobrir o próximo termo de uma sequência – problema facilmente contornado caso saibamos sua expressão do termo geral. Foi perceptível o quão vantajoso o uso do SequenceCalc poderia ser na solução de problemas desse tipo, apresentando, de imediato, tanto a lei de formação quando o próximo termo da sequência solicitada no enunciado, bastando apenas que a sequência em questão fosse informada.

No segundo problema, a possiblidade de uso do aplicativo não era identificada de início. Não estava explícito no enunciado dele a necessidade ou a possibilidade da utilização do termo geral de uma sequência. A partir disso, percebe-se facilmente a diferença de natureza entre este problema e o anterior, de forma que, neste, o SequenceCalc atua como uma ferramenta auxiliar, porém crucial para a solução do problema.

Este trabalho mostrou, portanto, que o SequenceCalc é uma ferramenta útil para solucionar os problemas que ele se propõe a resolver, mas também pode – de forma às vezes inesperada – auxiliar na solução de outros problemas onde sua funcionalidade não aparente ser relevante.

Pretende-se, futuramente, adicionar a ele novas funcionalidades relacionadas a sequências como: a identificação de tipos de sequência e de algumas sequências específicas conhecidas; a apresentação de mais de uma expressão de termo geral para dadas sequências, considerando que são possíveis uma infinidade de expressões diferentes e que algumas podem ser mais relevantes para a resolução de um problema do que outras; uma calculadora de interpolação polinomial mais ampla, que permita interpolar pontos arbitrários, informados pelo usuário; e uma maneira de exibir o gráfico dos polinômios interpolados no próprio aplicativo.

Apesar destes planos de implementação futura terem sido cogitados, alguns problemas conhecidos da versão atual do aplicativo precisam antes ser corrigidos. Os de maior prioridade são:

- A não definição de limites para as quantidades de termos das sequências e de dígitos dos números que podem ser informados no aplicativo. Isso pode fazer com que o aplicativo demore um tempo excessivo ou mesmo impraticável para concluir algumas operações, causando travamentos. Pretende-se solucionar esse problema a partir da definição destes limites ou de um tempo máximo de espera, depois do qual o aplicativo irá cancelar a tentativa de realizar a operação.
- A falta de uma forma não ambígua de informar dízimas periódicas no aplicativo. Atualmente, representam-se as dízimas repetindo o período por 8 ou mais vezes na digitação, mas o SequenceCalc não consegue identificar quando se deseja ou não que ele interprete o número informado como uma dízima. Para solucionar este problema, pretende-se tratar o uso de reticências imediatamente após a digitação do período da dízima para identifica-las e possibilitar a inserção de frações na forma *a*/*b* nas sequências.
- O aplicativo ainda não se encontra disponível para acesso publicamente. Pretende-se corrigir isso logo que possível, disponibilizando-o através da loja de aplicativos dos

smartphones Android, a Plays Store. No entanto, o instalador da versão atual do aplicativo pode ser acessado via serviço de armazenamento em nuvem particular do autor[14] – somente enquanto não for publicado na Play Store.

Outros problemas menos urgentes são: a falta de uma configuração de cores para a interface do aplicativo quando ativo o "modo escuro" dos smartphones e da configuração de "leitura de tela" para o uso por pessoas com problemas de visão. Esses são pontos que também pretende-se dar atenção, mas, neste momento, com menos prioridade.

Considerando tudo isso, elenca-se que o tema deste trabalho – o aplicativo SequenceCalc para cálculo do termo geral de uma sequência finita de números racionais – ainda tem muito a ser explorado e, a partir de seu desenvolvimento, pode trazer muitas outras contribuições além das apresentadas neste trabalho.

[14] Disponível em: https://1drv.ms/u/s!AgNj_9AISG_Kg5MOwNb64Of2533dEg?e=xgRhiS. Acesso em: 19 nov. 2021.

REFERÊNCIAS

ANDROID DEVELOPERS. Conheça o Android Studio. **Android Developers**, 2021. Disponível em: https://developer.android.com/studio/intro. Acesso em: 29 Março 2021.

BARROSO, Leônidas C. *et al.* **Cálculo Numérico (com aplicações)**. 2ª. ed. São Paulo: Harbra Ltda., 1987.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**, Brasília, 2018.

FREITAS, Rafael D. O.; CORRÊA, Rejane Izabel L.; VAZ, Patrícia M. S. **Cálculo Numérico**. Porto Alegre: SAGAH, 2019. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788595029453/. Acesso em: 19 nov. 2021.

GUIDINI, Priscila. O smartphone como nova mídia em uma sociedade conectada. **R. Dito Efeito**, Curitiba, v. 8, n. 12, p. 33-47, jan./jun. 2017. ISSN 1984-2376. Disponível em: https://periodicos.utfpr.edu.br/de/article/view/7041. Acesso em: 11 nov. 2021.

GUIDORIZZI, Hamilton L. **Um curso de cálculo:** volume 1. 6ª. ed. Rio de Janeiro: LTC, v. I, 2018. 608 p. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788521635574. Acesso em: 10 nov. 2021.

IEZZI, Gelson; HAZZAN, Samuel. **Fundamentos de Matemática Elementar:** Sequências, Matrizes, Determinantes e Sistemas. 8. ed. São Paulo: Atual, v. 4, 2013. 282 p.

LIMA, Elon L. **Análise Real volume 1:** Funções de Uma Variável. 8. ed. Rio de Janeiro: IMPA, 2006.

NOBRE, José Filho F.; ROCHA, Rogério A. Progressões Aritméticas de Ordem Superior. **Professor de Matemática Online**, Rio de Janeiro, v. 6, n. 1, p. 35-49, 2018. ISSN 2319-023X. Disponível em: http://pmo.sbm.org.br/wp-content/uploads/sites/16/dlm_uploads/2019/03/art3_vol6_2018_SBM_PMO-1.pdf. Acesso em: 12 jul. 2021.

STATCOUNTER. Mobile Operating System Market Share in Brazil - October 2021. **Statcounter:** GlobalStats, 2021. Disponível em: https://gs.statcounter.com/os-market-share/mobile/brazil/#monthly-201901-202110-bar. Acesso em: 11 nov. 2021.

VON ZUR GATHEN, Joachim; GERHARD, Jürgen. **Modern Computer Algebra:** Third Edition. 3ª. ed. New York: Cambridge University Press, 2013.

APÊNDICES

APÊNDICE A – Transcrição do texto do Módulo "Sobre" do Sequencecalc

Sobre

SequenceCalc é um aplicativo para cálculo do termo geral de uma sequência numérica finita de números racionais a partir de interpolação polinomial. Devido a isto, o termo geral obtido será sempre um polinômio de grau máximo igual a 1 número a menos que a quantidade de termos informados. A premissa deste aplicativo é calcular e apresentar uma expressão para o termo geral para qualquer sequência numérica de números racionais, mas não necessariamente a mais simples possível. Ele apresenta, da forma mais simples possível, a lei de formação de qualquer Progressão Aritmética (P. A.), mas não de Progressões Geométricas (P. G.), por exemplo.

Instruções

No Módulo 'Principal', digite a sequência numérica desejada separando os números com vírgula seguida de espaço. Também é possível utilizar apenas vírgulas, apenas espaços ou combinações de ambos os métodos como separador dos números. Caso feito desta forma, o aplicativo substituirá automaticamente as vírgulas e espaços por vírgulas seguidas de espaço. O aplicativo aceita qualquer sequência de números racionais como entrada gerando uma expressão de termo geral polinomial para a mesma. Para números negativos, deve-se utilizar traço indicativo deste sinal imediatamente antes do número em si, sem um espaço em branco separando o sinal do número. No entanto, caso um espaço seja inserido entre o sinal e o número, o aplicativo aplicará uma correção automática, eliminando o espaço existente. Essa correção apenas ocorrerá caso exista um único espaço em branco entre o sinal e o número. Em caso de quantidades maiores de espaço, o aplicativo exibirá uma mensagem de erro na tela, solicitando a correção da sintaxe de entrada da sequência. Para números positivos não se faz necessária a entrada de nenhum sinal em conjunto com o número. Para números com ponto flutuante (números decimais), deve-se utilizar o ponto (.) como separador decimal. Para sequências em que constam dízimas periódicas, a parte decimal do número (após o separador decimal) deve ser composta por 8 ou mais casas decimais para que o aplicativo entenda que se trata de uma dízima periódica. Ainda assim, a depender da dízima o aplicativo ainda poderá considerar que não se trata de uma dízima e realizar uma conversão incorreta para fração e em seguida gerar uma expressão de termo geral não exata. Caso, durante o uso do aplicativo, perceba algum problema causado por esse fato, entre em contato com o desenvolvedor a partir do e-mail informado no contato ao final desta tela, descrevendo o problema. Segue um exemplo de sintaxe de sequência numérica conforme estas instruções:
-1, -0.66666666, -0.33333333, 0, 0.33333333, 0.66666666, 1

O Módulo de 'Conversor' possui duas funcionalidades:
Converter números decimais em fração:
- Insira o número decimal no campo destinado a ele e acione o botão 'Simplificar/Converter'. As regras para inserção do número decimal são as mesmas utilizadas na inserção de números decimais no Módulo 'Principal'. Ao acionar o botão, o aplicativo mostrará na parte superior (espaço cinza, destinado aos resultados) o número decimal convertido em fração, na forma 'numerador/denominador' e também preencherá os campos de entrada 'numerador' e 'denominador' com os respectivos valores convertidos. A fração gerada sempre estará na forma irredutível (mais simplificada possível).

Simplificar frações (Torná-la irredutível):

- Insira o numerador e o denominador nos campos destinados a eles e acione o botão ‘Simplificar/Converter’. Tanto o numerador quanto o denominador devem ser números inteiros (positivos ou negativos) e o denominador deve ser diferente de zero. O Aplicativo impede automaticamente que qualquer outro tipo de entrada (que não sejam números inteiros) seja realizada. Caso se insira o denominador 0, o aplicativo mostrará uma mensagem de erro, aguardando uma entrada diferente de zero. Ao acionar o botão, o aplicativo irá simplificar ao máximo a fração inserida e apresentará o resultado na parte de cima (espaço destinado ao resultado). O aplicativo também calculará o valor decimal da fração e preenchera o campo de entrada ‘Valor Decimal’ com o resultado deste cálculo.

Apesar de possuir estas duas funcionalidades, o aplicativo não permitirá (por não ser possível) que ambas sejam utilizadas simultaneamente. Isso ocorrerá com o aplicativo deixando vazio o campo ‘Valor Decimal’ caso o numerador ou denominador sejam editados ou vice-versa (deixará vazios os campos ‘Numerador’ e ‘Denominador’ caso o campo ‘Valor Decimal’ seja editado). Caso algum dos campos ficar vazio e for necessário, o aplicativo o preencherá com um valor padrão e realizará os cálculos normalmente. Os valores padrão são: ‘0’ para o numerador, ‘1’ para o denominador e ‘0’ para o valor decimal. Estes valores padrão são apresentados nos respectivos campos de entrada, na cor cinza, caso nenhum valor seja inserido.

Créditos

Ideia, programação e design por Lidiomar Casteluber da Silva.

E-mail: lidiomar.c@gmail.com.