

Universidade Federal do Triângulo Mineiro
Faculdade de Matemática
Mestrado Profissional em Matemática



PROFMAT

A LINGUAGEM PYTHON COMO FERRAMENTA NO ENSINO BÁSICO

Brunno de Oliveira Borges

Orientando

Leonardo de Amorim e Silva

Orientador



Uberaba-MG

2021

Brunno de Oliveira Borges

A LINGUAGEM PYTHON COMO FERRAMENTA NO ENSINO BÁSICO

Dissertação apresentada ao Programa de Pós-Graduação em Matemática da Universidade do Triângulo Mineiro, como parte dos requisitos para obtenção do título de **MESTRE EM MATEMÁTICA**.

Área de Concentração: Matemática

Linha de Pesquisa: Ensino de Matemática

Orientador: Prof. Dr. Leonardo de Amorim e Silva

Uberaba-MG

2021

**Catálogo na fonte: Biblioteca da Universidade Federal do
Triângulo Mineiro**

B73I

Borges, Brunno de Oliveira

A linguagem python como ferramenta no ensino básico / Brunno de Oliveira Borges. -- 2021.

69 f. : il., graf., tab.

Dissertação (Mestrado Profissional em Matemática em Rede Nacional) -- Universidade Federal do Triângulo Mineiro, Uberaba, MG, 2021

Orientador: Prof. Dr. Leonardo de Amorim e Silva

1. Matemática. 2. Matemática (Ensino fundamental). 3. Programação (Matemática). 4. Python (Linguagem de programação de computador). I. Silva, Leonardo de Amorim e. II. Universidade Federal do Triângulo Mineiro. III. Título.

CDU 51(07):811.93



Universidade Federal do Triângulo Mineiro

Faculdade de Matemática

Mestrado Profissional em Matemática

R. Frei Paulino, 30 - Nossa Sra. da Abadia

Reitoria, Uberaba-MG, CEP 38025-180

Aluno(a): Brunno de Oliveira Borges

Número de Matrícula: 2018.1027.2

Área de Concentração: Matemática

Linha de Pesquisa: Ensino de Matemática

Pós-Graduação em Matemática: Nível Mestrado

Título da Dissertação: A LINGUAGEM PYTHON COMO FERRAMENTA NO ENSINO BÁSICO

Orientador: Prof. Dr. Leonardo de Amorim e Silva

Esta dissertação foi **APROVADA** em reunião virtual pública realizada na plataforma Google Meet, em 18 de novembro de 2021, às 14h00min, pela seguinte Banca Examinadora:

Uberaba-MG, 18 de novembro de 2021.

BRUNNO DE OLIVEIRA BORGES

A linguagem Python como ferramenta no ensino básico

Dissertação apresentada ao Programa de Pós-Graduação em Mestrado Profissional em Matemática, área de concentração Matemática da Universidade Federal do Triângulo Mineiro como requisito parcial para obtenção do título de mestre

Uberaba 18 de novembro de 2021

Banca Examinadora:

Dr. Leonardo de Amorim e Silva – Orientador
Universidade Federal do Triângulo Mineiro

Dr. Marcelo Ferreira
Universidade Federal do Triângulo Mineiro

Me. Rafaela Ferreira Afonso
Universidade Federal de Rondônia



Documento assinado eletronicamente por **LEONARDO DE AMORIM E SILVA, Professor do Magistério Superior**, em 19/11/2021, às 18:28, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#) e no art. 34 da [Portaria Reitoria/UFTM nº 87, de 17 de agosto de 2021](#).



Documento assinado eletronicamente por **MARCELO FERREIRA, Professor do Magistério Superior**, em 22/11/2021, às 11:29, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#) e no art. 34 da [Portaria Reitoria/UFTM nº 87, de 17 de agosto de 2021](#).



Documento assinado eletronicamente por **Rafaela Ferreira Afonso, Usuário Externo**, em 23/11/2021, às 09:59, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#) e no art. 34 da [Portaria Reitoria/UFTM nº 87, de 17 de agosto de 2021](#).



A autenticidade deste documento pode ser conferida no site http://sei.uftm.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0635768** e o código CRC **008A950B**.

Dedicatória

Dedico as conquistas deste trabalho aos meus pais JOSÉ DE ANCHIETA BORGES e SANDRA NÚBIA DE OLIVEIRA BORGES, que sempre me ensinaram sobre coragem e dedicação com exemplos diários durante minha educação, duas riquezas necessárias neste caminho de estudos. A minha esposa JÉSSICA CRISTINA SILVA BORGES, que ocupa um papel fundamental em minha vida e que me apoia em todos os momentos. Aos meus avós EDSON DE OLIVEIRA e VILMA APARECIDA DE OLIVEIRA, saibam que a sabedoria de vocês ilumina a cabeça de todos os membros da nossa família, os seus carinhos confortam os nossos corações diariamente.

Agradecimentos

Agradeço a Deus por me guiar, iluminar e dar-me tranquilidade e serenidade para seguir meu caminho e dar sequências aos meus objetivos, não desanimando com as dificuldades encontradas. Agradeço a Ele também por manter meu pai e minha mãe ao meu lado, com saúde.

Ao meu pai José de Anchieta Borges, por confiar em mim, pelo exemplo de coragem, determinação e amor, que ajudaram muito a definir o meu caráter, e cujas lembranças estarão sempre em minha memória.

A minha mãe Sandra Núbia de Oliveira Borges, pelo carinho, pela ternura, dedicação e amor, que sempre me apoiou e amparou-me quando se fazia necessário.

Aos meus pais, que, muitas vezes, renunciaram aos seus sonhos para que eu pudesse realizar o meu, partilho a alegria deste momento.

A minha esposa Jéssica Cristina Silva Borges, por todo o amor, carinho e compreensão, que muito me incentiva e apoia, que é a luz que me ilumina durante todos os momentos, conduzindo-me da tristeza para a felicidade.

Aos meus avós Edson de Oliveira e Vilma Aparecida de Oliveira, pelo amor incondicional eterno, por terem me ensinado a ser nobre, na essência da palavra, pela garra, perseverança e otimismo contagiantes.

A minha irmã Mariana Oliveira Borges, pelo amor incondicional, sempre. A distância não nos separa. Nossos corações estarão conectados sempre.

Aos meus colegas do Instituto Teresa Valsé, pelos momentos de troca de conhecimento, pela amizade e pela solicitude. Gratidão eterna.

Ao meu orientador Leonardo de Amorim e Silva pela confiança em meu trabalho, pelo respeito, por me ensinar, pela paciência e por sempre estar disposto em me ajudar. Gratidão pelo compartilhamento e trocas. Elas foram fundamentais para o resultado final dessa pesquisa.

“Alguns homens veem as coisas como são, e dizem “Por quê?”. Eu sonho com as coisas que nunca foram e digo “Por que não?”.”

—Geroge Bernard Shaw

Resumo

Durante a pandemia, o ambiente escolar se deparou com a necessidade da utilização de ferramentas tecnológicas para o desenvolvimento de atividades de ensino.

A atividade tecnológica tem interfaces aos cotidianos acadêmicos e profissionais da atualidade. Na docência, verifica-se a necessidade de conectar a programação com o ensino de matemática, com o objetivo de potencializar o aprendizado de certos assuntos matemáticos.

Corroborar-se com (RESNICK, 2013) em sua afirmação de que a programação é a forma de expressão deste século e que o ato de programar é um conjunto de ações que formulamos de forma antecipada para atingir certo objetivo, características que são facilmente encontradas nas aulas de matemática do ensino básico durante o desenvolvimento de conteúdos e resolução de exercícios.

A linguagem *Python* (ROSSUM; JR, 1995) foi selecionada para o desenvolvimento das atividades propostas pelo fato de ser uma linguagem de simples entendimento, possuir comandos semelhantes ao inglês, ser dinâmica e flexível, e a IDE (Interface de desenvolvimento) utilizada para a elaboração dos códigos é a *PyCharm*.

Um dos objetivos desta pesquisa é realizar uma discussão sobre os benefícios do aprendizado da programação no ensino básico e também propor atividades que envolvam programação e conteúdos matemáticos, oportunizando a alunos e professores descobertas e caminhos diferentes de uma aula tradicional.

Palavras-chave: Programação; Ensino Básico; Python e Matemática.

Abstract

During the pandemic, the school environment was faced with the need to use technological tools for the development of teaching activities.

Technological activity has interfaces to current academic and professional daily life. In teaching, there is a need to connect programming with mathematics teaching, in order to enhance the learning of certain mathematical subjects.

During the pandemic, the school environment was faced with the need to use technological tools for the development of teaching activities.

He corroborates (RESNICK, 2013) in his statement that programming is the form of expression of this century and that the act of programming is a set of actions that we formulate in advance to achieve a certain objective, characteristics that are easily found in elementary school math classes during content development and problem solving.

The language *Python* (ROSSUM; JR, 1995) was selected for the development of the proposed activities because it is a simple to understand language, has commands similar to English, is dynamic and flexible, and the IDE (Development Interface) used for the elaboration of the codes is the *PyCharm*.

One of the objectives of this research is to discuss the benefits of learning programming in basic education and also to propose activities that involve programming and mathematical content, providing students and teachers with discoveries and paths that are different from a traditional class.

Keywords: Programming; Basic education; Python and Mathematics.

Sumário

INTRODUÇÃO	2
1 Educação e Tecnologia	7
1.1 Tempos de Pandemia	7
1.2 Educação Matemática no Brasil	10
1.3 A Importância da Tecnologia na Educação	11
2 Programação no Ensino e Aprendizagem	16
2.1 Programação e seus Benefícios	16
2.2 A Linguagem Python	20
3 Aplicações	26
3.1 Matrizes	26
3.2 Equações do 2º Grau	50
3.3 Teorema de Pitágoras	54
3.4 Matemática Financeira	58
4 Considerações Finais	66

Lista de Figuras

2.1 Top 10 das linguagens de programação mais utilizadas - IEEE Spectrum 2019.	20
2.2 Índice Tiobe - Frequência de Pesquisas	21
2.3 Ranking do Github das linguagens mais utilizadas no mundo. . .	21

Introdução

A afinidade com tecnologia sempre foi presente em minha vida, desde criança com videogames e computadores. Na minha adolescência presenciei várias evoluções tecnológicas como o celular *touchscreen* e o desenvolvimento de poderosos computadores e videogames.

Em meu percurso como graduando do curso de Matemática da Universidade Federal de Uberlândia - UFU, localizada no sudeste de Minas Gerais, tive a oportunidade de desenvolver projetos com robótica educacional como bolsista PIBID - (Programa Institucional de Bolsas de Iniciação à Docência) que tem por objetivo antecipar o vínculo entre os futuros professores e as salas de aula da rede pública, fazendo uma articulação entre a educação superior (por meio das licenciaturas), a escola e os sistemas estaduais e municipais. Outro objetivo deste programa é incentivar a carreira do magistério nas áreas da educação básica que possuem maior carência de professores com formação específica: ciências e matemática do ensino fundamental; física, química, biologia e matemática para o ensino médio.

Ao participar de competições de robótica, foi possível perceber um aumento no interesse em programação e, conseqüentemente, a necessidade de guiar os alunos, tanto de ensino fundamental e de ensino médio, nesse processo. Sendo assim, durante as reuniões de desenvolvimento das estratégias em equipe, foi necessário abordar as discussões sobre a otimização de certos algoritmos desenvolvidos na interface de programação.

O meu interesse pelo assunto cresceu e a busca por novos horizontes foi inevitável. Após concluir a graduação e obter experiência com alunos em sala de aula, foi notório o desenvolvimento concomitante da tecnologia em nosso dia a dia, sendo isso possível graças às linguagens de programação.

Toda essa evolução é fundamentada na matemática, e com o objetivo de aprimorar meu currículo, participei do processo seletivo do Mestrado Profissional em Matemática em Rede Nacional – PROFMAT. Cursando a disciplina MA 36 – Recursos Computacionais no Ensino de Matemática, deparei-me com a linguagem de programação Python (ROSSUM; JR, 1995), que é a ferramenta que utilizamos para desenvolver as atividades propostas neste trabalho.

Unindo este contexto com a experiência como professor de matemática do ensino fundamental e médio, verifico a necessidade da elaboração de atividades que abordam conceitos matemáticos e faça a utilização da programação como ferramenta facilitadora e que potencializa o processo ensino-aprendizagem.

Transformações significativas estão ocorrendo na sociedade atual, o que reverbera uma nova cultura, modo de interpretação, de comunicação, e aprendizagem. Teóricos como (HILTON, 1964); (HELVEY, 1971); (LEVY, 1999) nomeiam essa realidade como cibercultura, forças de tradição de mudança, que atuam direta e indiretamente nos diversos processos educacionais.

"As redes sociais, portais e blogues, os videojogos, chats e sites de todo tipo, os sistemas de troca de mensagens e o comércio eletrônico, o cinema, rádio, música e televisão interativos via internet são, realmente, apenas algumas das expressões que surgem nesse âmbito e estão ajudando a estruturar praticamente a cibercultura."(RÜDIGER, 2013)

No contexto escolar, a disseminação das Tecnologias de Informação e Comunicação (TIC's) na vida cotidiana é um dos acontecimentos que advém desse processo.

Em nossa sociedade atual, é cada vez mais constante o uso das tecnologias, pois os estudantes estão a todo momento em contato com diversos recursos tecnológicos: celulares, computadores, tablets, dentre outros. Compreendemos que grande parte deles já colocam em ação seu processo de autoria digital, mas muitas vezes vivenciam uma realidade oposta no ambiente escolar. D'Ambrosio destaca que:

O jovem inserido em um ambiente, cada vez mais permeado com as novas tecnologias, encontra pela frente, nos diversos setores da sociedade, desafios e situações que exigem pensamento divergente e criativo. O jovem sabe que aprende muito mais fora da escola. Sabe que há uma nova prática para a aquisição do conhecimento. A escola está descompassada. (D'AMBROSIO, 1998).

Quer reconhecamos ou não, "as escolas e todos os espaços formais de educação são muito lentos na incorporação crítica de práticas que já fazem parte da cultura extraescolar de usos dos meios para a comunicação, interação e o trabalho em redes"(KANSKI, 2004).

Sobre isso, (SAVIANI, 2007) também destaca ser necessário introduzir as novas tecnologias nas atividades de sala de aula para que o ensino acompanhe as transformações e os estudantes adquiram os instrumentos essenciais para uma comunicação intersubjetiva.

Baseado neste fato, a proposta desta dissertação é apresentar uma dinâmica pedagógica de conceitos matemáticos e utilizar a linguagem de programação Python como ferramenta potencializadora do ensino destes conceitos.

Python é uma linguagem de programação de alto nível, interpretada e multi-paradigma. Lançada por Guido van Rossum no ano de 1991 é, atualmente, mantida de forma colaborativa e aberta. Muitas empresas utilizam essa tecnologia em seus ambientes virtuais e físicos. Há diversas aplicações para Python no campo profissional, das quais três merecem destaque nesta pesquisa:

→ *Desenvolvimento web*: Web frameworks (estruturas de web) como, Django¹ e Flask², são baseadas em Python, tornaram-se muito populares para desenvolvimento web. Esses frameworks ajudam o desenvolvedor de web a construir o lado back-end do seu código, é a parte responsável por conversar com o servidor.

→ *Data Science, machine learning e análise de dados*: Data Science (ciência de dados) é um campo interdisciplinar que usa métodos, pro-

¹Site: <https://www.djangoproject.com/>

²Site: <https://flask.palletsprojects.com>

cessos, algoritmos e sistemas científicos para extrair conhecimento e percepções de muitos dados estruturais e não estruturados. A ciência de dados está relacionada à mineração de dados, machine learning (aprendizagem de máquina) e big data.

→ *Scripting*: Geralmente, refere-se a escrever pequenos programas projetados para automatizar tarefas. A linguagem Python é adequada para esse tipo de tarefa, principalmente porque tem sintaxe relativamente simples e fácil de escrever. Outro ponto positivo é sua velocidade, ou seja, é rápido escrever algo pequeno e testá-lo.

A linguagem Python possui outras aplicações, por exemplo: aplicativos incorporados, jogos e programas de *desktop*. Diariamente, são criados novos aplicativos em dispositivos eletrônicos e softwares que auxiliam discentes e docentes no processo de ensino-aprendizagem, em especial na disciplina de Matemática.

Nesses termos, a questão principal que orientou a pesquisa foi: a utilização de Python potencializa o processo de ensino-aprendizagem nas aulas de matemática, em especial no ensino básico?

A partir desse problema, surgem outros questionamentos: a linguagem Python pode ser utilizada como ferramenta de ensino aprendizagem nas aulas de matemática? Essa linguagem pode ser instrumento de aprendizagem e quebra de paradigmas no espaço sala de aula? É possível propor atividades que realizam a releitura de problemas matemáticos a partir da linguagem Python?

O objetivo geral dessa dissertação consiste em propor atividades que realizam a releitura de problemas matemáticos, em especial do ensino básico, utilizando a linguagem Python para desenvolver o raciocínio lógico estruturado dos alunos e ao mesmo tempo motivá-los com o desenvolvimento de códigos.

Neste estudo, será utilizada a IDE PyCharm³, que além de ser uma interface visualmente agradável e personalizável, é ideal para aprendizagem

³Site: <https://www.jetbrains.com/pt-br/pycharm/>

com Python. A *PyCharm* conta com funções internas de Python para que possamos compilar um programa rapidamente e sem erros, o que é uma vantagem para o campo educacional, pois evita eventuais problemas técnicos que impedem otimizar o tempo de duração de uma aula.

A fim de alcançar os objetivos e colaborar com estudos que envolvam temas relacionados ao desenvolvimento de docentes que atuam no ensino fundamental e médio por meio de atividades e rotinas, a dissertação está estruturada em quatro capítulos.

No primeiro capítulo, *Educação e Tecnologia* apresenta as transformações e os impactos gerados em nosso cotidiano educacional pela constante transformação da tecnologia, a importância de conhecer uma linguagem de programação.

No segundo capítulo, *Programação no Ensino Aprendizagem*, é destacada a importância da programação no ambiente escolar.

O terceiro capítulo, intitulado *Aplicações*, versa sobre definições de diversos conteúdos do ensino básico (fórmula resolutiva de equações do 2º grau - Fórmula de Bhaskara, teorema de Pitágoras, matrizes, matemática financeira), implementações computacionais sobre cada uma dessas definições. Apresenta uma proposta de atividade que pode ser desenvolvida em sala de aula.

Para finalizar, o texto apresenta o quarto capítulo, denominado *Conclusões e Considerações finais*. Nele, apresenta-se a expectativa acerca da utilização de programação durante as aulas de matemática, em especial sobre a programação na linguagem Python, sobre os supostos pontos positivos e como esta atividade pode incitar uma mudança nos ritos tradicionais da educação matemática.

Capítulo 1

Educação e Tecnologia

"A revolução cibernética exige uma reestruturação dos programas e processos educacionais, porque só "os seres humanos que aprenderem a usar a máquina com sabedoria serão por ela liberados para alcançar a sua excelência" (HILTON, 1964)

1.1 Tempos de Pandemia

Em fevereiro de 2020, acontecia no Brasil as primeiras ações ligadas à pandemia do coronavírus, e após quinze dias, o país registrou a primeira contaminação.

Todos os setores iniciaram um processo de discussão sobre desenvolver um plano de adaptação ao novo cenário mundial, o que demandou ações urgentes e efetivas.

A educação no Brasil precisaria ser reestruturada para se adaptar aos obstáculos impostos pela pandemia. Finalmente, a tecnologia foi reconhecida como ferramenta essencial no processo de aprendizagem. A rotina mudou rapidamente, professores, que eram acostumados a desenvolver suas aulas no quadro branco, viram a necessidade de substituir seus pincéis por uma mesa digitalizadora (e outros recursos), tornaram-se usuários de *softwares* educacionais e plataformas de vídeo chamadas. Mozart Neves Ramos, membro do Conselho Nacional de Educação (CNE), destaca:

“Uma das coisas mais bonitas que vejo, mesmo neste cenário tão triste, é a capacidade que muitos professores tiveram de se reinventar. Assim como ninguém no mundo inteiro, eles não estavam preparados, mas notamos esse esforço para transformar as aulas presenciais no ensino remoto”.(RAMOS, 2020)

Como ações que podem dar suporte ao professor durante este momento de aulas remotas, a gerente de pesquisa e desenvolvimento do Itaú Social, Patrícia Mota Guedes, afirma:

"O ensino remoto sozinho não dá conta. O professor tem um papel fundamental, portanto precisa de um apoio cada vez maior. Isso, não só no sentido de formação, mas como também de valorização e recursos completos para trabalhar" (GUEDES, 2020)

Com a suspensão das aulas presenciais, os alunos intensificaram o desenvolvimento do conhecimento por meio de seus computadores, celulares e *tablets* e, com o passar do tempo, ressurgiu a discussão sobre o ensino híbrido, já previsto da BNCC¹. Vale ressaltar que o ensino remoto explicitou a desigualdade social no Brasil e a dificuldade de acesso à tecnologia. O economista e superintendente executivo do Instituto Unibanco, Ricardo Henrique, discorre sobre a suspensão das aulas presenciais:

“A suspensão das atividades escolares mostra também a grande diferença de acesso dos estudantes de escolas públicas a condições não presenciais, desde infraestrutura do domicílio, quantidade de pessoas por cômodos, até a disponibilidade de equipamentos e conectividade. Isso tende a gerar um aumento da desigualdade”.(HENRIQUES, 2020)

Neste período de mudanças, sabemos que existem alunos que são totalmente dependentes de aulas tradicionais, possuem dificuldade ao acesso a internet e não possuem equipamentos adequados para o ensino remoto. Neste sentido a diretora do Instituto Crescer e Doutora em Educação pela Universidade de São Paulo Luciana Allan, afirma:

¹BNCC - Base Nacional Comum Curricular

"Não podemos esperar que todos se adaptem repentinamente a estes novos tempos. Sabemos dos inúmeros problemas de conexão à Internet, mas é um ótimo momento para nos reinventarmos e criarmos coragem de testar o uso de ferramentas tecnológicas já disponíveis para estruturarmos alternativas no formato de educação a distância. Pensar fora da caixa pode ajudar a mitigar o problema momentâneo, como também colaborar para fortalecer a cultura digital e avançar rumo à uma nova educação, como há bastante tempo já temos discutido."(ALLAN, 2020)

As escolas, em sua grande maioria, não investiram em infraestrutura, treinamento e suporte necessários, assim deixando de criar a oportunidade de que os professores e alunos usufruam da tecnologia educacional moderna.

A pandemia é um momento terrível em que a população mundial está submetida, mas voltando os olhares para o campo educacional, é fato que ela revelou aspectos positivos e inesperados.

Para escolas e professores a pandemia causou melhorias rápidas com relação ao uso das tecnologias, observou-se a necessidade de experimentar novas ferramentas educacionais e identificar as melhores e mais estáveis. Este fato é essencial e promissor para que a relação entre educação e tecnologia seja aprimorada. Essas melhorias, ao que tudo indica, irão permanecer mesmo quando todo esse período turbulento passar.

Tecnologias estão efetivamente fazendo parte da rotina escolar, tanto de professores como de alunos. Esta nova forma de ensino faz com que as escolas reconheçam o valor de algumas dessas tecnologias educacionais, que por sinal nem eram utilizadas.

No início de 2021, algumas escolas já iniciaram o processo de retorno às aulas presenciais, mas este retorno promete trazer algumas novidades para o ambiente educativo, visto que, durante o isolamento social, a tecnologia extremamente utilizada por estudantes e professores com o objetivo de não interromper o processo de aprendizagem foram aprovadas e vieram para ficar.

1.2 Educação Matemática no Brasil

O Programa Internacional de Avaliação de Estudantes (Pisa), tradução de Programme for International Student Assessment, realizado pela Organização para a Cooperação e Desenvolvimento Econômico (OCDE), é um estudo comparativo entre as nações sobre o desempenho de estudantes na faixa etária dos 15 anos, idade em que se julga que o jovem já tenha finalizado a escola básica na maioria dos países, no Brasil o órgão responsável é o Inep².

Baseado no relatório do Pisa - 2018, podemos verificar que a situação da educação matemática é preocupante, visto que o Brasil investe aproximadamente 8,3% do seu orçamento em educação (BRASIL, 2019).

No relatório é informado que a maioria dos estudantes brasileiros se encontram no Nível 1 (27,1%) ou abaixo dele (41,0%), ou seja, 68,1% possuem pelo menos uma das seguintes habilidades relacionadas a disciplina de matemática:

- Responder questões que envolvem contextos familiares, nas quais todos os dados importantes estão explícitos e os pontos-chaves das questões estão claramente definidos;
- Reconhecer dados e executar procedimentos rotineiros apenas com instruções diretas e situações claras;
- Efetuar procedimentos óbvios que podem ser reconhecidos diretamente por instruções dadas.

Algumas observações relevantes são que 68% dos participantes do Pisa - 2018 são alunos da rede estadual e que os alunos das escolas particulares tiveram maior média comparada às redes estaduais e federais.

É preciso que os governantes reconheçam a matemática como uma ciência fundamental para o desenvolvimento de uma sociedade. É fácil concluir que se depende da matemática no cotidiano, que ela é necessária para que as informações sejam atingidas.

Sobre o reconhecimento da matemática, Garbi afirma:

²Inep - Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira

"Diante destes dados históricos, onde poderia ser localizado o início da matemática? Preliminarmente, seria útil tentar definir o que é matemática. Para poupar tempo ao leitor, é bom dizer que esta questão tem inquietado os sábios há muito tempo e jamais se chegou em uma resposta aceita por todos. Algumas pessoas preferem dizer, com certa dose de ironia mas com bastante razão: ' Eu não sei definir o que é Matemática, mas quando a vejo reconheço-a imediatamente"(GARBI, 1997)

No Brasil, existe uma tradição potencializadora que classifica a matemática como uma disciplina de difícil aprendizagem, os estudantes não conseguem perceber a pequena distância que existe entre eles e a matemática. Por este fato, é necessário começar uma reação a fim de que nossos estudantes reencontrem o sentido de se aprender matemática e cabe as nossos governantes, escolas e a comunidade acadêmica produzir diferentes caminhos para alcançar este objetivo.

Conforme Libânio, os novos caminhos a serem traçados no ensino da matemática devem ser baseados nas realidades dos diferentes alunos com o objetivo de verificar a matemática em seus cotidianos.

"Esses princípios levam em conta a natureza da prática educativa escolar numa determinada sociedade. As características do processo de conhecimento, as peculiaridades metodológicas das matérias e suas manifestações concretas na prática docente, as relações entre o ensino e o desenvolvimento dos alunos, as peculiaridades psicológicas de aprendizagem e desenvolvimento conforme idades."(LIBÂNIO, 1994)

1.3 A Importância da Tecnologia na Educação

Pensar em educação, tecnologia da informação e comunicação, traz em conjunto a percepção dos impactos e as transformações atuais, e a clareza que "a atual sala de aula parece ter seus dias contados, bem como a atual grade curricular"(DEMO, 2006).

A tecnologia está dominando todas as áreas do conhecimento e empresas como *Google*, *Apple*, *Microsoft*, *Uber*, *Airbnb* e *Oracle* estão entre as mais valiosas do mundo e poucas delas são relativamente jovens.

De acordo com Barack Obama, presidente dos Estados Unidos de 2009 a 2017, o interesse que crianças e adolescentes têm demonstrado pela inter-

net e pelos games permitirá que a programação seja ensinada nas escolas dos Estados Unidos, país que possui o melhor sistema de inteligência aparelhado do mundo. O ex-presidente dos Estados Unidos destaca a inclusão da programação no ensino das escolas americanas:

Há muitos jovens por aí que acredito que se no colégio tivessem recebido a oportunidade de saber disso – ‘Aqui está como pode desenvolver seu próprio jogo, mas isso exigirá que você saiba matemática e exigirá que saiba ciência. . . ’ – Não apenas deixaria prontos para o trabalho os jovens que não enfrentam quatro anos de faculdade, como também conquistaria as crianças porque elas sentiram que entenderam.(EUA, 2016)

Obama utilizou como exemplo Mark Zuckerberg, alguém que se tornou bem sucedido por ter aprendido a programar sozinho, devido ao seu interesse pelos games. Também afirmou que parte do que está fazendo é trabalhar junto às escolas de todo o país para tentar fazer com que a experiência dos jovens seja relevante. Em um pronunciamento em rede nacional, é evidente que o presidente assume que o sistema de ensino norte-americano precisa melhorar e que os professores precisam inserir programação em suas aulas:

[...] temos que certificar que todas as nossas crianças estão equipadas para os empregos do futuro - o que significa não apenas ter capacidade de trabalhar com computadores, mas desenvolver habilidades analíticas e de codificação para alimentar a nossa economia da inovação [...] Então, eu tenho um plano para ajudar a assegurar, que todas as nossas crianças tenham uma oportunidade de aprender ciência da computação, especialmente meninas e as minorias. É chamado de ciência da computação para todos[...](EUA, 2016)

É possível observar que o discurso de Barack Obama ressalta a necessidade de um ensino que acompanhe as transformações tecnológicas e econômicas. Partindo dessa realidade, compreende-se que, mesmo se tratando de um país de primeiro mundo, o processo de globalização exige que o sistema educacional brasileiro também se atente a tais demandas e proponha o ensino de programação na escola básica, com o intuito de preparar os alunos para o mercado de trabalho. Obama citou algumas estratégias que direcionaram tal proposta, são elas:

- Financiar propostas para os próximos três anos (2017, 2018 e 2019) a fim de que as escolas de ensino fundamental e médio forneçam oportunidades para que todos os alunos aprendam ciência da computação;
- Alavancar recursos existentes na Fundação Nacional de Ciência e no Serviço Nacional e Comunitário, para treinar professores.
- Reunir lideranças políticas, empresariais do ramo da tecnologia para se juntar a causa.

Sendo assim, o papel das tecnologias digitais na sala de aula não é somente para tornar o conteúdo interessante, mas permitir a aprendizagem de novas linguagens, nas quais o aluno seja mobilizado a aprender a ler, compreender e interpretar (MAGDA, 2008). Por isso, cabe ao professor participar de forma ativa do processo de construção do conhecimento do aluno, sendo um mediador e orientador da aprendizagem. Penteado destaca que esse novo cenário afeta:

"A forma como os alunos e professor se comportam na sala de aula e a forma como se comunicam entre si. O professor se vê diante de situações novas (os alunos também) em relação ao que usualmente está acostumado a enfrentar, exigindo estratégias diferentes. Essa nova organização do espaço físico não precisa estar necessariamente vinculada ao uso de computadores, mas um tal uso parece implicar uma mudança na distribuição dos alunos e dos demais componentes presentes na sala de aula."(PENTEADO, 1999)

É preciso que o professor esteja disposto a enfrentar situações desconhecidas e, por isso, ele estará se aventurando pela zona de risco (BORBA; PENTEADO, 2003), na qual não é possível ter o controle e a previsão do que acontecerá. Não significa que, na sala de aula, não há espaço para a imprevisibilidade, mas que isso pode ocorrer com maior frequência quando se utiliza as TICs³, devido a essa alteração da dinâmica da aula.

As TICs permitem que conteúdos matemáticos sejam abordados de outras formas, fazendo com que os alunos não considerem a matemática como

³TICs: são todos os meios técnicos usados para tratar a informação e auxiliar na comunicação, o que inclui o hardware de computadores, rede e telemóveis.

uma disciplina em que é preciso apenas memorizar fórmulas, procedimentos e algoritmos e tenham uma visão mais crítica e ampla.

Por isso, a simples instalação de equipamentos de informática nas escolas; acesso à internet, dentre outros, não indica sinônimo de um ensino de boa qualidade. Pelo contrário, esses recursos podem continuar camuflando práticas convencionais. De acordo com (CARNEIRO; PASSOS, 2014), a utilização das tecnologias apresentam atributos do ensino tradicional em que o professor é o detentor do conhecimento e o aluno apenas recebe e reproduz, não explorando toda sua capacidade de produção de conhecimento. Lima define pedagogia do treinamento como sendo o ato do professor explicar o conteúdo, resolver alguns exemplos e propor que os alunos façam uma grande quantidade de exercícios para poder avaliá-los posteriormente. Essa dinâmica caracteriza a aprendizagem do saber fazer, que, “por não implicar pensamento, acontece simplesmente pela manipulação das regras da operacionalidade do conceito, do treinamento no mecanismo algorítmico” (LIMA, 1998).

Com o objetivo de apresentar uma resposta à situação citada por Obama e unir isso com a necessidade de desenvolver o ensino da matemática no Brasil, a linguagem de programação Python, ou qualquer outra linguagem de programação, pode ser utilizada para desenvolver rotinas de atividades que envolvem conceitos matemáticos. Python se destaca por seu fácil processo de aprendizagem, sintaxe simples e rica biblioteca que auxilia o estudante no decorrer do desenvolvimento de cada código computacional (MARCUNDES, 2018).

No Brasil, existem parâmetros na BNCC⁴ que conduzem as escolas à utilização de tecnologias voltadas para educação, esta indicação é um alerta positivo aos professores que podem planejar suas aulas, utilizando as novas tecnologias.

Nesse sentido, serão apresentados neste trabalho algumas propostas de utilização da linguagem python como ferramenta que potencializa o ensino da matemática, efetivando uma ligação do conteúdo com as tecnologias

⁴BNCC - Base Nacional Comum Curricular

atuais.

Capítulo 2

Programação no Ensino e Aprendizagem

“Eu acho que todos neste país deveriam aprender a programar um computador. Deveriam aprender uma linguagem de programação, pois te ensina como pensar”, Steve Jobs.

2.1 Programação e seus Benefícios

As linguagens de programação têm sido entendidas como uma extensão da escrita. Segundo [Resnick \(2013\)](#), a habilidade de programar permite que o indivíduo “escreva” artefatos digitais (jogos, animações, histórias interativas, etc) que são formas de expressão deste século. Programar é um conjunto de ações que determinamos de forma antecipada para alcançar algum objetivo. Esta definição não é restrita apenas para linguagens de computadores, pode se estender para a realização de tarefas do cotidiano. Aprender a programar abre espaço para outras aprendizagens que estão associadas à resolução de problemas, abstração e comunicação de ideias.

Esta concepção despertou o oferecimento de vários recursos para incitar a aprendizagem de programação por estudantes da Educação Básica em todo o mundo. Tem-se como exemplos o movimento global chamado Hora do Código e os recursos disponibilizados por meio de plataformas web, tais

como, *Khan Academy*¹, *Programaê!*² e *Code.org*³.

O professor Felipe Ladslau, que atua na área de desenvolvimento de softwares há 9 anos e é gerente de projetos na “4 Steps”⁴, afirma que a programação e o desenvolvimento de software vão além da linguagem, e que a programação eleva e desenvolve outras características e habilidades como:

- **Raciocínio Lógico:** Para programar um sistema, é importante que se crie um algoritmo, usando uma linguagem de programação. Esta habilidade está frequentemente presente no aprendizado da matemática, na resolução de problemas matemáticos e em demonstrações de propriedades matemática, validando “Estudar Matemática é resolver problemas.[...]” (DANTE, 1998)
- **Abstração:** Operar na imaginação um objeto que se relaciona com a realidade ou não, tal habilidade é importante na programação pelo fato de que esse processo tem como início a busca por padrões.

Esta ideia converge com Levy (1993) que também alega que o uso de tecnologias potencializa habilidades dos estudantes.

Programar desenvolve a habilidade de resolver problemas, visto que o pensamento computacional é baseado nessa habilidade, de modo que o problema é dividido em uma sequência de partes menores. Em uma entrevista dada ao canal do *YouTube - Happy Code*⁵, o Professor Doutor da UNICAMP, Marcos Augusto Francisco Borges, acredita na importância de ensinar programação para crianças e jovens:

¹Site: <https://www.khanacademy.org/>

²Site: <http://programae.org.br/>

³Site: <https://code.org/>

⁴Site oficial: <https://4steps.com.br>

⁵Site do canal: <https://www.youtube.com/c/HappycodeBrasil>

“É muito importante ensinarmos nossas crianças e jovens a programar, não porque a gente quer que todos eles virem profissionais de informática, mas porque já está provado que uma criança que aprende a programar, aprende a pensar, aprende a resolver problemas mais complexos, aprende a dividir um problema grande em problemas menores, aprende a persistir para chegar em alguma coisa, é uma habilidade que prepara uma criança para no futuro ser um profissional excelente em qualquer área que ele vá atuar.”(BORGES, 2018)

O discurso do Professor Marcos converge com a estratégia traçada por [Polya \(1978\)](#), em específico com a segunda etapa de seu estudo sobre resolução de problemas e as quatro etapas para a resolução de um problema.

1. Compreender o problema;
2. Elaborar um plano;
3. Executar o plano;
4. Fazer o retrospecto ou verificação.

Nesta perspectiva, programação deixou de ser “coisa de nerd”, está se tornando um pré requisito para o aprendizado de todas as áreas de conhecimento, é uma habilidade a ser comparada com como saber ler e escrever. Em um futuro próximo, quem não aprender a programar, sempre estará condicionado a um outro profissional ou até mesmo pode ficar fora do mercado de trabalho. Não saber programar significa ser analfabeto no século 21.

Existem estudantes que possuem acesso a estes recursos de forma totalmente autônoma, via “navegação” na internet, grupos ou comunidades virtuais de discussão, dando início a suas experiências com programação. Há também um grupo de estudantes que suas escolas viabilizam, por meio das aulas de informática, a aprendizagem inicial de programação.

O *Massachusetts Insitute of Technology*, referência em tecnologia no mundo, possui um projeto chamado *OpenCourseWare*⁶. Essa plataforma, baseada em publicar praticamente todo o conteúdo dos cursos do MIT⁷, está aberta e disponível para o mundo e é uma atividade permanente do MIT.

⁶Site Oficial: <https://ocw.mit.edu/courses/>

⁷Massachusetts Insitute of Technology

A plataforma possui mais de 500 milhões de visitas e mais de 2400 materiais dos cursos do instituto. Um dos seus principais objetivos é que alunos de qualquer faixa de conhecimento desenvolvam suas habilidades e usem isto para enfrentar alguns dos desafios mundiais: desenvolvimento sustentável, mudança climática e erradicação do câncer.

Dentro desta plataforma, encontram-se vários cursos sobre programação, inclusive sobre Python, que é a ferramenta que está sendo discutida nesta pesquisa. Alguns exemplos de cursos sobre Python presentes na plataforma são:

- Introduction to Computer Science and Programming in Python;
- A Gentle Introduction to Programming Using Python;
- Programming for the Puzzled;
- Introduction to Algorithms.

É necessário ressaltar que a matemática e a programação estão intimamente ligadas, a programação elevou a matemática,. Podemos evidenciar este fato, uma vez que os problemas reais do século XXI só puderam ser solucionados devido a existência do computador, esta é uma das justificativas que a programação deve entrar no sistema educacional como uma parte fundamental da disciplina de matemática, afirma [Wolfram \(2017\)](#).

Segundo [Papert \(1994\)](#), o computador tem o potencial de transformar o ensino da matemática numa aprendizagem mais “lógica” e menos mecânica.

O currículo de matemática das escolas do Brasil precisa ser modernizado, repensado e é necessário fazer com que a programação e o raciocínio computacional esteja mais presente no cotidiano escolar. Verifica-se a importância de que a nossa sociedade entenda a como programar para solucionar problemas que irão surgir futuramente.

2.2 A Linguagem Python

Em 2019, ocorreu o sexto ranking interativo anual do IEEE⁸ Spectrum das principais linguagens de programação. Por meio de uma criteriosa análise, chegaram ao seguinte resultado:

Rank	Language	Type	Score
1	Python	🌐 📱 ⚙️	100.0
2	Java	🌐 📱	96.3
3	C	📱 📺 ⚙️	94.4
4	C++	📱 📺 ⚙️	87.5
5	R	📺	81.5
6	JavaScript	🌐	79.4
7	C#	🌐 📱 📺 ⚙️	74.5
8	Matlab	📺	70.6
9	Swift	📱 📺	69.1
10	Go	🌐 📺	68.0

Figura 2.1: Top 10 das linguagens de programação mais utilizadas - IEEE Spectrum 2019.

Verifica-se que a popularidade da linguagem Python é impulsionada pela quantidade expressiva de bibliotecas especializadas disponível para a linguagem, principalmente em inteligência artificial. A linguagem possui mais de 125.000 bibliotecas em diversos campos da tecnologia: desenvolvimento web, configuração de nuvem, inteligência artificial, *machine learning* e análise de dados.

A linguagem foi lançada pela primeira vez em 1991, e toda esse potencial no campo da programação e sua popularidade não foram previstos de acontecer da forma como está acontecendo.

⁸IEEE - Institute of Electrical and Electronics Engineers é a maior organização profissional técnica do mundo dedicada ao avanço da tecnologia para o benefício da humanidade.

Nos últimos anos, o interesse pela linguagem Python vem crescendo. Um índice que mede a frequência de pesquisa no *Google*, *YouTube* e outras ferramentas de busca indica que a Python é a 3ª linguagem mais procurada. Vale ressaltar que linguagem dominantes como C e *Java* perderam certa popularidade e Python rompeu seus limites no período de 2018 a 2020.

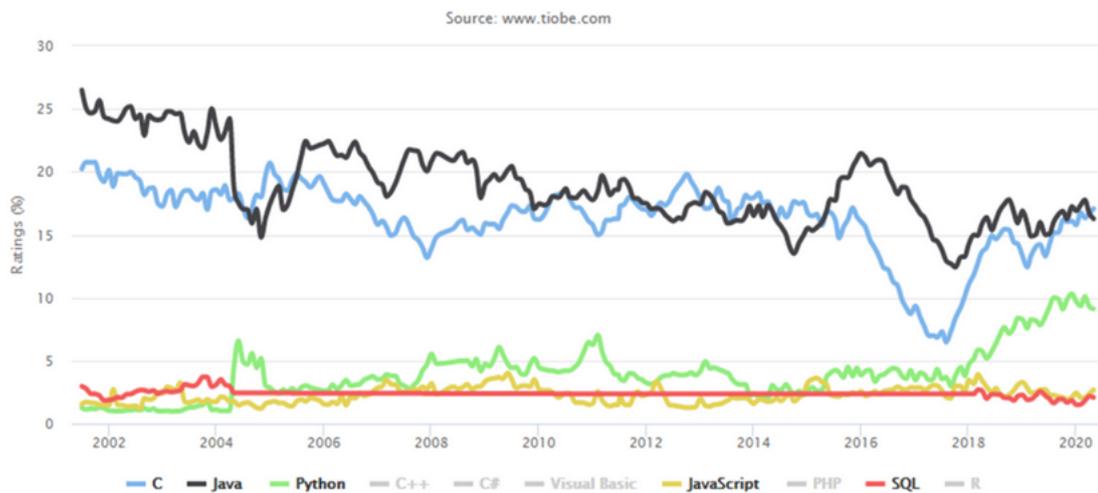


Figura 2.2: Índice Tiobe - Frequência de Pesquisas

Com relação à utilização da linguagem, tem-se o ranking do Github de linguagens mais utilizadas, mostrando que Python se tornou a segunda linguagem mais utilizada no mundo em 2019.



Figura 2.3: Ranking do Github das linguagens mais utilizadas no mundo.

Programadores profissionais e estudantes que utilizam a linguagem Python utilizam a expressão “*programar em Python é como escrever uma*

carta em inglês para o computador”, pelo fato da linguagem utilizar comandos intuitivos.

A linguagem Python foi desenvolvida sob dezenove princípios⁹:

1. Bonito é melhor que feio.
2. Explícito é melhor do que implícito.
3. Simples é melhor do que complexo.
4. Complexo é melhor do que complicado.
5. Horizontal é melhor do que aninhado.
6. Esparso é melhor que denso.
7. Legibilidade conta.
8. Casos especiais não são especiais o suficiente para quebrar regras.
9. Porém, a praticidade supera a pureza.
10. Os erros nunca devem passar silenciosamente.
11. A menos que sejam explicitamente silenciados.
12. Diante da ambiguidade, recuse a tentação de adivinhar.
13. Deve haver uma, e de preferência apenas uma, forma óbvia de se fazer algo.
14. Embora essa forma possa não ser óbvia no início, a menos que você seja holandês.
15. Agora é melhor que nunca.
16. Mas “nunca” é melhor do que “imediatamente agora”.
17. Se a implementação é difícil de explicar, é uma má ideia.
18. Se a implementação for fácil de explicar, pode ser uma boa ideia.

⁹Crie um arquivo .py e digite “import this”, estes princípios será exibidos em inglês.

19. Namespaces são uma ótima ideia - vamos fazer mais disso!

Diferenciando duas das linguagens mais utilizadas no mundo , observe-se:

- **Java** é uma linguagem de programação de propósito geral e orientada a objetos, projetada para desenvolver várias aplicações.
- **Python** é uma linguagem de programação simples, de alto nível, projetada principalmente para inteligência artificial, aprendizado de máquina (uma técnica de análise de dados que ensina os computadores a fazer o que naturalmente os seres humanos e aos animais já fazem: aprender com a experiência.), desenvolvimento web.

Uma das justificativas para a ascensão da linguagem *Python* é o fato de apresentar sintaxe simples, legibilidade de código e comandos semelhantes ao inglês, fazendo com que a codificação se torne muito mais fácil e eficiente do que com outras linguagens de programação.

Comparando a linguagem *Python* com a linguagem *Java*, a primeira é uma linguagem dinâmica e mais flexível, comparada à segunda. Pode-se perceber que, enquanto na linguagem *Java* precisa-se declarar o tipo de uma variável, em *Python* não é necessária essa ação, sendo assim, *Python* por ser mais simples e eficiente possui menos linhas de comando do que *Java*.

Em *Java*, há mais regras gramaticais, sem levar em consideração o uso contante do ponto-e-vírgula, já em *Python*, dependendo da proposta, não é preciso utilizar o ponto-e-vírgula para finalizar uma linha de comando. No exemplo abaixo, foi desenvolvida a mesma atividade (exibir na tela a mensagem (“*Hello World!!!*”), utilizando *Java* e *Python*.

Uma proposta usual quando se quer aprender certa linguagem de programação é o programa “*Hello World*”, o qual pode ser desenvolvido em praticamente todas as linguagens e tem por objetivo apresentar de forma breve os conceitos de certa linguagem.

Para exibir “*Hello World!!!*” na tela utilizando a linguagem *Python*, é necessário apenas uma linha de comando, este é mais um ponto que faz

com que a linguagem seja bastante utilizada por iniciantes, devido a sua simplicidade, veja o exemplo.

```
1 print("Hello World!!!")
```

Na tela da linha de comando escrita em *Java* para exibir “*Hello World!!!*”, foram utilizadas 5 linhas de comando, como é possível constatar abaixo.

```
1 public class Main
2 {
3     public static void main(String[] args) throws Exception
4     {
5         System.out.println("Hello World!!!");
6     }
7 }
```

O desenvolvimento das linhas de comandos acima justificam a característica “simplicidade” da linguagem *Python*, que facilita a aprendizagem da programação.

Em questão de portabilidade, as linguagens estão no mesmo nível, possuindo compatibilidade com *Windows*, *Mac* e *Linux*. A agilidade prática com que *Python* trabalha com dados científicos e operações em desenvolvimento é superior, é utilizado em cenários mais experimentais e possui atualmente um movimento tecnológico crescente.

Observando os dados apresentados, acredita-se que *Python* irá ultrapassar a popularidade de *Java* que, atualmente, apresenta-se como a linguagem dominante.

A linguagem *Python* encontra-se no centro do mundo quando o assunto é inteligência artificial, cada vez mais os empregos estão sendo criados pelo desenvolvimento desta área que depende muito de *Python*.

Um levantamento realizado pela GeekHunter, startup de consultoria, referência em recrutamento de profissionais de TI, mostrou que o total de vagas abertas no setor de tecnologia da informação em 2020 teve um aumento de 310%. E, de acordo com dados do Banco Mundial, até 2024 haverá a criação de, aproximadamente, 420 mil novas vagas de emprego na área. São números bastante expressivos, ainda mais se levarmos em consideração a crise econômica que atravessamos e a alta taxa de desemprego. (NSC, 2021)

Atualmente, existe uma escassez de programadores, estão sendo criadas vagas de empregos referente a pessoas que tenham conhecimento em tecnologia e linguagens e de programação. Este fato implicará a busca deste conhecimento e, principalmente, nas linguagens de programação mais eficientes e populares.

Uma das preocupações com o trabalho em inteligência artificial envolve a crescente demanda e o deficit atual desses profissionais. Na Europa, o cálculo era que faltavam 600 mil trabalhadores em carreiras relacionadas a tecnologias da informação e comunicação. Há, também, uma disputa pelos talentos. Levantamento da União Europeia identificou 240 profissionais de países do bloco no Vale do Silício, principal sede de empresas de tecnologia dos Estados Unidos. (VALENTE, 2020)

De forma categórica, a simplicidade, gratuidade e popularidade foram as principais características que influenciaram a escolha da linguagem *Python* para o desenvolvimento desta pesquisa.

Capítulo 3

Aplicações

“Segundo a filosofia Logo, o aprendizado acontece através do processo de a criança inteligente “ensinar” o computador burro, ao invés do computador ensinar a criança burra. Com essa proposta, Papert inverte o atual quadro do uso do computador na escola. O computador deixa de ser o meio de transferir a informação e passa a ser a ferramenta com a qual a criança pode formalizar os seus conhecimentos intuitivos.” (PAPERT, 1988, pág. 9)

Para o desenvolvimento desta seção é necessário que os envolvidos no processo tenham um conhecimento prévio da linguagem de programação Python, visto que o foco principal é fazer com que o aluno desenvolva um programa utilizando conceitos matemáticos.

Esta linha de desenvolvimento será feita a partir de atividades que irão juntamente com a linguagem Python abordar cada um dos conteúdos matemáticos selecionados e suas respectivas implementações computacionais. Tais atividades apresentarão tabelas com questionamentos fundamentais sobre os assuntos que serão abordados, com o objetivo de revisar o conteúdo a ser trabalhado e guiar a produção do código.

3.1 Matrizes

Nesta seção, serão expostas algumas aplicações em Python que utilizam as definições sobre matrizes. Serão implementados códigos em linguagem Python para fazer a leitura de uma matriz inserida pelo usuário, desenvol-

vendo assim o conceito de posição de elementos de uma matriz e a respectiva indexação do número de linhas e colunas.

Esses programas utilizarão operadores condicionais em Python, juntamente com os conceitos de: diagonal principal, diagonal secundária, matriz identidade, matriz triangular, matriz simétrica e matriz inversível. Serão desenvolvidas também implementações que contemplem as operações entre matrizes e entre números e matrizes.

Antes de iniciar as aplicações, é importante registrar que existe uma biblioteca chamada *NumPy*¹, que possui recursos específicos já implementados que poderiam resolver grande destas atividades. Porém, o objetivo das aplicações é fazer com que o aluno trabalhe com os conceitos e comandos básicos de Python para resolver as situações propostas.

A aplicação a seguir tem por objetivo trabalhar com os alunos conceitos básicos de matrizes, assim como a posição dos elementos que a compõem. Neste momento, diversas situações serão descritas e comentadas durante o desenvolvimento.

Aplicação 3.1

Desenvolver um programa que crie uma matriz 3×3 que permita ao usuário digitar os valores dos elementos dessa matriz, exibindo no final a matriz com a formatação correta.

Antes de tudo, deve-se assegurar que o conceito de matrizes e a localização de seus elementos é familiar ao aluno que está desenvolvendo a aplicação, visto que este fato é fundamental para toda e qualquer teoria sobre matrizes. Só assim, será atingido o objetivo de fazer com que o aluno amplie o seu conhecimento sobre o assunto.

Questionário	
Perguntas:	Respostas esperadas:
Qual é a definição de matriz?	Uma matriz é uma tabela formada por elementos distribuídos em m linhas e n colunas, com $m, n \in \mathbb{N}$.

¹Site oficial da biblioteca NumPy: <https://numpy.org/>

Como é interpretada a localização dos elementos de uma matriz em função dos índices?	Cada elemento de uma matriz está disposto em linhas e colunas e seu índice referencia, respectivamente, a linha e coluna a qual ele pertence.
Qual é a definição de matriz quadrada?	É uma matriz que possui o número de linhas igual ao número de colunas.

A principal característica ao construir uma matriz é definir a quantidade de linhas e colunas. Neste caso, conforme descrito anteriormente, propõe-se trabalhar com uma matriz 3×3 , ou seja, uma matriz quadrada de ordem 3. Assim, deve-se definir na linha de comando uma lista suporte, que permite a inserção dos valores da matriz pelo usuário, seguindo a ordenação usual de linhas e colunas.

```
m = [[0,0,0],[0,0,0],[0,0,0]]
```

Para preencher a matriz, serão utilizados dois comandos de repetição (*loop*) do tipo *for*, sendo o primeiro responsável por percorrer as linhas da matriz e o segundo, as colunas. Isso será feito por meio da implementação apresentada a seguir.

```
for i in range(0, 3):  
    for j in range(0, 3):  
        m[i][j] = float(input(f'Digite o valor da posição  
        ↪ a{i + 1}{j + 1}: '))
```

Feito isso, os elementos da matriz já estão inseridos e agora é o momento de exibir a matriz, utilizando o código abaixo.

```
for i in range(0, 3):  
    for j in range(0, 3):
```

```
print(f'[{m[i][j]:^5}]', end='')
print()
```

A linha de comando completa que sintetiza o programa está apresentada a seguir.

```
1 m = [[0,0,0],[0,0,0],[0,0,0]]
2
3 for i in range(0, 3):
4     for j in range(0, 3):
5         m[i][j] = float(input(f'Digite o valor da posição
6             ↪ a{i + 1}{j + 1}: '))
7
8 for i in range(0, 3):
9     for j in range(0, 3):
10        print(f'[{m[i][j]:^5}]', end='')
11    print()
```

A próxima atividade tem por principal objetivo trabalhar a ideia de diagonal principal de uma matriz e como calcular o traço de uma matriz. Utilizando os conceitos trabalhados na atividade anterior, é de grande valor levantar questionamentos sobre as definições de diagonal principal e traço de uma matriz, estas podem ser feitas a partir da tabela a seguir.

Processo	
Perguntas:	Respostas esperadas:
O que é a diagonal principal de uma matriz quadrada?	São todos elementos que possuem seus índices iguais, ou seja, o número que representa a linha é igual ao número que representa a coluna.
Qualquer matriz possui diagonal principal?	Não, apenas as matrizes quadradas.
O que é o traço de uma matriz?	A soma dos elementos da diagonal principal é denominada traço da matriz.

Segue o enunciado da próxima atividade relacionada a matrizes:

Aplicação 3.2

Criar um programa em que o usuário forneça uma matriz de ordem 4 e faça a soma dos valores de sua diagonal principal.

Com o intuito de desenvolver esta aplicação, é possível verificar que a parte inicial propõe os mesmos questionamentos feitos na aplicação anterior. Será criada uma variável suporte para auxiliar o aluno na soma de valores.

```
m = [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
traco = 0
```

Utilizando os conceitos de diagonal principal e traço de uma matriz, o aluno deverá desenvolver uma linha de comando, a qual irá receber os valores da matriz 4×4 e, conseqüentemente, irá identificar os valores que pertencem à diagonal principal e somá-los.

```
for i in range(0, 4): # Quantidade de linhas da matriz
    for j in range(0, 4): # Quantidade de colunas da matriz
        m[i][j] = float(input(f'Digite o valor da posição
        ↪ a{i + 1}{j + 1}: ')) # Entrada de elementos da
        ↪ matriz
    if i == j: # Verifica se os índices das posições
        ↪ dos elementos da matriz são iguais.
        traco = traco + m[i][j] # Faz a soma dos
        ↪ elementos da diagonal principal.
```

Concluindo a linha de comando, a parte final desta aplicação é mostrar a matriz digitada pelo usuário e apresentar o resultado para o traço da matriz.

```
for i in range(0, 4):
    for j in range(0, 4):
        print(f'[{m[i][j]:^5}]', end='')
    print()
```

```
print(f'A soma dos elementos da diagonal principal é igual a  
↳ {traco}.')
```

Vale ressaltar que, dentro das aplicações sobre matriz, o aluno ao desenvolver o código, está a todo momento utilizando seu domínio sobre a localização de elementos dentro de uma matriz.

Finalizando a aplicação, obtém-se a seguinte linha de comando que reúne todos os raciocínios e questionamentos que surgiram durante o desenvolvimento do código.

```
1 m = [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]  
↳ # Matriz suporte  
2 traco = 0  
3  
4 for i in range(0, 4): # Quantidade de linhas da matriz  
5     for j in range(0, 4): # Quantidade de colunas da matriz  
6         m[i][j] = float(input(f'Digite o valor da posição  
↳ a{i + 1}{j + 1}: ')) # Entrada de elementos da  
↳ matriz  
7         if i == j: # Verifica se os índices das posições  
↳ dos elementos da matriz são iguais.  
8             traco = traco + m[i][j] # Faz a soma dos  
↳ elementos da diagonal principal.  
9  
10 for i in range(0, 4):  
11     for j in range(0, 4):  
12         print(f'[{m[i][j]:^5}]', end='')  
13     print()  
14  
15 print(f'A soma dos elementos da diagonal principal é igual a  
↳ {traco}.')
```

Na aplicação a seguir, os conceitos trabalhados anteriormente permane-

cerão e será preciso incluir a definição de matriz identidade, sendo assim, será acrescentado nas tabelas anteriores o seguinte questionamento:

Processo	
Perguntas:	Respostas esperadas:
Qual a definição de matriz identidade?	Toda matriz quadrada em que os elementos da diagonal principal são iguais a 1 e o restante dos elementos são iguais a zero.
Qual a relação do traço de uma matriz identidade e sua ordem?	O valor que representa o traço de uma matriz identidade é igual a ordem desta mesma matriz.

Aplicação 3.3

Criar um programa em que o usuário forneça uma matriz de ordem 3 e verifique se a matriz digitada é uma matriz identidade.

Discutidos os conceitos relacionados à definição de matriz identidade, o objetivo do aluno é traçar uma estratégia para que seu programa identifique através dos índices se os elementos:

- fazem parte da diagonal principal;
- são todos iguais a um;
- que não estão localizados na diagonal principal são iguais a zero.

Assim, o aluno irá, primeiramente criar um código que faça a verificação de uma característica particular da matriz quadrada de ordem 3.

```
m = [[0, 0, 0], [0, 0, 0], [0, 0, 0]] # Matriz suporte
verificador_1 = 0
verificador_2 = 0
```

Como o traço de uma matriz identidade de ordem 3 é igual a 3, o aluno pode desenvolver um código para realizar esta verificação, lembrando que este fato não garante que a matriz seja uma matriz identidade.

Neste momento, será utilizado o primeiro verificador, ele funcionará como um “gatilho”, que irá percorrer os elementos da matriz que possuem o número da linha igual ao número da coluna e, ao mesmo tempo, irá somá-los. Esta estratégia é representada pelo código a seguir:

```
for i in range(0, 3): # Quantidade de linhas da matriz
    for j in range(0, 3): # Quantidade de colunas da matriz
        m[i][j] = float(input(f'Digite o valor da posição
        ↪ a{i + 1}{j + 1}: ')) # Entrada de elementos da
        ↪ matriz
    if i == j and m[i][j] == 1: # Verifica se os
        ↪ índices das posições dos elementos da matriz são
        ↪ iguais.
        verificador_1 = verificador_1 + m[i][j] # Faz a
        ↪ soma dos elementos da diagonal principal.
```

Para finalizar a execução desta aplicação, o aluno deve criar uma estratégia para verificar se todos os elementos que não pertencem a diagonal principal são iguais de zero. Para garantir tal fato, basta observar que uma matriz de ordem 3 possui 9 elementos, como os 3 elementos da diagonal principal não precisam ser verificados, restam 6 elementos que precisam ser iguais de 0. É neste momento que deve-se utilizar o segundo “gatilho”, para cada elemento fora da diagonal principal que for igual a 0, o “gatilho” será incrementado em 1. O raciocínio acima pode ser traduzido para um código da seguinte forma:

```
for i in range(0, 3):
    for j in range(0, 3):
        if i != j and m[i][j] == 0: # Verifica se todos os
            ↪ elementos da diagonal principal são iguais a 1;
            # Verifica se todos os elementos que não são da
            ↪ diagonal principal são diferentes de 0.
            verificador_2 = verificador_2 + 1
```

Na etapa final, existirá uma verificação de ativação dos verificadores, ou seja, o primeiro verificador tem por objetivo constatar se todos os elementos da diagonal principal são iguais a 1. O segundo tem por objetivo examinar se todos os elementos que não estão na diagonal principal são iguais a zero.

No momento em que as variáveis que representam os verificadores assumem os valores esperados, isso irá garantir que a matriz é uma matriz identidade. Para representar a situação, há a seguinte linha de comando:

```
if verificador_1 == 3 and verificador_2 == 6: # Aciona a
    ↪ variável para verificar se a matriz é identidade ou não.
    print('A matriz digitada é uma matriz identidade de
        ↪ ordem 3.')
else:
    print('A matriz digitada não é uma matriz identidade de
        ↪ ordem 3.')
```

Reunindo os raciocínios e códigos gerados a partir do desenvolvimento e produção do programa que tem por objetivo verificar se uma matriz digitada pelo usuário é uma matriz identidade ou não, obtém-se o seguinte programa:

```
1 m = [[0, 0, 0], [0, 0, 0], [0, 0, 0]] # Matriz suporte
2 verificador_1 = 0
3 verificador_2 = 0
4
5 for i in range(0, 3): # Quantidade de linhas da matriz
6     for j in range(0, 3): # Quantidade de colunas da matriz
7         m[i][j] = float(input(f'Digite o valor da posição
            ↪ a{i + 1}{j + 1}: ')) # Entrada de elementos da
            ↪ matriz
8         if m[i][j] == 1 : # Verifica se os índices das
            ↪ posições dos elementos da matriz são iguais.
```

```

9         verificador_1 = verificador_1 + m[i][j] # Faz a
           ↳ soma dos elementos da diagonal principal.
10
11     for i in range(0, 3):
12         for j in range(0, 3):
13             if i != j and m[i][j] == 0: # Verifica se todos os
           ↳ elementos da diagonal principal são iguais a 1;
           # Verifica se todos os elementos que não são da
           ↳ diagonal principal são diferentes de 0.
14                 verificador_2 = verificador_2 + 1
15
16
17     if verificador_1 == 3 and verificador_2 == 6: # Aciona a
           ↳ variável para verificar se a matriz é identidade ou não.
18         print('A matriz digitada é uma matriz identidade de
           ↳ ordem 3.')
19     else:
20         print('A matriz digitada não é uma matriz identidade de
           ↳ ordem 3.')

```

Com o objetivo de resolver as Aplicações 3.4 e 3.5, serão utilizados os conceitos anteriores sobre matrizes e adicionando alguns questionamentos exibidos na tabela a seguir:

Processo	
Perguntas:	Respostas esperadas:
O que é uma matriz triangular?	Matriz triangular é aquela cujos elementos situados "de um mesmo lado" da diagonal principal são iguais a 0, ou seja, considerando a matriz $M = (a_{ij})$, ela será classificada como triangular se $a_{ij} = 0$ para $i < j$ ou $a_{ij} = 0$ para $i > j$.

O que é uma matriz triangular superior?	Matriz triangular superior é aquela cujos elementos situados abaixo da diagonal principal são iguais a 0, ou seja, considerando a matriz $M = (a_{ij})$, ela será classificada como triangular superior se $a_{ij} = 0$ para $i > j$.
O que é uma matriz triangular inferior?	Matriz triangular inferior é aquela cujos elementos situados acima da diagonal principal são iguais a 0, ou seja, considerando a matriz $M = (a_{ij})$, ela será classificada como triangular superior se $a_{ij} = 0$ para $i < j$.

Aplicação 3.4

Criar um programa em que o usuário forneça uma matriz de ordem 3 e verifique se a matriz digitada é uma matriz triangular superior.

Para desenvolver um código que apresente uma solução para esta aplicação, utilizam-se partes da lógica da resolução das aplicações anteriores, define-se uma matriz quadrada de ordem 3 e os verificadores. As diferenças que serão utilizadas nesta solução são:

- Verificar se elementos que estão à esquerda da diagonal principal são iguais a zero, ou seja, para elementos com o índice ij composto por $i > j$ teremos, a_{21} , a_{31} e a_{32} iguais a zero.

Para isso, deve-se desenvolver a seguinte linha de comandos:

```
for i in range(0, 3):  
    for j in range(0, 3):
```

```

if i > j and m[i][j] == 0: # Verifica se todos os
    ↪ elementos à esquerda da diagonal principal são
    ↪ iguais a zero.
    verificador_1 = verificador_1 + 1

```

Os verificadores devem ser incrementados três vezes caso a matriz digitada seja uma matriz triangular superior. Para que esta condição seja constatada há o seguinte código:

```

if verificador_1 == 3: # Caso o verificador seja acionado
    ↪ nesta quantidade (3), a matriz será classificada como
    ↪ diagonal inferior.
    print('A matriz é uma matriz triangular superior.')
else:
    print('A matriz não é uma matriz triangular superior')

```

Munido de todos os códigos que compõem cada parte da discussão da resolução desta aplicação, verifica-se o seguinte programa:

```

1 m = [[0, 0, 0], [0, 0, 0], [0, 0, 0]] # Matriz suporte
2 verificador_1 = 0
3
4 for i in range(0, 3): # Quantidade de linhas da matriz
5     for j in range(0, 3): # Quantidade de colunas da matriz
6         m[i][j] = float(input(f'Digite o valor da posição
7             ↪ a{i + 1}{j + 1}: ')) # Entrada de elementos da
8             ↪ matriz
9     print('-=' * 30)
10 if m == [[0, 0, 0], [0, 0, 0], [0, 0, 0]]: # Verifica se a
11     ↪ matriz é a matriz nula, caso seja, ela também é diagonal
12     ↪ inferior.
13     print('A matriz é uma matriz triangular superior.')
14 else:
15     for i in range(0, 3):

```

```
12     for j in range(0, 3):
13         if i < j and m[i][j] == 0: # Verifica se todos
            ↪ elementos à esquerda da diagonal principal
            ↪ são iguais de zero.
14             verificador_1 = verificador_1 + 1
15
16     if verificador_1 == 3: # Caso o verificador seja
            ↪ acionado nesta quantidade (3), a matriz será
            ↪ classificada como diagonal inferior.
17         print('A matriz é uma matriz triangular superior.')
18     else:
19         print('A matriz não é uma matriz triangular
            ↪ superior')
```

Fazendo os devidos ajustes, é trivial produzir uma solução para a Aplicação 3.5, visto que agora os elementos que estão à esquerda da diagonal principal são diferentes de zero e os que estão à direita são iguais a zero.

Aplicação 3.5

Criar um programa em que o usuário forneça uma matriz de ordem 3 e verifique se a matriz digitada é uma matriz triangular inferior.

Realizando as modificações necessárias com relação aos índices, pode-se apresentar a seguinte solução para a Aplicação 3.5, onde se obtém a seguinte linha de comando:

```
1 m = [[0, 0, 0], [0, 0, 0], [0, 0, 0]] # Matriz suporte
2 verificador_1 = 0
3
4 for i in range(0, 3): # Quantidade de linhas da matriz
5     for j in range(0, 3): # Quantidade de colunas da matriz
6         m[i][j] = float(input(f'Digite o valor da posição
            ↪ a{i + 1}{j + 1}: ')) # Entrada de elementos da
            ↪ matriz
```

```

7
8 if m == [[0, 0, 0], [0, 0, 0], [0, 0, 0]]: # Verifica se a
    ↪ matriz é a matriz nula, caso seja, ela também é
    ↪ triangular inferior.
9     print('A matriz é uma matriz triangular inferior.')
10 else:
11     for i in range(0, 3):
12         for j in range(0, 3):
13             if i > j and m[i][j] == 0: # Verifica se todos
                ↪ elementos à direita da diagonal principal
                ↪ são iguais de zero.
14                 verificador_1 = verificador_1 + 1
15
16     if verificador_1 == 3: # Caso o verificador seja
        ↪ acionado, a matriz é triangular inferior.
17         print('A matriz é uma matriz triangular superior.')
18     else:
19         print('A matriz não é uma matriz triangular
        ↪ inferior.')

```

Para iniciar a discussão e produção de uma solução para a Aplicação 3.6, algumas questões devem ser levantadas, são elas:

Processo	
Perguntas:	Respostas esperadas:
Qual a definição de matriz transposta?	Uma matriz $A_{n \times n} = (a_{ij})$ possui uma matriz $B_{m \times n} = (b_{ji})$ chamada de transposta onde as colunas de B são ordenadamente iguais as linhas de A , ou seja, $a_{ij} = b_{ji}$.
Qual a relação entre os índices dos elementos de uma matriz e sua transposta?	Um elemento de uma matriz localizado em ij estará localizado na matriz transposta em ji .

Como classificar uma matriz como simétrica?	Matriz simétrica é uma matriz quadrada de ordem n , tal que ela seja idêntica a sua transposta.
---	---

Aplicação 3.6

Criar um programa em que o usuário forneça uma matriz de ordem 3 e verifique se a matriz digitada é uma matriz simétrica.

A parte inicial da solução desta aplicação é idêntica às partes iniciais das soluções das aplicações anteriores que envolvem matrizes. Existe uma diferença, pois duas matrizes serão comparadas, assim, é preciso que seja criada uma matriz suporte a mais, como se observa a seguir:

```
m = [[0, 0, 0],[0, 0, 0],[0, 0, 0]]
n = [[0, 0, 0],[0, 0, 0],[0, 0, 0]]
```

Uma dessas matrizes será utilizada para receber os valores que o usuário digitar, para isso verifica-se o seguinte código:

```
for i in range(0, 3): # Quantidade de linhas da matriz
    for j in range(0, 3): # Quantidade de colunas da matriz
        m[i][j] = float(input(f'Digite o valor da posição
        ↪ a{i + 1}{j + 1}: '))
```

Utilizando os questionamentos sobre matriz transposta e a relação entre os índices que localizam cada elemento de uma matriz, sabe-se que, em uma matriz, o elemento que está localizado na posição $i = 2$ e $j = 3$ (linha 2 e coluna 3), na matriz transposta estará localizado na posição $i = 3$ e $j = 2$ (linha 3 e coluna 2).

Para efetuar isso em um programa de computador, utilizando a linguagem *Python*, será utilizada a segunda matriz suporte para atribuir os elementos da matriz digitada pelo usuário em suas respectivas posições, assim criando a matriz transposta relativa à matriz digitada. Assim, observa-se a seguinte linha de comando:

```
for i in range(0, 3): # Quantidade de linhas da matriz
    for j in range(0, 3): # Quantidade de colunas da matriz
        n[i][j] = m[j][i]
```

Para finalizar o processo deve-se utilizar a definição de matriz simétrica para elaborar um código que irá comparar elemento a elemento as duas matrizes, a matriz digitada pelo usuário e a sua transposta. E assim concluir se a matriz é simétrica ou não. Para isso tem-se o código:

```
if a == b:
    print("A matriz é simétrica.")
else:
    print("A matriz não é simétrica.")
```

Organizando a estratégia de resolução da Aplicação 3.6, chega-se ao seguinte produto:

```
1 a = [[0, 0, 0],[0, 0, 0],[0, 0, 0]]
2 b = [[0, 0, 0],[0, 0, 0], [0, 0, 0]]
3
4 for i in range(0, 3): # Quantidade de linhas da matriz
5     for j in range(0, 3): # Quantidade de colunas da matriz
6         a[i][j] = float(input(f'Digite o valor da posição
7             ↪ a{i + 1}{j + 1}: '))
8
9 for i in range(0, 3): # Quantidade de linhas da matriz
10    for j in range(0, 3): # Quantidade de colunas da matriz
11        b[i][j] = a[j][i]
12
13 if a == b:
14     print("A matriz é simétrica.")
```

```

14 else:
15     print("A matriz não é simétrica.")

```

Em continuidade às atividades desenvolvidas sobre matrizes, na Aplicação 3.7, será trabalhado o conceito de determinantes de matrizes e a relação dele com o cálculo da matriz inversa.

Assim, os questionamentos a serem levantados antes do início da produção do programa são:

Processo	
Perguntas:	Respostas esperadas:
O que é o determinante de uma matriz?	Determinante é um número que está associado a uma matriz quadrada.
Quais métodos são utilizados para o cálculo do determinante de uma matriz?	Para calcular o determinante de uma matriz pode-se utilizar regra de Sarrus, teorema fundamental (de Laplace), propriedades dos determinantes.
Qual a relação do valor do determinante com a existência da matriz inversa?	Se o determinante de uma matriz for diferente de zero, ela irá admitir inversa, caso contrário ela não irá admitir inversa.

Os questionamentos abordados anteriormente são essenciais para a construção de um protocolo de resolução da aplicação a seguir, ter o domínio de um processo que calcule o determinante irá facilitar nas decisões a serem tomadas na organização do programa.

Discutidas as definições que serão utilizadas no desenvolvimento de uma linha de raciocínio e conseqüentemente na produção do programa a ser desenvolvido, passa-se ao enunciado da próxima aplicação:

Aplicação 3.7

Criar um programa em que o usuário forneça uma matriz de ordem 3 e verifique se a matriz digitada é uma matriz inversível.

No desenvolvimento desta aplicação, de forma análoga aos programas das aplicações sobre matrizes apresentadas anteriormente, ela irá desenvolver o mesmo papel. Assim, tem-se:

```
m = [[0, 0, 0],[0, 0, 0],[0, 0, 0]]
```

Para que o usuário entre com os valores da matriz que se deseja verificar se admite inversa, verifica-se o seguinte código:

```
for i in range(0, 3): # Quantidade de linhas da matriz
    for j in range(0, 3): # Quantidade de colunas da matriz
        m[i][j] = float(input(f'Digite o valor da posição
        ↪ a{i + 1}{j + 1}: '))
```

Para calcular o determinante da matriz inserida pelo usuário, será utilizada a regra de Sarrus para matrizes quadradas de ordem 3, ou seja, se d é o determinante da matriz quadrada de ordem 3, então se observa:

$$d = a_{11} \cdot (a_{22} \cdot a_{33} - a_{23} \cdot a_{32}) - a_{12} \cdot (a_{21} \cdot a_{33} - a_{23} \cdot a_{31}) + a_{13} \cdot (a_{21} \cdot a_{32} - a_{22} \cdot a_{31})$$

Produzindo um código que utilize esta regra para calcular o determinante da matriz inserida, foi gerado:

```
d = m[0][0] * (m[1][1] * m[2][2] - m[1][2] * m[2][1]) -
↪ m[0][1] * (m[1][0] * m[2][2] - m[1][2] * m[2][0]) +
↪ m[0][2] * (m[1][0] * m[2][1] - m[1][1] * m[2][0])
```

Uma observação deve ser feita sobre a relação aos índices utilizados na localização de elementos de uma matriz e a localização de elementos de uma lista em *Python*, os índices nas matrizes iniciam em 1 já na linguagem de programação que estamos utilizando começamos do zero.

Com o valor do determinante calculado, deve-se avaliar. Caso ele seja igual a zero, a matriz não irá admitir inversa. Caso contrário, ela será inversível. Sintetizando esta definição em um código, é apresentada a seguinte linha de comando:

```

if d != 0:
    print('A matriz possui inversa.')
else:
    print('A matriz NÃO possui inversa.')

```

Reunindo os códigos produzidos em cada etapa do desenvolvimento da Aplicação 3.7, verifica-se:

```

1 m = [[0, 0, 0],[0, 0, 0],[0, 0, 0]]
2
3 for i in range(0, 3): # Quantidade de linhas da matriz
4     for j in range(0, 3): # Quantidade de colunas da matriz
5         m[i][j] = float(input(f'Digite o valor da posição
6             ↪ a{i + 1}{j + 1}: '))
7
8 d = m[0][0] * (m[1][1] * m[2][2] - m[1][2]*m[2][1]) -
9     ↪ m[0][1] * (m[1][0] * m[2][2] - m[1][2] * m[2][0]) +
10    ↪ m[0][2] * (m[1][0] * m[2][1] - m[1][1] * m[2][0])
11
12 if d != 0:
13     print('A matriz possui inversa.')
14 else:
15     print('A matriz NÃO possui inversa.')

```

Para finalizar a seção, será desenvolvida uma aplicação que resolva um sistema de equações com três equações e três incógnitas. O enunciado desta aplicação segue abaixo:

Aplicação 3.8

Criar um programa que o usuário forneça a matriz incompleta (ordem 3) do sistema, a matriz coluna formada pelos termos independentes das equações do sistema e resolva o sistema utilizando a regra de Cramer.

Antes do início do desenvolvimento da aplicação utilizando a linguagem *Python*, é necessário abrir uma discussão sobre algumas particularidades que versam a resolução de sistemas utilizando a regra de Cramer.

A seguir, é apresentada uma tabela dos questionamentos a serem levantados e as respostas esperadas dos alunos que irão desenvolver este programa.

Processo	
Perguntas:	Respostas esperadas:
Um sistema possui outra forma de ser representado?	Sim, sua forma matricial.
Qual é a característica da matriz incompleta de um sistema?	A matriz incompleta de um sistema é formada pelos coeficientes que acompanham as incógnitas do sistema.
Qual é a característica da matriz completa?	A matriz completa é formada pelos coeficientes das incógnitas do sistema e pelos termos independentes de cada equação.
Qual a definição de matriz coluna?	Matriz que possui apenas uma coluna.
O que podemos afirmar sobre o sistema caso o determinante da sua matriz incompleta seja igual a zero?	O sistema será possível e terá solução única.

Qual o enunciado da regra de Cramer?	<p>Dado um sistema linear com n equações e n incógnitas, se o determinante da matriz incompleta do sistema é diferente de zero, teremos a solução $(x_1, x_2, x_3, \dots, x_n)$, onde $\alpha_i = \frac{D_i}{D}$, em que D_i é o determinante da matriz obtida da matriz incompleta do sistema, substituindo-se a i-ésima coluna pela coluna dos termos independentes das equações do sistema.</p>
--------------------------------------	--

Como será trabalhado com um sistema de equações de ordem 3, a matriz incompleta será quadrada de ordem 3. Assumindo que os alunos saibam extrair a matriz incompleta do sistema será criada uma matriz suporte com o papel de receber os elementos da matriz incompleta do sistema. Assim, percebe-se o seguinte código:

```
m = [[0, 0, 0],[0, 0, 0],[0, 0, 0]]

for i in range(0, 3): # Quantidade de linhas da matriz
    for j in range(0, 3): # Quantidade de colunas da matriz
        m[i][j] = float(input(f'Digite o valor da posição
        ↪ a{i + 1}{j + 1}: '))
```

A matriz coluna que recebeu os termos independentes das equações do sistema precisa ser inserida pelos usuários. Assim, utilizando do mesmo raciocínio anterior, será criada uma matriz suporte com o objetivo de armazenar esses valores. Para concluir esta etapa do desenvolvimento obtém-se o comando:

```
p = [0,0,0]
```

```

for i in range(0, 3):
    p[i] = float(input(f'Digite o valor da posição b{i +
        ↪ 1}1: '))

```

Com os valores da matriz incompleta do sistema e da matriz dos termos independentes das equações, é necessário verificar se o determinante da matriz incompleta é diferente de zero para que a regra de Cramer possa ser utilizada. Dessa forma, chega-se ao seguinte código que foi obtido no desenvolvimento da Aplicação 3.7, utilizando a regra de Sarrus.

```

d = m[0][0] * (m[1][1] * m[2][2] - m[1][2]*m[2][1]) -
    ↪ m[0][1] * (m[1][0] * m[2][2] - m[1][2] * m[2][0]) +
    ↪ m[0][2] * (m[1][0] * m[2][1] - m[1][1] * m[2][0])

```

Para verificar se o determinante é diferente de zero e dar início ao processo de aplicação da regra de Cramer, precisa-se da seguinte condicional:

```

if d != 0:
    # Código para o uso da regra de Cramer.
else:
    print('O Sistema não possui solução única ou não possui
        ↪ solução.')

```

Nesta etapa, as colunas 1, 2 e 3 da matriz incompleta serão substituídas pelos valores da matriz que possui os termos independentes das equações. Assim, serão criadas 6 novas matrizes matrizes suportes, que são clones da matriz incompleta e que receberão a troca dos valores desejados. Como já mencionado no início da seção, é preciso de conhecimento na linguagem *Python*, pois, neste processo, utiliza-se uma manobra chamada *deep copy* para criar as matrizes suportes que também são clones.

É necessário importar a biblioteca *copy* para que seja feita a *deep copy* e assim garantir que a qualquer modificação feita nesta cópia, a matriz original não será alterada. Assim, este procedimento se consolida da seguinte forma em *Python*:

```

m_a = copy.deepcopy(m)
m_b = copy.deepcopy(m)
m_c = copy.deepcopy(m)
p_a = copy.deepcopy(p)
p_b = copy.deepcopy(p)
p_c = copy.deepcopy(p)

# Substituindo as colunas pela matriz dos termos
↳ independentes.

for i in range(0,3):
    m_a[i][0] = p_a[i]
    m_b[i][1] = p_b[i]
    m_c[i][2] = p_c[i]

```

Utilizando a regra de Sarrus, serão calculados os três determinantes das matrizes que foram criadas ao substituir a primeira, segunda e terceira coluna pelos termos independentes das equações.

```

det_a = m_a[0][0] * (m_a[1][1] * m_a[2][2] - m_a[1][2] *
↳ m_a[2][1]) - m_a[0][1] * (m_a[1][0] * m_a[2][2] -
↳ m_a[1][2] * m_a[2][0]) + m_a[0][2] * (m_a[1][0] *
↳ m_a[2][1] - m_a[1][1] * m_a[2][0])
det_b = m_b[0][0] * (m_b[1][1] * m_b[2][2] - m_b[1][2] *
↳ m_b[2][1]) - m_b[0][1] * (m_b[1][0] * m_b[2][2] -
↳ m_b[1][2] * m_b[2][0]) + m_b[0][2] * (m_b[1][0] *
↳ m_b[2][1] - m_b[1][1] * m_b[2][0])
det_c = m_c[0][0] * (m_c[1][1] * m_c[2][2] - m_c[1][2] *
↳ m_c[2][1]) - m_c[0][1] * (m_c[1][0] * m_c[2][2] -
↳ m_c[1][2] * m_c[2][0]) + m_c[0][2] * (m_c[1][0] *
↳ m_c[2][1] - m_c[1][1] * m_c[2][0])

```

Com os valores dos determinantes necessários, agora é momento de uti-

lizar a regra de Cramer para encontrar a solução do sistema, o qual inicialmente o usuário inseriu a matriz incompleta e a matriz com os termos independentes das equações.

Utilizando a regra de Cramer, calculando a razão entre o determinante da matriz obtida pela substituição da coluna pelo determinante da matriz incompleta do sistema e fazendo isto para cada matriz que foi gerada pela substituição de coluna, foi produzido o seguinte código que apresenta a solução do sistema:

```
sol_a = det_a / det
sol_b = det_b / det
sol_c = det_c / det

print(f'A solução do sistema é: x = {sol_a}, y = {sol_b} e z
↳ = {sol_c}')
```

Para apresentar o programa completo, foram reunidas todas as etapas do raciocínio e como produto verifica-se a seguinte linha de comando:

```
1  import copy
2  m = [[0, 0, 0],[0, 0, 0],[0, 0, 0]]
3
4  for i in range(0, 3): # Quantidade de linhas da matriz
5      for j in range(0, 3): # Quantidade de colunas da matriz
6          m[i][j] = float(input(f'Digite o valor da posição
↳ a{i + 1}{j + 1}: '))
7
8  p = [0,0,0]
9
10 for i in range(0, 3):
11     p[i] = float(input(f'Digite o valor da posição b{i +
↳ 1}1: '))
12
```

```

13 det = m[0][0] * (m[1][1] * m[2][2] - m[1][2]*m[2][1]) -
    ↪ m[0][1] * (m[1][0] * m[2][2] - m[1][2] * m[2][0]) +
    ↪ m[0][2] * (m[1][0] * m[2][1] - m[1][1] * m[2][0])
14
15 if det != 0:
16
17     m_a = copy.deepcopy(m)
18     m_b = copy.deepcopy(m)
19     m_c = copy.deepcopy(m)
20     p_a = copy.deepcopy(p)
21     p_b = copy.deepcopy(p)
22     p_c = copy.deepcopy(p)
23
24     # Substituindo as colunas pela matriz dos termos
    ↪ independentes.
25     for i in range(0,3):
26         m_a[i][0] = p_a[i]
27         m_b[i][1] = p_b[i]
28         m_c[i][2] = p_c[i]
29
30     det_a = m_a[0][0] * (m_a[1][1] * m_a[2][2] - m_a[1][2] *
    ↪ m_a[2][1]) - m_a[0][1] * (m_a[1][0] * m_a[2][2] -
    ↪ m_a[1][2] * m_a[2][0]) + m_a[0][2] * (m_a[1][0] *
    ↪ m_a[2][1] - m_a[1][1] * m_a[2][0])
31     det_b = m_b[0][0] * (m_b[1][1] * m_b[2][2] - m_b[1][2] *
    ↪ m_b[2][1]) - m_b[0][1] * (m_b[1][0] * m_b[2][2] -
    ↪ m_b[1][2] * m_b[2][0]) + m_b[0][2] * (m_b[1][0] *
    ↪ m_b[2][1] - m_b[1][1] * m_b[2][0])

```

```

32     det_c = m_c[0][0] * (m_c[1][1] * m_c[2][2] - m_c[1][2] *
    ↪ m_c[2][1]) - m_c[0][1] * (m_c[1][0] * m_c[2][2] -
    ↪ m_c[1][2] * m_c[2][0]) + m_c[0][2] * (m_c[1][0] *
    ↪ m_c[2][1] - m_c[1][1] * m_c[2][0])
33
34     sol_a = det_a / det
35     sol_b = det_b / det
36     sol_c = det_c / det
37
38     print(f'A solução do sistema é: x = {sol_a}, y = {sol_b}
    ↪ e z = {sol_c}')
39
40 else:
41     print('O Sistema não possui solução única ou não possui
    ↪ solução.')
```

3.2 Equações do 2º Grau

Reconhecer e entender o problema a ser resolvido é de extrema importância, pois deve-se traçar um percurso entre o ponto de partida e os objetivos que se deseja alcançar, estruturando um protocolo para a resolução de uma equação do segundo grau.

Neste sentido, há uma lista de itens ordenados que deverão ser investigados durante o processo da produção do programa. Para guiar esta investigação, sugere-se a elaboração da tabela a seguir com os questionamentos e as respostas esperadas para cada questão.

Processo	
Perguntas:	Respostas esperadas:
Qual a forma geral de uma equação do segundo grau?	$ax^2 + bx + c = 0.$
Quais são os coeficientes de uma equação do segundo grau?	Os coeficientes são a , b e c .

Como calcular o discriminante (Δ)?	$\Delta = b^2 - 4ac.$
Qual a relação do valor do discriminante com a quantidade de soluções de uma equação do segundo grau?	Se $\Delta < 0$, a equação não possui solução real; se $\Delta = 0$, a equação possui duas soluções reais e iguais, ou seja, apenas uma solução; se $\Delta > 0$, a equação possui duas soluções reais e diferentes.
Após encontrar o valor do discriminante, como determinar efetivamente as soluções da equação?	Utiliza-se a fórmula resolvente para equações do 2º grau $\frac{-b \pm \sqrt{\Delta}}{2a}$. Se $\Delta = 0$, essa fórmula se reduz apenas a $-\frac{b}{2a}$.

Para resolver uma equação do segundo grau, deve-se reconhecer sua forma geral reduzida $ax^2 + bx + c = 0$, para identificar facilmente os coeficientes da equação. A fim de que o computador inicie o processo de resolução da equação, ele precisa ser informado sobre os valores desses coeficientes. Assim, espera-se que o código do programa apresente as seguintes linhas:

```
a = float(input('Digite o coeficiente de x²: '))
b = float(input('Digite o coeficiente de x: '))
c = float(input('Digite o termo independente: '))
```

Na próxima etapa, é imprescindível que se verifique a condição de existência de uma equação do segundo grau, ou seja, verificar se o valor do coeficiente a é diferente de 0. As linhas a seguir retratam tal condição de existência.

```
if a != 0:
    print('A equação é do 2º grau.')
else:
    print('A equação não é do 2º grau.')
```

Finalizadas estas etapas, deve-se calcular o valor do discriminante a fim de determinar a quantidade de soluções. A solução de uma equação do segundo grau é um valor que faz com que a igualdade $ax^2 + bx + c = 0$ seja verdadeira. Tais equações possuem um protocolo definido para a quantificação e obtenção de suas soluções, conforme descrito na tabela anterior. Com isso, o programa deve apresentar a seguinte linha de comando para que o cálculo seja efetuado a partir dos valores anteriormente inseridos.

```
delta = (b ** 2) - (4 * a * c)
```

Em seguida, deve-se iniciar uma discussão sobre a quantidade de soluções em função do valor encontrado para o discriminante, determinando assim três caminhos a serem seguidos descritos por condicionais. Os comandos abaixo são relativos a essas condições.

```
if delta < 0:
    print('A equação não possui soluções reais')
elif delta == 0:
    print('A equação possui duas soluções reais e iguais')
else:
    print('A equação possui duas soluções reais e diferentes')
```

Com o debate sobre a quantidade de soluções da equação do segundo grau finalizado, agora é o momento de calcular essas soluções. Para isso, será feito o uso da fórmula resolvente de equações do 2º grau caso o discriminante seja positivo, por meio dos comandos inseridos nas linhas a seguir.

```
x = (- b + (delta ** (1 / 2))) / (2 * a)
y = (- b - (delta ** (1 / 2))) / (2 * a)
```

Caso o discriminante seja igual a 0, é válido ressaltar que para calcular a solução podemos utilizar $\left(-\frac{b}{2a}\right)$, pelo fato de que $\sqrt{0} = 0$. Assim, será implementado:

$$x = -b / (2 * a)$$

Em linhas gerais, o processo de aplicação de conhecimentos prévios para a produção de um programa que resolva uma equação do segundo grau, pode ser observado por meio do seguinte código:

```
1 # -----
2 # CALCULADORA PARA RESOLUÇÃO DE UMA EQUAÇÃO DO 2º GRAU
3 # -----
4
5 a = float(input('Digite o coeficiente de x²: ')) #
   ↳ Coeficiente que acompanha x²
6 b = float(input('Digite o coeficiente de x: ')) #
   ↳ Coeficiente que acompanha x
7 c = float(input('Digite o termo independente: ')) # Termo
   ↳ independente
8
9 if a != 0: # Condição de existência de uma equação do 2º
   ↳ grau
10
11     delta = (b ** 2) - (4 * a * c)
12
13     if delta < 0: # Verificando se o discriminante é
   ↳ negativo.
14
15         print('A equação não possui soluções reais.')
16
17     elif delta == 0: # Verificando se o discriminante é
   ↳ nulo.
18
19         x = -b / (2 * a)
```

```

20     print(f'A equação possui duas soluções iguais a {x}
21         ↪ .')
22
23     else: # Verificando se o discriminante é positivo
24
25         x = (- b + (delta ** (1 / 2))) / (2 * a)
26         y = (- b - (delta ** (1 / 2))) / (2 * a)
27         print(f'A equação possui duas soluções reais a
28             ↪ primeira é igual a {x} e a segunda'
29                 f'igual a {y}.')
30
31     else: # Verificando que não é uma equação do 2º grau.
32         print('A equação não é do 2º grau.')

```

3.3 Teorema de Pitágoras

O Teorema de Pitágoras é uma relação matemática que envolve as medidas dos lados de um triângulo retângulo.

Nesta atividade, será apresentada uma dinâmica que envolve a recíproca do Teorema de Pitágoras. De forma mais específica, será elaborada uma estratégia para programar o computador para verificar se três números formam um triângulo retângulo. Os questionamentos descritos na tabela a seguir norteiam tal processo:

Processo	
Perguntas:	Respostas esperadas:
Quantos lados possui um triângulo?	3 lados.
Qual a restrição para as medidas dos lados de um triângulo?	Todas as medidas devem ser positivas.
Qual a condição de existência de um triângulo?	Cada lado de um triângulo possui medida menor que a soma dos outros dois lados.

O que é um triângulo retângulo?	Triângulo que possui um ângulo reto.
Qual o enunciado do Teorema de Pitágoras?	Em um triângulo retângulo, a soma dos quadrados dos catetos é igual à hipotenusa ao quadrado.
Qual é a recíproca do Teorema de Pitágoras?	Se a soma dos quadrados dos catetos é igual ao quadrado da hipotenusa, então o triângulo é retângulo.

A definição de triângulo, a análise de seus elementos e a relação entre estes devem ser discutidas com o intuito de reconhecer o polígono, que é o objeto central para o desenvolvimento do programa com foco em reconhecê-lo como um triângulo retângulo ou não.

Durante o debate que inicia o processo de produção do programa, é preciso verificar, obrigatoriamente, que o triângulo é um polígono de três lados. Dessa forma, ele deverá fornecer três medidas de comprimento que serão fundamentais para o desenvolvimento da atividade, visto que são esses valores que validam a tese da recíproca do Teorema de Pitágoras.

De imediato, verifica-se que o panorama desta primeira discussão pode ser expresso na seguinte linha de comando:

```
a = float(input('Digite a medida do primeiro lado do
↳ triângulo:'))
b = float(input('Digite a medida do segundo lado do
↳ triângulo:'))
c = float(input('Digite a medida do terceiro lado do
↳ triângulo:'))
```

Finalizado este momento de reconhecimento do triângulo e do comprimento de cada um de seus lados, é de suma importância verificar agora se a condição de existência de um triângulo será satisfeita, lembrando que os valores de a , b e c são positivos.

Para verificar se a condição de existência está satisfeita é necessário que três desigualdades envolvendo os comprimentos dos lados do triângulo

sejam verificadas. As condicionais serão formuladas a partir da seguinte sentença - "A medida de um lado do triângulo sempre será menor que a soma das medidas dos outros lados."

Podemos convalidar a discussão no código seguinte.

```
if ((a > 0) and (b > 0) and (c > 0)) and ((a < b + c) and (b
↳ < a + c) and (c < a + b)):

else:
    print('ERRO: Não existe triângulo quando a medida de
↳ pelo menos um de seus lados é negativa ou quando as
↳ medidas não satisfazem a condição de existência de
↳ um triângulo.')
```

Verificada a existência do triângulo, é o momento de averiguar se ele pode ser classificado como um triângulo retângulo. Existem afirmações que comprovam se um triângulo é retângulo, mas o que interessa, neste momento, é a recíproca do Teorema de Pitágoras.

Para validar a recíproca do Teorema de Pitágoras é necessário utilizar os dados inseridos para verificar se está satisfeita a hipótese de tal teorema, ou seja, "A hipotenusa ao quadrado deve ser igual a soma dos quadrados dos catetos".

Para concluir esta parte do debate, o raciocínio sobre as condições gera a seguinte linha de comando:

```
m = [a, b, c]
lado_a = max(m)
lado_b = min(m)
lado_c = 0
for i in range(0, 3):
    if m[i] != lado_a and m[i] != lado_b:
        lado_c = m[i]
```

```
if (lado_a ** 2) == (lado_b ** 2) + (lado_c ** 2):
    print('O triângulo é retângulo.')

else:
    print('O triângulo não é retângulo.')
```

Em seguida e não menos importante, fixar a definição de que o maior lado de um triângulo caso ela seja retângulo é denominado hipotenusa.

Para sintetizar toda a discussão da seção, organizando as conclusões e apresentando o programa produzido, verifica-se o seguinte código.

```
1 # -----
2 # VERIFICANDO SE UM TRIÂNGULO É RETÂNGULO OU NÃO
3 # -----
4 a = float(input('Digite a medida do primeiro lado do
   ↳ triângulo:'))
5 b = float(input('Digite a medida do segundo lado do
   ↳ triângulo:'))
6 c = float(input('Digite a medida do terceiro lado do
   ↳ triângulo:'))
7 m = [a, b, c]
8 lado_a = max(m)
9 lado_b = min(m)
10 lado_c = 0
11
12 if ((a > 0) and (b > 0) and (c > 0)) and ((a < b + c) and (b
   ↳ < a + c) and (c < a + b)):
13
14     for i in range(0, 3):
15         if m[i] != lado_a and m[i] != lado_b:
16             lado_c = m[i]
17
18     if (lado_a ** 2) == (lado_b ** 2) + (lado_c ** 2):
```

```

19         print('O triângulo é retângulo.')
20
21     else:
22         print('O triângulo não é retângulo.')
23
24 else:
25     print('ERRO: Não existe triângulo quando a medida de
        ↳ pelo menos um de seus lados é negativa ou quando as
        ↳ medidas não satisfazem a condição de existência de
        ↳ um triângulo.')
```

3.4 Matemática Financeira

A presente seção tem por objetivo utilizar conhecimentos sobre matemática financeira para produzir um programa que resolva situações que são vivenciadas no cotidiano.

Uma sequência será desenvolvida a partir dos conceitos de juros simples e compostos, finalizando a seção e o capítulo com o cálculo de parcelas.

Antes de iniciar a programação, uma discussão sobre os principais termos da matemática financeira deve ser iniciada a partir de perguntas e respostas esperadas. É possível observar, na tabela a seguir, os questionamentos a serem levantados.

Processo	
Perguntas:	Respostas esperadas:
Qual é o nome dado ao valor monetário que uma pessoa empresta a outra durante um certo tempo?	Capital ou principal e é representado pela letra <i>C</i> .
Qual o nome do valor que o empréstador cobra pelo uso do dinheiro?	Juros e é indicado pela letra <i>J</i> .

Qual o significado da taxa de juros?	A taxa de juros é expressa como porcentagem do capital, ela representa os juros gerado por um capital numa certa unidade de tempo. A letra <i>i</i> de <i>interest</i> que significa juros em inglês é utilizada para representar taxa de juros.
Como calcular o juros gerados a partir de um capital no período da taxa?	Podemos calcular, utilizando a fórmula $J = C \cdot i$.
Qual o nome dado ao valor que representa a soma do capital e juros gerados em um certo período?	Este valor é chamado de montante e utiliza-se a letra <i>M</i> para representá-lo. ($M = C + J$)
Qual a definição de juros simples?	Os juros simples são o resultado obtido pela multiplicação do capital pela taxa e pelo prazo da aplicação ou empréstimo, é dado pela fórmula $J = C \cdot i \cdot t$
Qual a definição de juros compostos?	Os juros compostos são calculados considerando a atualização, ou seja, ele incide sobre o juros que foi gerado no período anterior.

Uma sugestão para aplicar o questionamento anterior, seria a discussão sobre a diferença entre juros simples e juros composto, discutindo que os juros simples são baseados no total do valor de um empréstimo ou depósito, enquanto os juros compostos se baseiam nesse mesmo total mais os juros que se acumulam em cada período.

Com os questionamentos anteriores efetuados, o próximo passo é apresentar um problema particular e sua solução, por exemplo:

Exemplo 3.1

(IEZZI, 2013) Um capital de R\$12.000,00 foi aplicado durante 3 meses à taxa de 5%*a.t.*(ao trimestre). Vamos calcular os juros e o montante recebidos após 3 meses. Em reais, após 3 meses, os juros recebidos foram:

$$J = 12000 \cdot (0,05) = 600$$

Assim, o montante recebido, em reais, foi:

$$M = 12000 + 600 = 12600$$

Após reconhecer o uso dos conceitos fundamentais de matemática financeira a partir de um exemplo particular, será apresentado a seguinte atividade genérica para ser desenvolvida em *Python*.

Aplicação 3.9

Criar um programa que o usuário forneça o capital, a taxa de juros e o tempo de aplicação ou empréstimo a juros simples e calcule o montante após o tempo informado.

O processo inicial de desenvolvimento do programa é a identificação dos valores que são necessários para o cálculo do montante, é válido lembrar que a unidade de tempo utilizada deve estar de acordo com a unidade da taxa de juros, sendo assim, há o seguinte código:

```
capital = float(input('Digite o valor do capital: '))
taxa = float(input('Digite o valor da taxa em decimal: '))
tempo = float(input('Digite o tempo de aplicação/empréstimo:
↳ '))
```

Conhecidos os valores do capital, taxa e tempo de aplicação ou empréstimo, deve-se calcular o valor do juros que o capital gerou utilizando a fórmula $J = C \cdot i \cdot t$, traduzindo para a linguagem *Python*, obtem-se:

```
juros = capital * taxa * juro
```

Para calcular o montante, vamos utiliza-se a fórmula discutida no início da seção $M = C + J$, assim observa-se:

```
montante = capital + juros
```

Reunindo todo o raciocínio para a produção do programa, chega-se ao seguinte código:

```
1 capital = float(input('Digite o valor do capital: '))
2 taxa = float(input('Digite o valor da taxa em decimal: '))
3 tempo = float(input('Digite o tempo de aplicação/empréstimo:
↳ '))
4
5 juros = capital * taxa * tempo
6
7 montante = capital + juros
8
9 print(f'0 montante da aplicação/empréstimo de R${capital} a
↳ uma taxa de {taxa}% gera após {tempo} meses é igual a
↳ R${montante}.')
```

A diferença entre a próxima aplicação e a anterior é o cálculo em um sistema de juros compostos, como pode ser visto no enunciado:

Aplicação 3.10

Criar um programa que o usuário forneça o capital, a taxa de juros e o tempo de aplicação ou empréstimo a juros compostos e calcule o montante após o tempo informado.

Como na aplicação anterior, o usuário precisa fornecer o valor do capital, taxa e tempo, assim percebe-se o código:

```
capital = float(input('Digite o valor do capital: '))
taxa = float(input('Digite o valor da taxa em decimal: '))
tempo = float(input('Digite o tempo de aplicação/empréstimo:
↳ '))
```

Para desenvolver a próxima etapa do programa deve-se calcular o mon-

tante da aplicação/empréstimo no regime de juros compostos, assim, utiliza-se a fórmula $M = C(1 + i)^t$. Seguindo o raciocínio, verifica-se o código:

```
montante = capital * (1 + taxa) ** tempo
```

Organizando as conclusões obtidas anteriormente obt-em-se o seguinte programa:

```
1 capital = float(input('Digite o valor do capital: '))
2 taxa = float(input('Digite o valor da taxa em decimal: '))
3 tempo = float(input('Digite o tempo de aplicação/empréstimo:
↳ '))
4
5 montante = capital * (1 + taxa) ** tempo
6
7 print(f'O montante da aplicação/empréstimo de R${capital} a
↳ uma taxa de {taxa}% gera após {tempo} meses é igual a
↳ R${montante}.')
```

Para a próxima atividade, os alunos que forem resolvê-la deverão ter conhecimento sobre sequência uniforme de pagamentos. Assim, o questionamento a ser levantado deve ser:

Processo	
Perguntas:	Respostas esperadas:
O que é uma sequência uniforme de pagamentos?	É um conjunto formado pelas prestações de um financiamento.
Como calcular o valor das prestações de um capital financiado (V) que deve ser pago em prestações iguais (R) a taxa de juros compostos (i) por um determinado período de tempo (n)?	Utilizando a fórmula $V = R \cdot \frac{(1 + i)^n - 1}{(1 + i)^n \cdot i}$.

A seguir, tem-se um exemplo particular de aplicação de sequência uniforme de pagamentos para que auxilie o aluno a entender a situação e consiga elaborar um plano de ação para produzir o programa que resolve a próxima aplicação.

Exemplo 3.2

(IEZZI, 2013) Um banco concedeu um empréstimo para uma pessoa adquirir um carro. O pagamento deveria ser feito em 12 prestações mensais de R\$1400,00 cada uma, sem entrada. Qual o valor do empréstimo, sabendo-se que a taxa de juros compostos cobrada pelo banco foi de 3% a.m.?

Solução:

O empréstimo deve ser pago em 12 prestações mensais e iguais. Assim, tem-se $R = 1400$, $n = 12$ e $i = 3\%$.

O valor do empréstimo corresponde ao valor atual desses pagamentos, que, conforme a fórmula dada é igual a:

$$V = 1400 \cdot \frac{(1,03)^{12} - 1}{(1,03)^{12} \cdot 0,03}$$
$$V = 13.935,61$$

Conclui-se assim que o valor emprestado pelo banco foi de R\$13935,61.

Observados os detalhes do exemplo anterior sobre o cálculo do valor das prestações a serem pagas em um financiamento, tem-se a seguinte aplicação a ser desenvolvida posteriormente.

Aplicação 3.11

Criar um programa que o usuário forneça o valor do bem financiado, a quantidade de prestações, a taxa de juros, se haverá entrada, calcule o valor de cada prestação que serão iguais e a quantidade de juros cobrada pelo financiador.

Em um primeiro momento, o aluno precisa perceber que para iniciar o

desenvolvimento do programa, ele precisa receber os valores do financiamento, entrada, quantidade de parcelas e taxa de juros, que serão utilizados na fórmula descrita na tabela sobre as perguntas e respostas esperadas. Assim, há o seguinte código:

```
valor_fin = float(input('Digite o valor financiado: '))
entrada = float(input('Digite o valor da entrada: '))
prestacoes = int(input('Digite a quantidade de prestações:
↳ '))
taxa = float(input('Utilizando números decimais digite a
↳ taxa de juros: '))
```

Para calcular o valor de cada parcela, a fórmula $V = R \cdot \frac{(1+i)^n - 1}{(1+i)^n \cdot i}$ deve ser manipulada, ou seja, o R que representa o valor da parcela deve ser isolado já que todos os outros valores são conhecidos.

Assim, verifica-se:

$$V = R \cdot \frac{(1+i)^n - 1}{(1+i)^n \cdot i} \Rightarrow R = V \cdot \frac{(1+i)^n \cdot i}{(1+i)^n - 1}$$

Com a fórmula definida adequadamente para ser utilizada na aplicação, observa-se o seguinte código:

```
valor_pres = valor_fin * (((1 + taxa) ** prestacoes) *
↳ taxa) / (((1 + taxa) ** prestacoes) - 1))
```

A proposta da aplicação também inclui o valor que foi pago de juros durante todo o processo, para isso deve-se somar todas as parcelas e do resultado subtrair o valor que foi financiado. Como resultado, verificam-se os juros que foram pagos durante todo o período de pagamento.

Para calcular os juros, há o seguinte código:

```
juros_pg = (valor_pres * prestacoes) - valor_fin
```

Reunindo todo o raciocínio desenvolvido na produção do programa mais uma mensagem para o usuário sobre o valor da parcela e a quantidade

de juros que será paga durante todo o período do financiamento, tem-se o seguinte programa finalizado:

```
1 valor_fin = float(input('Digite o valor financiado: '))
2 entrada = float(input('Digite o valor da entrada: '))
3 prestacoes = int(input('Digite a quantidade de prestações:
↳ '))
4 taxa = float(input('Utilizando números decimais digite a
↳ taxa de juros: '))
5
6 # Calculando o valor de cada prestação.
7
8 valor_pres = valor_fin * (((1 + taxa) ** prestacoes) *
↳ taxa) / (((1 + taxa) ** prestacoes) - 1))
9
10 juros_pg = (valor_pres * prestacoes) - valor_fin
11
12 print(f'O valor de cada prestação será de R${valor_pres}. A
↳ quantidade de juros pagos durante todo o processo é
↳ R${juros_pg}.')
```

Capítulo 4

Considerações Finais

Desenvolver o conhecimento matemático é algo complexo, é um desafio diário tanto para alunos quanto para professores. No ensino da matemática, é necessário transformar algo abstrato e complexo em um conteúdo agradável e, ao mesmo tempo permitir que os alunos e professores extrapolem seus limites. Uma oportunidade de fazer com que esse fato se concretize é realizar a mediação entre conteúdos matemáticos e o uso de tecnologias, neste caso a elaboração de programas de computadores.

A pesquisa encontrou sua essência no fato de que trabalhar com linguagens computacionais no ensino fundamental é uma possibilidade para a nova sala de aula. E a escolha da linguagem Python para o desenvolvimento das propostas de atividades foi essencial, por sua sintaxe simples e que poderia ser utilizada para o público do ensino básico.

Com o objetivo de criar atividades que envolvam conceitos do ensino básico utilizando programação, foram utilizados conhecimentos em Python para transformar problemas cotidianos de alunos de ensino básico em um processo de questionamento sistemático de conteúdos matemáticos que conduzirá a produção de um programa solução.

Em experiências como professores de matemática que compreendem de forma básica a programação em Python, em momentos de fuga da aula tradicional, podem ser elaboradas pequenas atividades testes para exibir aos alunos a possibilidade de transformar todo aquele conteúdo trabalhado em aula e nas resolução de exercícios, em programas de computadores.

Considerações Finais

Durante estes poucos momentos, percebe-se euforia na maioria dos alunos, fato este que estimula a desenvolver trabalhos e pesquisas desta natureza.

Em um primeiro momento, as atividades podem gerar uma resistência inicial em alguns alunos pelo fato de que a linguagem não possui uma interface lúdica. Para alguns, o processo pode exigir tempo, pois as atividades envolvem construções cognitivas e cada aluno possui sua forma de se desenvolver.

O profissional que decidir utilizar estas atividades deve dominar na linguagem pelo menos as ferramentas que serão utilizadas, para que o momento de dúvidas seja leve, porém o ambiente da programação é bem amplo e sugere vários caminhos para a resolução de um mesmo problema.

"Foi possível considerar que o uso de uma linguagem de programação, que possua uma sintaxe simples e que seja fácil compreensão, pode contribuir no complemento de um conhecimento pré-existente de uma criança, bastando apenas que haja a colaboração entre o professor de programação e o professor da disciplina de matemática, para que sejam desenvolvidos exemplos capazes de serem compreendidos."(PESENTE, 2019)

Existem trabalhos e pesquisas que reafirmam a importância destas atividades e que obtiveram alguns resultados satisfatórios e impulsionantes à novas práticas.

"Os códigos computacionais permitiram que os alunos respondessem aos exercícios com maior número de detalhes, separando o valor por valor em variáveis, para que o computador processasse todas as informações e desse o resultado correto."(PESENTE, 2019)

Esta pesquisa tem como objetivo motivar futuros trabalhos nesta linha de desenvolvimento de atividades que envolvam matemática e programação. É importante incentivar alunos e professores ao domínio máximo da linguagem Python, linguagem esta que nos fornece uma enorme quantidade de bibliotecas para trabalhar com diversos conteúdos (gráficos, estatística, probabilidade e etc), assim podendo ser elaboradas novas propostas de atividades que requerem comandos mais complexos da linguagem.

Por meio desta pesquisa, avalia-se que existe a possibilidade de adap-

tações, ou seja, relacionar outros conteúdos matemáticos que não foram citados neste texto com a programação de computadores.

Não se pode afirmar que esta pesquisa possui um final, pois, a aplicação das atividades propostas poderão trazer novos trabalhos por parte de professores que aplicarão as atividades e pesquisadores que utilizarão este texto como incentivo e motivação para futuras produções científicas, tornando assim esta pesquisa um processo constante de desenvolvimento de novas atividades, novas ferramentas e novos conceitos de uso da programação em sala de aula.

Existe a necessidade de construir novas práticas, e esta pesquisa é uma oportunidade de repensar a formação de professores e as disciplinas das licenciaturas em matemática de forma a inserir a programação em um formato mais didático. Urge apresentar a programação aos futuros professores não apenas como uma ferramenta utilizada em matemática aplicada e computacional, mas sim como uma ferramenta facilitadora no ensino aprendizagem.

Novas atitudes são necessárias, e esta pesquisa apresenta atividades que faz com que o conhecimento matemática dialogue com recursos tecnológicos disponíveis, oferecendo assim novas possibilidades de produção de saberes que transcendem a perspectiva da sala de aula convencional.

Levando-se em conta o que foi observado, é fato que este é um vasto campo de pesquisa, e sempre que um pesquisador desenvolver algo sobre este assunto, irá enfrentar desafios. A tecnologia está presente constantemente no cotidiano, porém está sendo subutilizada, para que isso seja contornado é preciso um planejamento, um estudo multidisciplinar e uma notável mudança de hábitos didáticos para que o processo de ensino e aprendizagem possa ser alterado e comece a caminhar com a tecnologia presente na atualidade numa perspectiva de educação inovadora e transformadora.

Referências Bibliográficas

ALLAN, L. **Como a tecnologia pode ajudar nossas escolas a vencer o Coronavírus?** 2020. Acessado: 9 abr. 2021. Disponível em: <<https://tinyurl.com/y45t7hv5>>.

BORBA, M. C.; PENTEADO, M. G. **Informática e Educação Matemática**. 5. ed. Belo Horizonte: Autêntica, 2003.

BORGES, M. A. F. **Importância da programação para crianças**. 2018. Acessado em 12 abr 2021. Disponível em: <<https://tinyurl.com/cb5emsnm>>.

BRASIL. **Pisa 2018. Relatório Nacional**. [S.l.], 2019.

CARNEIRO, R. F.; PASSOS, C. L. B. A utilização das tecnologias da informação e comunicação nas aulas de matemática: Limites e possibilidades. **Revista Eletrônica de Educação**, v. 8, n. 2, p. 101–119, 2014.

D'AMBROSIO, U. **Etnomatemática: arte ou técnica de explicar ou conhecer**. 4. ed. São Paulo: Ática, 1998.

DANTE, L. R. **Didática da Resolução de Problemas de Matemática**. 2. ed. São Paulo: Ática, 1998.

DEMO, P. **Formação permanente e tecnologias educacionais**. Petrópolis: Editora Vozes, 2006.

EUA. **Presidente (2009-2017:Barack Hussein Obama II). Ciência da computação para todos**. 2016. Acessado: 9 abr. 2021. Disponível em: <<https://tinyurl.com/4zzajz8p>>.

GARBI, G. G. **O Romance das equações algébricas**. São Paulo: Makron Books, 1997.

GUEDES, P. M. **Especialistas em educação analisam: o ano letivo de 2020 está perdido?** 2020. Acessado: 9 abr. 2021. Disponível em: <<https://tinyurl.com/9kwzww22>>.

HELVEY, T. C. **The age of information: an interdisciplinary survey of cybernetics**. Englewood Cliffs, N.J: Educational Technology Publications, 1971.

HENRIQUES, R. **Especialistas em educação analisam: o ano letivo de 2020 está perdido?** 2020. Acessado: 9 abr. 2021. Disponível em: <<https://tinyurl.com/9kwzww22>>.

Referências Bibliográficas

HILTON, A. M. An ethos for the age of cyberculture. **Proceeding Spring Joint Computer Conference**, 1964.

IEZZI, G. **Fundamentos de matemática elementar, 11 : Matemática Comercial, Matemática Financeira e Estatística Descritiva**. São Paulo: Atual, 2013.

KANSKI, V. **Tecnologias e ensino presencial e a distância**. Campinas, SP: Papyrus, 2004.

LEVY, P. **As tecnologias da inteligência: o futuro do pensamento na era da informática**. Rio de Janeiro: Editora 34, 1993.

_____. **Cibercultura**. São Paulo: Editora 34, 1999.

LIBÂNIO, J. C. **Didática**. 19. ed. São Paulo: Cortez, 1994.

LIMA, L. C. Da mecânica do pensamento ao pensamento emancipado da mecânica. **Programa Integrar, Caderno do Professor, Trabalho e Tecnologia**, p. 95–103, 1998. CUT/SP.

MAGDA, S. **O livro didático e a escolarização da leitura**. 2008. Acessado: 12 abr. 2021. Disponível em: <<https://tinyurl.com/jmz3s8a>>.

MARCONDES, G. A. B. **Matemática com Python: Um Guia Prático**. 1. ed. [S.l.]: Novatec, 2018.

NSC, E. **Setor de TI tem crescimento de 310% das vagas no último ano**. 2021. Acessado em 20 abr 2021. Disponível em: <<https://tinyurl.com/2d2ydh4w>>.

PAPERT, S. **A Máquina das Crianças: Repensando a Escola na Era da Informática**. Porto Alegre: Artes Médicas, 1994.

PENTEADO, M. G. **Novos atores, Novos cenários: Discutindo a inserção dos computadores na profissão docente**, in BICUDO, Maria Aparecida V. **Pesquisa em Educação matemática: Concepções e Perspectivas**. São Paulo: Editora Unesp, 1999.

PESENTE, G. M. **O Ensino da Matemática por meio da Linguagem de Programação Python**. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, 2019.

POLYA, G. **A Arte de Resolver Problemas**. 2. ed. Rio de Janeiro: Interciência, 1978. Tradução: Heitor Lisboa de Araújo.

RAMOS, M. N. **Especialistas em educação analisam: o ano letivo de 2020 está perdido?** 2020. Acessado: 9 abr. 2021. Disponível em: <<https://tinyurl.com/9kwzww22>>.

RESNICK, M. **Learn to Code, Code to Learn - EdSurge**. 2013. Acessado: 12 abr. 2021. Disponível em: <<https://tinyurl.com/2zyjmszr>>.

Referências Bibliográficas

ROSSUM, G. V.; JR, F. L. D. **Python tutorial**. [S.l.]: Centrum voor Wiskunde en Informatica Amsterdam, Holanda, 1995.

RÜDIGER, F. **As teorias da cibercultura: perspectivas, questões e autores**. 2. ed. Porto Alegre: Sulina, 2013.

SAVIANI, D. **História das idéias pedagógicas no Brasil**. Campinas, SP: Autores Associados, 2007.

VALENTE, J. **Inteligência artificial e o impacto nos empregos e profissões**. 2020. Acessado em 19 abr 2021. Disponível em: <<https://tinyurl.com/4mmzrsm2>>.

WOLFRAM, C. **Conrad Wolfram: “80% do que se aprende nas aulas de matemática não serve para nada”**. 2017. Acessado em 12 abr 2021. Disponível em: <<https://tinyurl.com/3k4jf2zc>>.