



UNIVERSIDADE DO SUL DE SANTA CATARINA

DIEGO CANQUERINI DE MORAIS

**A LINGUAGEM DE PROGRAMAÇÃO PYTHON
PARA O ENSINO DA MATEMÁTICA**

Florianópolis, 2021

DIEGO CANQUERINI DE MORAIS

**A LINGUAGEM DE PROGRAMAÇÃO PYTHON
PARA O ENSINO DA MATEMÁTICA**

Projeto de Pesquisa para o Trabalho de Conclusão de Curso apresentado ao Curso de Matemática – Licenciatura da Universidade do Sul de Santa Catarina, campus Unisul Virtual, como requisito parcial à obtenção do título de Licenciado (a) em Matemática.

Orientador: Prof. MSc. Dalmo Gomes de Carvalho.

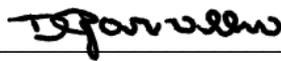
Florianópolis, 2021

DIEGO CANQUERINI DE MORAIS

A LINGUAGEM DE PROGRAMAÇÃO PYTHON PARA O ENSINO DA MATEMÁTICA

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Licenciado em Matemática e aprovado em sua forma final pelo Curso de Graduação em Matemática da Universidade do Sul de Santa Catarina.

Tubarão, 18 de Novembro de 2021.



Professor e orientador Dalmo Gomes de Carvalho, MSc.
Universidade do Sul de Santa Catarina



Prof. Carlos Henrique Hobold, MSc.
Universidade do Sul de Santa Catarina



Prof. Mario Selhorst, MSc.
Universidade do Sul de Santa Catarina

LISTA DE ILUSTRAÇÕES

Figura 1 – IDE Jupyter Notebook.....	24
Figura 2 – Entrada e Saída de dados.....	24
Figura 3 - Interface Gráfica Google Colab.....	25
Figura 4 - Código Python - Operações Básicas.....	29
Figura 5 - Saída do programa.....	29
Figura 6 - Código Python - Operações com potências.....	31
Figura 7 - Saída do programa.....	31
Figura 8 - Código Python - Operações de Radiciação.....	32
Figura 9 - Saída do programa.....	32
Figura 10 - Código Python - Raízes da equação.....	34
Figura 11 - Saída do programa.....	34
Figura 12 - Código Python - Definição e formatação do gráfico.....	35
Figura 13 - Código Python - Definições do gráfico.....	36
Figura 14 - Código Python - Progressão aritmética.....	38
Figura 15 - Saída do programa - Termos da PA.....	38
Figura 16 - Código Python - Progressão geométrica.....	39
Figura 17 - Saída do programa.....	39
Figura 18 - Código Python - Definição do gráfico da PA.....	40
Figura 19 - Código Python - Definição do gráfico da PG.....	41
Figura 20 - Código Python - Definição da função.....	42
Figura 21 - Saída do programa.....	43
Figura 22 - Código Python - Definição da gráfico da função.....	43

LISTA DE GRÁFICOS

Gráfico 1 - Função de 1° grau.....	35
Gráfico 2 - Função de 2°Grau.....	36
Gráfico 3 - Progressão aritmética.....	40
Gráfico 4 - Progressão geométrica.....	41
Gráfico 5 - Função exponencial.....	44

SUMÁRIO

1 CONSIDERAÇÕES INICIAIS.....	7
1.1 TEMA E DELIMITAÇÃO DO TEMA.....	7
1.2 PROBLEMATIZAÇÃO	8
1.3 JUSTIFICATIVAS.....	9
1.4 OBJETIVOS.....	11
1.4.1 Objetivo Geral.....	11
1.4.2 Objetivos Específicos.....	11
1.5 Tipo de pesquisa.....	12
1.6 Estrutura do trabalho.....	13
2 MODELAGEM MATEMÁTICA NA EDUCAÇÃO	15
2.1 Aprendizagem matemática no Ensino Médio.....	15
2.2 Raciocínio lógico e o computador.....	17
2.3 Algoritmo e modelagem matemática.....	18
3 A LINGUAGEM CIENTÍFICA PYTHON.....	20
3.1 O que é uma linguagem científica de programação?.....	20
3.2 <i>Software</i> , diferenças básicas e plataforma de ensino.....	23
3.3 Importando Bibliotecas.....	25
4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS.....	27
4.1 SEQUÊNCIA DIDÁTICAS: OPERAÇÕES BÁSICAS.....	28
4.2 SEQUÊNCIA DIDÁTICA: POTENCIAÇÃO E RADICIAÇÃO.....	30
4.3 SEQUÊNCIA DIDÁTICA: FUNÇÃO DE 1º E 2º GRAU.....	33
4.4 SEQUÊNCIA DIDÁTICA: PA E PG	37
4.5 SEQUÊNCIA DIDÁTICA: FUNÇÃO EXPONENCIAL.....	41
5 CONSIDERAÇÕES FINAIS.....	45
REFERÊNCIAS BIBLIOGRÁFICAS.....	46

1 CONSIDERAÇÕES INICIAIS

O presente documento é um estudo das operações matemática utilizando linguagem de programação Python. Por ter alcançado o topo da popularidade (2021) essa linguagem, que foi criada por um matemático holandês, pode ter muito a somar como ferramenta didática. As habilidades de uso do raciocínio lógico e abstrato são fundamentais para o exercício de diversas profissões e atividades. Tanto a matemática quanto a utilização de linguagens de programação permitem desenvolvê-los. Unir os recursos da solução matemática com Python pode ajudar na construção e prática dessas habilidades, além de apresentar aos estudantes conceitos básicos da linguagem que, segundo pesquisas recentes (2021)¹, tornou-se a mais utilizada do mundo. Entre as vantagens evidentes, temos a simplicidade de expressão dos algoritmos matemáticos, e sua quase inesgotável forma de visualização de dados. A visualização de gráficos e matrizes é facilmente construída utilizando a linguagem. Este estudo, além disso, busca interpretar o fenômeno da linguagem simbólica, através da criação de modelagem matemática, utilizando algoritmos. A ciência de dados ou *Big Data* é um ramo em ascensão considerável na atualidade, e apesar desse trabalho não ser um tratado profundo do tema, visa instigar alunos e docentes a buscar ferramentas contemporâneas, dinâmicas e eficientes, e aplicá-las em sala de aula.

1.1 TEMA E DELIMITAÇÃO DO TEMA

Tema: Contribuição da linguagem de programação *Python* no desenvolvimento de habilidades matemáticas.

Delimitação do tema: Comparar e verificar a eficiência da linguagem *Python* de programação, em ambiente escolar, presencial e remoto, comparando os documentados de diferentes dissertações e estudos sobre o tema, tendo como suporte metodológico a

1 Índice Tiobe. <<https://www.tiobe.com/tiobe-index/>> Acesso 29/10/2021.

modelagem matemática no ensino da matemática, que pode ser fundamento para resolução dos problemas, pensada para nível equivalente aos alunos do Ensino Médio – anos iniciais na disciplina de matemática.

1.2 PROBLEMATIZAÇÃO

Nos dias atuais enfrentamos muitos desafios como educador, e um dos principais desafios dos docentes atualmente é captar a atenção dos alunos, tirando o foco destes dos aparelhos eletrônicos. Em muitas escolas, os aparelhos individuais, celulares, etc., são proibidos em sala de aula, porém na realidade o que ocorre é bem diferente. Em uma aula onde os celulares são expressamente proibidos, o professor consegue manter o foco dos alunos mais tempo sobre o assunto, desconsiderando as distrações típicas de uma sala de aula. Porém, em uma classe onde os alunos têm permissão para utilizar os aparelhos pessoais, durante a aula, o professor se vê em uma posição realmente desfavorável. Aqui podemos considerar a estratégia do ‘fogo contra fogo’ utilizada pelas brigadas de incêndio, ou seja, utilizar a mesma tecnologia para trazer o foco dos alunos novamente ao conteúdo corrente.

Como cada vez mais os computadores estão presentes em nosso meio, certamente, poderíamos usar estes para desenvolver a lógica de programação, ensinar e aprender conceitos matemáticos; familiarizando os estudantes com o ambiente virtual, uma vez que podemos ter a máquina como aliada. De acordo com pesquisa do DataFolha, 2021, apresentada em janeiro, sobre a Educação brasileira, aponta que aproximadamente 4 milhões de estudantes brasileiros entre 6 e 34 anos deixaram as aulas em 2020, o que significa 8,4% de evasão escolar. Na educação básica, a taxa é ainda maior: 10,8% dos alunos largaram a escola em 2020, sendo 4,6% no ensino fundamental. Para termos de comparação, em 2019 as taxas oficiais de evasão foram de 4,8% no ensino médio e 1,2% no fundamental. Os números dão a dimensão do desafio em um cenário no qual o retorno progressivo às aulas presenciais permanece indefinido².

2 ‘Pandemia’ de abandono e evasão escolar, disponível em: <<https://projetc colabora.com.br/ods4/pandemia-de-abandono-e-evasao-escolar/>> Acesso em 31/07/2021.

Apesar do acesso remoto não ser viável para todos os estudantes, ele se tornou indispensável em tempo de pandemia, e muitas vezes a única alternativa para os estudantes não perderem o fio condutor do estudo. Nesse ponto encontramos um outro grande problema: a inclusão digital. Muitos estudantes são autodidatas, ou tem a orientação de um adulto, porém nem todos conseguem acompanhar os estudos remotos. Como contraponto temos um aumento gigantesco de crianças e adolescentes com acesso a Celulares, e a rede *web*.

Segundo (PESENTE, 2019)³ “As Tecnologia de Informação e Comunicação (TICs) são meios que dispomos para chegar nos alunos. E nesse momento de crise da educação, onde não sabemos quando retornarão as aulas presenciais, a inclusão digital e as ferramentas digitais podem, de fato, atrair novamente a atenção dos estudantes.” Porque a informática está presente em nosso dia a dia, e o mercado de trabalho carece de profissionais qualificados, que possuam o *background* necessário para desempenhar as funções dos cargos. Em Ciência de dados, temos a estatística como requisito base. Já na área de desenvolvimento, temos necessidade de modelagem matemática, álgebra, lógica, etc.

Assim, percebemos também o aumento da demanda de softwares voltados para a educação. “A linguagem fornece os conceitos e as formas de organização do real que constituem a mediação entre o sujeito e o objeto de conhecimento” (FOSSILE, 2010). Assim, surge a questão: Como podemos utilizar a linguagem científica *Python* para a modelagem matemática, trazendo a inclusão digital de maneira saudável?

3 PESENTE, Guilherme Moraes, O ENSINO DE MATEMÁTICA POR MEIO DA LINGUAGEM DE PROGRAMAÇÃO PYTHON.

1.3 JUSTIFICATIVAS

As TICs, que são na verdade todas as Tecnologias da Informação e Comunicação que impactam a produtividade corporativa, hoje em dia possuem grande importância para ajudar professores, alunos, instituições e os demais que fazem seu uso no processo do desenvolvimento cognitivo do aluno. A preferência pelo uso de TICs dar-se pela constante interação deste público com estes objetos digitais, o que pode proporcionar a interação entre aluno e tecnologias digitais, tornando as aulas dinâmicas e atrativas para os alunos, beneficiando-os durante o processo de desenvolvimento acadêmico e cognitivo.

Os efeitos de uma aula maçante de teoria matemática, poderia ser suavizada pela interação com o computador de modo dinâmico e interativo. Segundo LÓPEZ (2011) se faz necessário que o educador, como agente responsável do processo educativo, esteja, constantemente, sendo desafiado a equilibrar a necessidade de formação dos educandos com a exigência da ampliação e transformação do próprio conhecimento. Almejamos que o uso dos recursos da informática como: internet, softwares educacionais, softwares de propósitos gerais, ajudem a criar situações favoráveis à aprendizagem dos conceitos e à superação das dificuldades dos alunos, de forma a construir uma base sólida de conhecimentos nas diversas áreas.”

As experiências propostas pelo poder público, no sentido de capacitar o professor a trabalhar com as tecnologias digitais nos ambientes escolares, são, via de regra, desarticuladas da formação inicial de professores. Nos currículos das licenciaturas, as tecnologias não têm ainda um lugar de destaque. A ênfase no processo de capacitação recai sobre a formação continuada. (MAIA, BARRETO. 2021)

Esse trabalho tem por propósito apresentar as bases científicas e a eficiência da linguagem de programação *Python* por meio do ensino da modelagem matemática, favorecendo o aprendizado de conceitos da matemática, da lógica e da ciência de computação. De acordo com Biembengut (2000) “a ideia de modelagem suscita a imagem de um escultor trabalhando com argila, produzindo um objeto. Esse objeto é um modelo. O escultor munido de material – argila, técnica, intuição e criatividade – faz seu modelo, que na certa representa alguma coisa, seja real ou imaginária.”

Ainda considerando a Modelagem matemática, Bassanezi (2002) caracteriza dois tipos: “o modelo objeto e o modelo teórico, sendo que para cada tipo vamos ter as diferentes representações. No modelo objeto, vamos representar um objeto ou um fato concreto por meio de suas características predominantes, utilizando uma representação pictórica, conceitual ou simbólica. No modelo teórico, vamos utilizar como ponto de partida o modelo objeto para vincular a uma teoria geral existente, a fim de estabelecer o modelo com as variáveis essenciais do fenômeno, sendo que as relações são obtidas por meio de hipóteses ou experimentos⁴”.

No presente documento, vamos nos deter ao modelo teórico. E teremos por objeto o ensino da matemática em ambiente virtual. Pois segundo LÓPEZ (2011) o computador será considerado um recurso facilitador da construção do conhecimento no momento em que você utilizá-lo como auxiliar da sua prática pedagógica... será a soma do esforço e competência pessoal com a capacidade de empregar apropriadamente um *software*, que tornará o computador uma ferramenta útil para o planejamento e execução de atividades pedagógicas.”

1.4 OBJETIVOS

Para a realização desta pesquisa, definiu-se como finalidade, testar a capacidade da linguagem Python para formulação e resolução de problemas matemáticos, onde traçou-se um objetivo geral que foi desdobrado em outros sete objetivos específicos que são apresentados a seguir.

1.4.1 Objetivo Geral

Investigar a eficiência da linguagem de programação *Python*, como recurso didático para formular e resolver problemas matemáticos no contexto da educação básica.

4 Bassanezi (2002). Apud FLEMMING, Diva Marília. 2019

1.4.2 Objetivos Específicos

- ❖ Trazer dinamismo para o ensino da matemática e desenvolver habilidade de lógica, cálculo e abstração matemática;
- ❖ Desenvolver habilidade de construção de modelagem matemática e integrar nos conceitos de matemática com a linguagem de programação;
- ❖ Estabelecer a importância do uso de tecnologias no ensino e na aprendizagem da matemática;
- ❖ Introduzir à linguagem científica Python para desenvolvimento matemático;
- ❖ Aprender a acessar a Plataforma de inserção dos códigos *Jupyter Notebook*⁵;
- ❖ Propor situações didáticas com uso da linguagem Python para ensinar conceitos matemáticos da educação básica.

1.5 Tipo de pesquisa

O plano inicial da pesquisa constitui na exploração do ambiente em que foram levantados e definidos o objeto de pesquisa. Desta forma, esse estudo pode ser caracterizado como uma pesquisa exploratória, qualitativa e bibliográfica.

- ❖ Quanto à abordagem, temos um enfoque qualitativo. Porque nela é testada a eficiência da linguagem científica para construção e resolução de problemas. Essa eficiência se refere a facilidade de expressão, ou modelagem do problema, e a velocidade e precisão da resposta. O ambiente natural para coleta de dados é o ambiente virtual (IDE). Segundo Godoy (1995, pag. 21) algumas características básicas identificam os estudos denominados “qualitativos”. Segundo esta perspectiva, um fenômeno pode ser melhor compreendido no contexto em que ocorre e do qual é parte, devendo ser analisado numa perspectiva integrada. Para tanto, o pesquisador vai a campo buscando “captar” o fenômeno em estudo a partir da perspectiva das pessoas nele envolvidas,

5 Google Colaboratory. <<https://colab.research.google.com/notebooks/>> Acesso em 31/07/2021

considerando todos os pontos de vista relevantes. Vários tipos de dados são coletados e analisados para que se entenda a dinâmica do fenômeno.

- ❖ Quanto aos objetivos é uma pesquisa exploratória. A pesquisa exploratória é um procedimento metodológico muito usado para fazer trabalhos acadêmicos. Com ela, o estudante adquire “maior familiaridade com o problema e assim consegue construir as hipóteses” GIL (2002, p. 41). A pesquisa exploratória busca amparo, sobretudo, na abordagem qualitativa. Que visa explorar novos recursos da área, e sua aplicação em sala de aula. O desenvolvimento de simulações que complementem a análise experimental e estejam diretamente interligadas ao conteúdo teórico, leva o aluno a assumir um papel mais ativo, pois, além da realização do experimento, ele é induzido a buscar o embasamento teórico para a compreensão dos resultados da simulação e correta comparação deste com os dados experimentais, tornando a aprendizagem mais dinâmica. Temos assim, uma pesquisa onde foi necessário explorar as operações matemáticas, e suas possíveis correlações em algoritmo e código.

- ❖ É considerada uma pesquisa bibliográfica, pois segundo Rovigati (2009, p. 1) “A pesquisa bibliográfica é o passo inicial na construção efetiva do processo de investigação, quer dizer, após a escolha de um assunto é necessário fazer uma revisão bibliográfica do tema apontado. Essa pesquisa auxilia na escolha em um método mais apropriado, assim como em um conhecimento das variáveis e na autenticidade da pesquisa.” A linguagem Python tem vasto material impresso e digital, porém é uma linguagem técnica e científica, e essa revisão visa facilitar alguns conceitos fundamentais, possibilitando sua fácil utilização pelo professor ou pelo aluno em sala.

1.6 Estrutura do trabalho

O presente estudo, inicialmente, é composto pela introdução, na qual constam o tema e delimitação do tema, a problematização, os objetivos: geral e específicos, as justificativas e o tipo de pesquisa realizada.

No segundo capítulo, que contém a fundamentação teórica, faz-se uma abordagem acerca dos conceitos envolvendo a modelagem matemática tais como: definição de modelagem matemática, de modelos matemáticos, as etapas do processo de modelagem matemática, e a modelagem como metodologia de ensino.

O terceiro capítulo, que também faz parte da fundamentação teórica, trazem-se todas as informações a respeito da linguagem científica Python. No primeiro momento, fez-se um estudo geral sobre a aprendizagem significativa, parâmetros para a organização do ensino de Matemática no Ensino Médio, e a lógica de construção de um algoritmo. Em seguida partimos para o estudo dos Softwares, em suas diferentes plataformas que poderiam ser utilizados na pesquisa.

O quarto capítulo contém propostas de sequências didáticas de conceitos matemáticos da educação básica com a utilização dos códigos Python.

No quinto capítulo constam as conclusões e considerações finais obtidas com a construção desta monografia. E, por fim, as referências bibliográficas.

2 MODELAGEM MATEMÁTICA NA EDUCAÇÃO

A ideia de modelo matemático, para Burak (2010), refere-se a uma representação em linguagem matemática, geralmente sob a forma de uma equação, inequação, sistema de equações, a planta baixa de uma casa ou um mapa, uma tabela. Até mesmo “[...] uma lista de preços em uma tabela, por exemplo, como capaz de ajudar na tomada de decisões” (BURAK; KLÜBER, 2013, p.42) pode ser considerada um modelo matemático. Segundo o autor, os modelos matemáticos têm como objetivo explicar matematicamente situações do cotidiano das pessoas, ajudando-as a fazer previsões e tomar decisões.

A ideia desenvolvida é a Matemática como um sistema de códigos e regras que a tornam uma linguagem de comunicação de ideias e permite modelar a realidade e interpretá-la. Considerando as operações matemáticas como nosso objeto de pesquisa, e o *software* como ferramenta de ensino, ambos foram utilizados como ferramentas criativas associadas com a aula teórica, onde são transmitidas as modelagens a serem criadas.

2.1 Aprendizagem matemática no Ensino Médio

Os conhecimentos obtidos em conteúdos como os encontrados na disciplina de matemática acompanha a trajetória de alunos desde o início de sua jornada acadêmica. Nos anos iniciais, as crianças aprendem operações básicas como soma e subtração, e com o tempo, o conhecimento se desenvolve, novos desafios vão sendo propostos e ocorrendo o avanço destes conhecimentos existentes.

Para algumas pessoas, estes conhecimentos são mais difíceis do que para os demais, isto acaba acarretando no desestímulo destas crianças, assim os educadores precisam usar outras metodologias, visando a melhor compreensão do conteúdo.

A chave para ter sucesso na sala de aula de matemática é aprender a pensar dentro da caixa. No entanto, os matemáticos do mundo real pensam fora da caixa para que possam resolver os problemas do mundo real à medida que surgem. As crianças de hoje precisam ser pensadoras, criativas e inovadoras, como os matemáticos, para que possam resolver problemas do mundo real e se tornar os inovadores do amanhã.

Ao se estabelecer um primeiro conjunto de parâmetros para a organização do ensino de Matemática no Ensino Médio, pretende-se contemplar a necessidade da sua adequação para o desenvolvimento e promoção de alunos, com diferentes motivações, interesses e capacidades, criando condições para a sua inserção num mundo em mudança e contribuindo para desenvolver as capacidades que deles serão exigidas em sua vida social e profissional. Em um mundo onde as necessidades sociais, culturais e profissionais ganham novos contornos, todas as áreas requerem alguma competência em Matemática e a possibilidade de compreender conceitos e procedimentos matemáticos é necessária tanto para tirar conclusões e fazer argumentações, quanto para o cidadão agir como consumidor prudente ou tomar decisões em sua vida pessoal e profissional. (PCN, pag. 40)

A Matemática no Ensino Médio tem um valor formativo, que ajuda a estruturar o pensamento e o raciocínio dedutivo, porém também desempenha um papel instrumental, pois é uma ferramenta que serve para a vida cotidiana e para muitas tarefas específicas em quase todas as atividades humanas.

Em seu papel formativo, a Matemática contribui para o desenvolvimento de processos de pensamento e a aquisição de atitudes, cuja utilidade e alcance transcendem o âmbito da própria Matemática, podendo formar no aluno a capacidade de resolver problemas genuínos, gerando hábitos de investigação, proporcionando confiança e desprendimento para analisar e enfrentar situações novas, propiciando a formação de uma visão ampla e científica da realidade, a percepção da beleza e da harmonia, o desenvolvimento da criatividade e de outras capacidades pessoais.

No que diz respeito ao caráter instrumental da Matemática no Ensino Médio, ela deve ser vista pelo aluno como um conjunto de técnicas e estratégias para serem aplicadas a outras áreas do conhecimento, assim como para a atividade profissional. Não se trata de os alunos possuírem muitas e sofisticadas estratégias, mas sim de desenvolverem a iniciativa e a segurança para adaptá-las a diferentes contextos, usando-as adequadamente no momento oportuno.

Nesse sentido, é preciso que o aluno perceba a Matemática como um sistema de códigos e regras que a tornam uma linguagem de comunicação de ideias e permite modelar a realidade e interpretá-la. Assim, os números e a álgebra como sistemas de códigos, a geometria na leitura e interpretação do espaço, a estatística e a probabilidade na compreensão

de fenômenos em universos finitos são subáreas da Matemática especialmente ligadas às aplicações.

A essas concepções da Matemática no Ensino Médio se junta a idéia de que, no Ensino Fundamental, os alunos devem ter se aproximado de vários campos do conhecimento matemático e agora estão em condições de utilizá-los e ampliá-los e desenvolver de modo mais amplo capacidades tão importantes quanto as de abstração, raciocínio em todas as suas vertentes, resolução de problemas de qualquer tipo, investigação, análise e compreensão de fatos matemáticos e de interpretação da própria realidade. (PCN, pag.41)

2.2 Raciocínio lógico e o computador

Um dos desafios na aprendizagem matemática está relacionado ao aluno estar disposto a aprender o conceito da matemática, porém, existe o desafio relacionado ao preparo que os professores recebem em sua formação (BEZERRA, 2016).

Conforme Bezerra (2016) aponta em sua pesquisa, muitos professores dos anos iniciais durante sua capacitação profissional, neste caso, o curso de pedagogia, não os prepara para o ensino de diversas disciplinas e sim para alfabetização da criança. Por meio desta constatação os professores “julgam que o que aprenderam na faculdade não os prepara de modo suficiente para lecionar nas séries do Ensino Fundamental I” (BEZERRA, 2016, p.2)

A matemática não deve ser relacionada como um desafio inviável de se resolver, mas sim, como uma disciplina que tem papel importante na formação intelectual, pensamento crítico, raciocínio lógico, situações do cotidiano e desenvolvimento cognitivo para compreensão de outras áreas curriculares (PCN, 1998, p.28).

Tal disciplina não pode ser conotada como uma barreira pelos alunos, mas, uma base que não é fortificada nas séries iniciais, torna-se um elemento complicador quando passa do Ensino Fundamental para o médio, uma vez que encontram conteúdos que para sua melhor compreensão, dependem de uma base cognitiva sólida que deveria ser fornecida no Ensino Fundamental.

2.3 Algoritmo e modelagem matemática

Um algoritmo de programação é criado a partir de um procedimento de modelagem matemática. Assim, convém que definamos com clareza, o que entendemos como Modelagem Matemática. Segundo (BIEMBENGUT, 2000) temos as etapas:

a) Interação

- reconhecimento da situação-problema;
- familiarização com o assunto a ser modelado → referencial teórico.

b) Matematização

- formulação do problema → hipótese;
- resolução do problema em termos do modelo.

c) Modelo matemático

- Interpretação da solução;
- validação do modelo → avaliação.

De uma forma mais clara, podemos considerar a Modelagem no contexto educacional, baseado em Burak (1992, 2004), em cinco etapas: 1) a escolha do tema; 2) a pesquisa exploratória; 3) o levantamento dos problemas; 4) a resolução dos problemas e desenvolvimento do conteúdo matemático no contexto do tema; 5) a análise crítica da solução.

Temos que considerar para esse estudo a diferença e separação dos *softwares* em dois grupos: *Softwares educacionais* e *Softwares gerais*. Entende-se como *software* educacional todo aquele que possa ser usado para algum objetivo educacional, pedagogicamente defensável, por professores e alunos, qualquer que seja a natureza para a qual tenha sido criado. (LUCENA, 1992 apud LOPEZ, 2011)

Desta forma podemos iniciar a **Interação** com uma simples operação de soma de frutas. Apresentamos uma quantidade x para cada fruta, definindo assim os valores das variáveis de entrada. Em seguida passamos para **Matematização**, onde definimos a operação

de soma e ordenamos os valores. E por fim chegamos ao **Modelo Matemático** onde enxergamos o resultado da aplicação dos valores (soma) dentro do algoritmo correspondente, validando o modelo. Evidentemente, os modelos devem seguir uma escala gradual de complexidade correspondente ao nível dos alunos. Dessa forma, a teoria sai do ‘quadro branco’ do professor e cria vida na tela do computador.

Os aplicativos computacionais de propósito gerais são *softwares* amplamente utilizados nas mais diversas atividades administrativas e de produção, que pelas suas características generalistas, podem ser empregados com sucesso em atividades educacionais.

No presente trabalho utilizaremos *software* de propósito geral para atingir os fins propostos no objetivo. Mas antes de utilizarmos aquele, criaremos um **Pseudocódigo**, que é uma forma genérica de escrever um algoritmo, utilizando uma linguagem simples (nativa a quem o escreve, de forma a ser entendida por qualquer pessoa) sem necessidade de conhecer a sintaxe de nenhuma linguagem de programação.

Um algoritmo não representa, necessariamente, um programa de computador, e sim os passos necessários para realizar uma tarefa. Sua implementação pode ser feita por um computador, por outro tipo de autômato ou mesmo por um ser humano. Diferentes algoritmos podem realizar a mesma tarefa usando um conjunto diferenciado de instruções em mais ou menos tempo, espaço ou esforço do que outros. Tal diferença pode ser reflexo da complexidade computacional aplicada, que depende de estruturas de dados adequadas ao algoritmo. Por exemplo, um algoritmo para se vestir pode especificar que você vista primeiro as meias e os sapatos antes de vestir a calça enquanto outro algoritmo especifica que você deve primeiro vestir a calça e depois as meias e os sapatos. Fica claro que o primeiro algoritmo é mais difícil de executar que o segundo apesar de ambos levarem ao mesmo resultado.

Assim, a escrita de programas em pseudocódigo tem dois objetivos: primeiro eliminar totalmente a sintaxe e os detalhes técnicos da linguagem. Segundo se concentrar na lógica e na ordem das operações dos algoritmos. Ainda de acordo com PCN, “acompanhar criticamente o desenvolvimento tecnológico contemporâneo, tomando contato com os avanços das novas tecnologias nas diferentes áreas do conhecimento para se posicionar frente às questões de nossa atualidade. Utilizar o conhecimento matemático como apoio para compreender e julgar as aplicações tecnológicas dos diferentes campos científicos” são elementos de contextualização sociocultural.

3 A LINGUAGEM CIENTÍFICA PYTHON

Entre as muitas utilidades da linguagem Python, destacamos a aplicações dos conceitos de lógica, algoritmos e linguagem de programação aplicados à resolução de problemas de engenharia e economia de tempo com cálculos extensos e trabalhos largamente utilizada nos cursos de graduação superior. O desenvolvimento de simulações que complementem a análise experimental e estejam diretamente interligadas ao conteúdo teórico, leva o aluno a assumir um papel mais ativo, pois, além da realização do experimento, ele é induzido a buscar o embasamento teórico para a compreensão dos resultados da simulação e correta comparação deste com os dados experimentais, tornando a aprendizagem mais dinâmica. Além disso, as simulações podem estimular a utilização de linguagens de programação em disciplinas de física, facilitar o aprendizado do aluno em assuntos conceitualmente mais desafiadores e ajudar na solução de equações com cálculos analíticos mais complexos (DORNELES; ARAUJO; VEIT, 2006).

Os laboratórios de informática são novos espaços de aula que podem favorecer o processo de construção e aquisição de conhecimento dos alunos e propiciar aos professores condições para o trabalho adequado no novo ambiente, independente da faixa etária escolar (MAIA; BARRETO, 2012; OLIVEIRA; FERREIRA; CRUZ, 2019) e cada vez mais os *Softwares* se tornam aliados do docente, e ferramenta educacional que aproxima o educando da ciência em constante evolução.

3.1 O que é uma linguagem científica de programação?

Com o passar do tempo e com a evolução tecnológica, as metodologias de ensino-aprendizagem passam por mudanças de forma a acompanhar essa evolução, uma vez que o mercado e a sociedade exigem cada vez mais profissionais capacitados e competentes que tenham acompanhado tal evolução. No contexto da engenharia, a informática está cada vez mais presente, desde os programas de desenho auxiliado por computador até aplicativos do

tipo “calculadora”, que desempenham a função de realizar cálculos trabalhosos e complexos apenas com os dados inseridos pelo usuário. Essa é uma área já consolidada da pesquisa em educação, porém em constante evolução e que exige que a formação de professores e alunos contemplem a utilização de programas e o computador como objeto para estudo (MAIA; BARRETO, 2012; OLIVEIRA; FERREIRA; CRUZ, 2019).

Uma linguagem nova pode ser aprendida através de uma aprendizagem significativa. Que é aquela em que ideias expressas simbolicamente interagem de maneira substantiva e não arbitral com aquilo que o aprendiz já sabe. Substantiva quer dizer não-literal, não ao pé-da-letra, e não-arbitrária significa que a interação não é com qualquer ideia prévia, mas sim com algum conhecimento especificamente relevante já existente na estrutura cognitiva do sujeito que aprende.

A este conhecimento, especificamente relevante à nova aprendizagem, o qual pode ser, por exemplo, um símbolo já significativo, um conceito, uma proposição, um modelo mental, uma imagem, David Ausubel (1918-2008) chamava de **subsunçor** ou ideia-âncora. Em termos simples, subsunçor é o nome que se dá a um conhecimento específico, existente na estrutura de conhecimentos do indivíduo, que permite dar significado a um novo conhecimento que lhe é apresentado ou por ele descoberto. Tanto por recepção como por descobrimento, a atribuição de significados a novos conhecimentos depende da existência de conhecimentos prévios especificamente relevantes e da interação com eles.

O subsunçor pode ter maior ou menor estabilidade cognitiva, pode estar mais ou menos diferenciado, ou seja, mais ou menos elaborado em termos de significados. Contudo, como o processo é interativo, quando serve de ideia-âncora para um novo conhecimento ele próprio se modifica adquirindo novos significados, corroborando significados já existentes. É importante reiterar que a aprendizagem significativa se caracteriza pela interação entre conhecimentos prévios e conhecimentos novos, e que essa interação é não-literal e não-arbitrária. Nesse processo, os novos conhecimentos adquirem significado para o sujeito e os conhecimentos prévios adquirem novos significados ou maior estabilidade cognitiva.

Porém, um subsunçor ao não ser utilizado pode se “desgastar” com o passar dos tempos. No conceito de cognição isso é comum, pois o primeiro subsunçor já não é mais utilizado, porém, na aprendizagem significativa, todo conhecimento pode ser reaprendido de uma forma mais rápida e eficiente (MOREIRA, 2012, p.4).

Na programação de computadores, alguns comandos que não são utilizados com certa frequência podem ser esquecidos com o passar dos tempos, porém, se o programador precisar vir a utilizá-lo novamente, seu processo de reaprendizagem será mais rápido em relação à primeira vez.

Desta forma, a aprendizagem significativa é dita como uma aprendizagem na qual é possível que o indivíduo possa esquecer algum conhecimento. Este esquecimento é residual, ou seja, ele se encontra dentro de um subsunçor que pode ser acessado a qualquer momento.

Assim como temos diferentes línguas para falarmos, existem diversas linguagens que nos permitem “falar” com os computadores. Entre as línguas de produção, o **Python** é uma das mais fáceis de aprender e uma das que mais cresce no mundo em termos de utilização.

Pode ser utilizado em diversas áreas:

- *Data Science* (Ciência de dados);
- Automação de processos;
- Desenvolvimento de sites;
- Inteligência artificial;
- Vários outros

O seu nome apesar de geralmente ser vinculado a cobra, não tem essa origem.... Na verdade, ele é uma homenagem a um grupo de comédia inglês chamado Monty Python. É uma linguagem de programação de alto nível, interpretada de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum⁶ em 1991. Atualmente, possui um modelo de desenvolvimento comunitário, aberto e gerenciado pela organização sem fins lucrativos Python Software Foundation⁷. Apesar de várias partes da linguagem possuírem padrões e especificações formais, a linguagem, como um todo, não é formalmente especificada.

As políticas públicas para a implantação das tecnologias digitais na escola ultrapassaram a exclusiva aquisição de hardware e preocuparam-se também com os problemas

⁶ Guido van Rossum. Matemático e programador de computadores holandês, mais conhecido por ser o autor da linguagem de programação Python. Na comunidade Python, ele é conhecido como Benevolent Dictator for Life (BDFL), o que significa que ele continua a supervisionar o processo de desenvolvimento do Python, tomando decisões quando necessário.

⁷ <https://www.python.org/psf/>

de software. O Governo Federal, em 2003, determinou a adoção preferencial de software livres como solução informática em suas repartições (Souza, 2008). Essa experiência que visava, primeiramente, a economia com despesas para pagamento de royalties e taxas de licenças de uso de programas de computador, serviu de exemplo para que outras instâncias da administração pública fizessem o mesmo. Dentre os motivos que fizeram governos e empresas adotarem softwares livres como soluções em informática estão: i) o controle dos custos iniciais e de upgrades das licenças de software; ii) a redução da dependência de empresas desenvolvedoras de software proprietário; e iii) a promoção do uso de programas de computador no setor público. (Dravis, 2003)

3.2 *Software*, diferenças básicas e plataforma de ensino

A aplicação de softwares simuladores como ferramentas didáticas de apoio à educação tecnológica não é recente, e tem demonstrado ser uma das principais soluções para a viabilização da oferta de cursos que necessitem de demonstrações e ensaios práticos. Existem muitas soluções de *software*, comercialmente disponíveis, para atender a essa demanda. Também existem diversas alternativas de software gratuito (sejam livres ou versões de demonstração funcionais) para contornar essa dificuldade. (RAMALHO; MOREIRA, 2021)

Os códigos em Python precisam de uma plataforma para serem escritos. Essas plataformas na programação são chamadas de IDEs⁸. Nesse documento utilizaremos o *Jupyter Notebook* que é uma ferramenta gratuita que existe dentro do *Anaconda* (uma espécie de grande caixa de ferramentas do Python). O *Jupyter Notebook*, também pode ser acessado pelo *Software Sagemath*⁹. Que é uma opção mais ‘leve’ comparada ao Pacote Anaconda. E apresenta os mesmos recursos para compilação do código. Para usarmos o *Jupyter Notebook*, iremos instalar o *Anaconda*¹⁰ *software* na realização das construções de código. Existem várias outras IDEs disponíveis: *Spyder*¹¹, *PyCharm*, *Atom*, *Google Colab*, etc.

8 IDE, ou ambiente de desenvolvimento integrado, é um software que combina ferramentas comuns de desenvolvimento em uma única interface gráfica do usuário (GUI), facilitando o desenvolvimento de aplicações.

9 Site: <<https://www.sagemath.org/>> Acesso em 01/08/2021

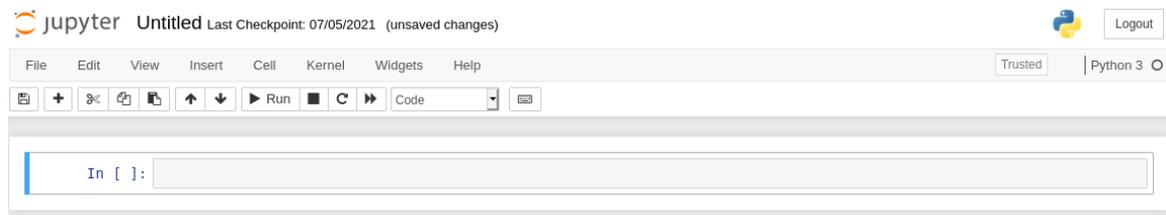
10 <https://www.anaconda.com/products/individual/download-success>. Acesso em 31/08/2021.

11 Disponível em: <<https://www.spyder-ide.org/>> Acesso em 5/09/2021.

Em um primeiro momento, o *software* pode se equiparar a uma calculadora, porém os recursos dados pelas bibliotecas são vastos, e acompanham a necessidade da atualidade. Assim, a introdução formal de um algoritmo, ou modelagem matemática utilizando *Python* abrange muito mais do que uma simples calculadora pode fazer.

Após baixar o *software* e realizar a instalação apropriada, executamos o aplicativo que irá ser aberto no navegador padrão instalado. Na página inicial que foi aberta, ache o botão *New* (*Novo*) no canto superior direito, e em seguida clique em *Python3*. Assim, teremos o compilador *Python* disponível para a entrada das informações. A apresentação ou *interface gráfica* básica onde os códigos são digitados é como abaixo.

Figura 1 - Saída padrão do Jupyter Notebook



Fonte: produzido pelo Autor, 2021.

Figura 2 - Entrada de dados Jupyter Notebook.

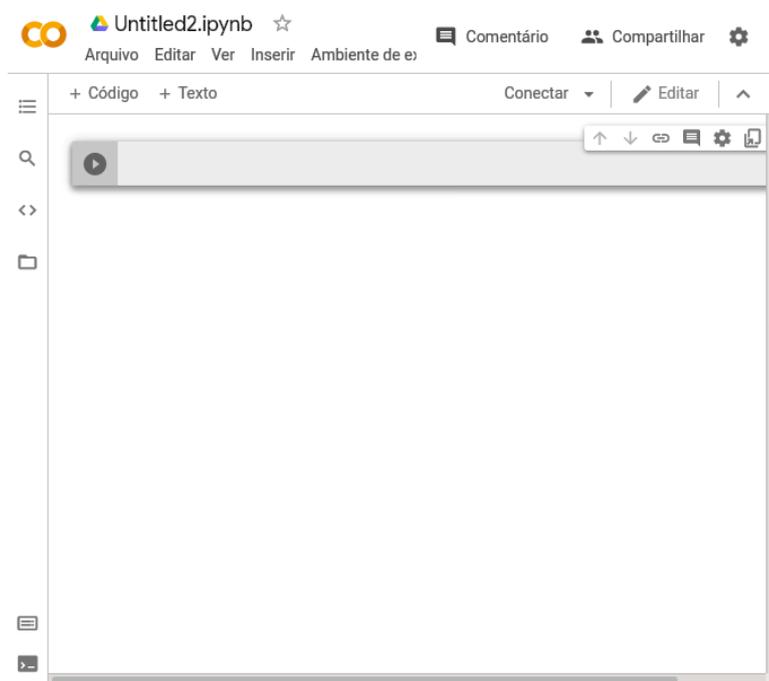


Fonte: produzido pelo Autor, 2021.

Na caixa *In[]*: é onde o aluno irá digitar os códigos e em seguida teclar: **Ctrl+Enter** para executar o código, que irá criar uma saída *Out[1]* abaixo com a resposta no console. A partir da primeira entrada, segue uma numeração das etapas, onde podemos ver claramente a ideia de algoritmo sendo aplicada. A sequência apropriada garante a execução do código de acordo com o esperado.

Para o ensino remoto temos a alternativa do *Google Colaboratory*¹². Falamos, uma alternativa, porque o ensino remoto no Brasil neste momento de Pandemia (atualmente 09/2021) é uma realidade. Nem sempre o aluno pode ir à escola; ou talvez instalar o pacote *Anaconda* não seja do interesse da escola ou do aluno. Neste caso, temos uma opção remota que funciona na ‘nuvem’ que não utiliza a memória do computador. A seguir apresentamos a interface do *Google Colab*.

Figura 3 - *Google Colaboratory - Colab*



Fonte: <https://colab.research.google.com>. Acesso, 11/9/2021.

3.3 Importando Bibliotecas

O primeiro passo para iniciarmos a Interação da nossa aplicação é entender, o que na ciência da computação, é uma Biblioteca: uma coleção de subprogramas utilizados no desenvolvimento de *software*. Bibliotecas contêm código e dados auxiliares, que provém serviços a programas independentes, o que permite o compartilhamento e a alteração de código e dados de forma modular. Alguns executáveis são tanto programas independentes

12 Disponível em: <<https://colab.research.google.com/>> Acesso em 5/09/2021.

quanto bibliotecas, mas a maioria das bibliotecas não são executáveis. Executáveis e bibliotecas fazem referências mútuas conhecidas como *ligações*, tarefa tipicamente realizada por um ligador.

A maior parte dos sistemas operacionais modernos provê bibliotecas que implementam a maioria dos serviços do sistema, que transformaram em comodidades os serviços que uma aplicação moderna espera que sejam providos pelo sistema operacional. Assim sendo, a maior parte do código utilizado em aplicações modernas é fornecido por estas bibliotecas. A Biblioteca Padrão do *Python* é uma coleção de módulos de *script* acessíveis a um programa *Python* para simplificar o processo de programação e remover a necessidade de reescrever os comandos mais usados.

As principais bibliotecas que utilizaremos para este estudo, adaptado de Marcondes (2018, pag. 19) são:

- *numpy* – computação científica;
- *math* – funções trigonométricas;
- *fractions* – manipulação de frações;
- *sympy* – matemática simbólica;
- *pylab* – gráficos bidimensionais;
- *matplotlib* – figuras e gráficos;

O comando utilizado para ‘carregar’ a biblioteca e tornar disponível suas funcionalidades é o comando ***import***. Se, por exemplo, utilizarmos o comando:

```
>>> import numpy
```

Neste caso, todas as funções da biblioteca *numpy* são importadas. Isso pode ser inconveniente caso o programa incorpore mais de uma biblioteca. Por essa razão, que é basicamente, não gerar conflito entre bibliotecas diferentes que possuem funções com mesmo nome, utilizamos uma função de cada vez, como por exemplo:

```
>>> from math import sin
```

Nesse caso, somente a **função *sin*** da **biblioteca *math*** é importada. E, de uma maneira geral, é importada apenas a função que será utilizada na execução do código.

4. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

Neste capítulo são apresentadas cinco propostas de atividades didáticas que poderão servir de aporte para uma futura aplicação em sala, com estudantes da educação básica. Todas as atividades foram elaboradas com conteúdos específicos da matemática da educação básica.

Para a plotagem dos gráficos utilizou-se o *Software Spyder*, por ser de fácil manuseio e permitir fazer cálculos matemáticos, bem como planilhas, gráficos e outras funções. Para uma futura aplicação nada impede que outros softwares sejam utilizados.

A criação dos códigos seguintes foi realizada pelo autor, através do *Software Spyder*, porém podem ser realizadas em qualquer compilador Python¹³. O texto em verde precedido pelo ‘#’ são comentários que não tem importância real na execução, no entanto auxiliam na interpretação do código em cada uma de suas partes. Para a Criação do algoritmo em Python, seguimos algumas recomendações iniciais:

1. Tenha o cálculo e algoritmo formal escrito no papel;
2. Abra a IDE de compilação de código Python¹⁴;
3. Na janela de inserção de dados escreva o seu código;
4. Depois de completo, clique em executar (F5) ou (Ctrl+Return);
5. Verifique se o resultado está coerente, se não estiver verifique os erros.

Em todas as atividades didáticas propostas, a seguir, desenvolve-se cada uma de acordo com as instruções explanadas anteriormente.

¹³ Para utilização do *Jupyter Notebook* é necessária a instalação do pacote *Anaconda*. Disponível em <https://www.anaconda.com/products/individual> Acesso em 11/09/2021.

¹⁴ Aqui caber ressaltar que o compilador Python, em uma das interfaces citadas, esteja instalado e rodando no computador.

4.1 SEQUÊNCIA DIDÁTICA: Operações Básicas

Nesta atividade vamos desenvolver conceitos das operações matemáticas de adição, subtração, multiplicação, divisão e exponenciação. Assim esta atividade inicial visa ensinar aluno as operações básicas aplicadas em código Python. A declaração de valores para variáveis, e a utilização das operações básicas, permitem um primeiro contato no estudante com o objeto de estudo.

1 - Objetivos específicos:

- ❖ Realizar operações básicas da matemática de adição, subtração, multiplicação e divisão;
- ❖ Aprender como definir valores para variáveis, e perceber a equivalência de valores numéricos resultante das operações matemáticas.

2 – Metodologia:

A atividade será desenvolvida a partir das etapas seguintes:

Etapas 1: Escrever o pseudocódigo no papel, junto com algoritmo em Python;

Etapas 2: Utilizar o compilador Python para a realização da proposta;

Etapas 3: Validação de acerto e verificação de erros.

3 – Sequência Didática:

O Python disponibiliza à partida, para além de algumas funcionalidades básicas, várias operações numéricas que podemos utilizar: + (adição), - (subtração), * (multiplicação), / (divisão), ** (exponenciação), // (divisão inteira), % (resto da divisão inteira). Agrupamos expressões usando parênteses, como é usual. As expressões são validadas tecendo **SHIFT+RETURN** ou **F5**.

Definindo os valores de entrada $a = 4$ e $b = 7$. Em seguida as operações. E por último a impressão no console. Temos,

Figura 4 - Código Python - Operações Básicas

```
#Define valores, Somente inteiros
a = 4
b = 7

#Operações básicas
soma = a + b
subtracao = a - b
multiplicacao = a * b
divisao = a / b

print ("Resultados das operações entre a e b")
print("a e b definidos como valores int")
print("Resultados somente inteiros")
print ('a =', a)
print ('b =', b)
print ('a + b =', soma)
print('a - b =', subtracao)
print ('a * b =', multiplicacao)
print('a / b =', divisao)
```

Fonte: produzido pelo Autor, 2021.

De acordo com o código acima, em 'define os valores' das variáveis (a e b), passamos os valores para o cálculo. Em seguida, em 'operações básicas' definimos as variáveis: soma, subtração, multiplicação e divisão, que irão armazenar o valor das operações. Por fim, imprimimos o resultado no console com '*print*', onde podemos ver o resultado de fato.

Figura 5 - Saída do programa

```
Resultados das operações entre a e b
a e b definidos como valores int
Resultados somente inteiros
a = 4
b = 7
a + b = 11
a - b = -3
a * b = 28
a / b = 0.5714285714285714
```

Fonte: produzido pelo Autor, 2021.

4 – Resultados esperados com a atividade:

Com a atividade espera-se que o estudante aprenda a avaliar a razoabilidade de um resultado numérico na construção de argumentos sobre afirmações quantitativas e aprimore as capacidades de interpretar e construir significados para os números naturais, inteiros e racionais.

4.2 SEQUÊNCIAS DIDÁTICA: Potenciação e radiciação

Nesta atividade trataremos de conceitos das operações matemáticas de potenciação e radiciação, que amplamente utilizado em vários campos de pesquisa. Podemos apreciar a correlação do expoente fracionário para representação da raiz quadrada. Agrupamos expressões usando parênteses, para tal representação.

1 - Objetivos específicos:

- ❖ Resolver operações com potenciação e radiciação;
- ❖ Construir significados para os números naturais, utilizando as operações básicas de potenciação e radiciação.

2 – Metodologia de desenvolvimento da atividade:

A atividade será desenvolvida a partir das etapas seguintes:

Etapa 1: Escrever o pseudocódigo no papel, junto com algoritmo em Python

Etapa 2: Utilizar o compilador Python para a realização da proposta;

Etapa 3: Validação de acerto e verificação de erros.

3 – Sequência Didática:

As expressões são validadas tecendo SHIFT+RETURN ou F5.

Definindo os valores de entrada $a = 3$ e $b = 8$. Em seguida as operações são definidas e montadas. Por último temos a impressão do código.

Figura 6 - Código Python - Operações com potências

```
# Limpa a área do console
print ('\n' * 100)

# Define valores
a = 3
b = 8

# Operações com potências
a_quadrado = a**2
a_cubo = a**3
a_quarta = a**4
a_elevado_b = a**b

print ("Resultados das operações de potências")
print ("a =", a)
print ("b =", b)
print ("a ao quadrado =", a_quadrado)
print ('a ao cubo =', a_cubo)
print ('a a quarta =', a_quarta)
print ('a elevado a b =', a_elevado_b)
```

Fonte: produzido pelo Autor, 2021.

De acordo com o código acima, em '# Define os valores' das variáveis (a e b), passamos os valores para o cálculo. Em seguida em '# Operações com potências' definimos as variáveis: a_quadrado, a_cubo, a_quarta e a_elevado_b, como sendo: (x^2 , x^3 , x^4 , x^n) que irão armazenar o valor das operações. Por fim, imprimimos o resultado no console 'print', aonde podemos ver o resultado de fato. Executando, para a saída do programa teremos algo como:

Figura 7 - Saída do programa

```
Resultados das operações de potências
a = 3
b = 8
a ao quadrado = 9
a ao cubo = 27
a a quarta = 81
a elevado a b = 6561
```

Fonte: produzido pelo Autor, 2021.

Para a Radiciação, temos que definir um valor que seja quadrado perfeito. Para variável, neste caso seja = 256.

Figura 8 - Código Python - Operações de Radiciação

```
# Limpa a área do console
print ('\n' * 100)

# Define valores
a = 256

# Operações com potências
a_raiz_quadrada = a**(1.0/2)
a_raiz_cubica = a**(1.0/3)
a_raiz_quarta = a**(1.0/4)

print ('Resultado das operações de radiciação.')
print ('a =', a)
print ('Raiz quadrada de a =', a_raiz_quadrada)
print ('Raiz cubica de a =', a_raiz_cubica)
print ('Raiz quarta de a =', a_raiz_quarta)
```

Fonte: produzido pelo Autor, 2021.

Da mesma forma, definimos as variáveis (sem acentos) : `a_raiz_quadrada`, `a_raiz_cubica`, e `a_raiz_quarta` respectivamente: \sqrt{a} , $\sqrt[3]{a}$, $\sqrt[4]{a}$. Note que a operação é a potenciação com o expoente fracionário. Pois, um número elevado ao expoente $\frac{1}{2}$ é o mesmo que estar sendo colocado dentro da raiz quadrada. A saída esperada do programa é:

Figura 9 - Saída do programa

```
Resultado das operações de radiciação.
a = 256
Raiz quadrada de a = 16.0
Raiz cubica de a = 6.3496042078727974
Raiz quarta de a = 4.0
```

Fonte: produzido pelo Autor, 2021.

4 – Resultados esperados com a atividade:

Com a atividade espera-se que o estudante aprenda a avaliar a razoabilidade de um resultado numérico na construção de argumentos sobre afirmações quantitativas e se aprimore nas capacidades de interpretar e construir significados para os números naturais, inteiros e racionais, e das operações de potenciação e radiciação.

4.3 SEQUÊNCIA DIDÁTICA: Operações com funções de 1º e 2º grau

Nesta atividade utilizamos a linguagem Python para encontrar as raízes de duas funções, de primeiro e segundo grau. Apreciamos o fato de a função de primeiro grau ter apenas uma raiz, enquanto a função de segundo grau ter duas raízes. Também é mostrado como pode-se construir gráficos com as funções.

1 - Objetivo específico:

- ❖ Encontrar as raízes das funções de 1º e 2º grau;
- ❖ Construir o gráfico das funções de 1º e 2º grau;
- ❖ Executar o código e corrigir possíveis erros.

2 – Metodologia:

A atividade será desenvolvida a partir das etapas seguintes:

Etapas 1: Escrever o pseudocódigo no papel, junto com algoritmo em Python

Etapas 2: Utilizar o compilador Python para a realização da proposta;

Etapas 3: Validação de acerto e verificação de erros.

3 – Sequência Didática:

Primeiramente importamos *roots* (raízes) da biblioteca *Numpy*. Definindo o função $f(x) = 2x + 1$ e $f(x) = 2x^2 + 4x - 6$. Em seguida as operações são definidas e montadas. Por último temos a impressão do código, que tem por finalidade o cálculo da solução, ou raízes de uma equação de primeiro e de segundo grau.

De acordo com o código a seguir, em `'from numpy import roots'` fazemos a importação da biblioteca. Em seguida em `'# Define coeficientes'` definimos os coeficientes da função que irão armazenar o valor das operações. Por fim, imprimimos o resultado no console `'print'`, onde podemos ver o resultado de fato.

O resultado é o código da figura abaixo.

Figura 10 - Código Python - Raízes da equação

```
# Importa função de cálculo de raiz (root) da biblioteca numpy.
from numpy import roots

# Limpa a área de console para facilitar a visualização do resultado.
print ('\n' * 100)

# Define coeficientes da Função do Primeiro Grau 2x + 1.
coeficientes = [2, 1]
raiz = roots(coeficientes)

print (u'\n Raiz da função 2x + 1.')
print (raiz)

# Define coeficientes da Função do Segundo Grau 2x**2 +4 x - 6.
coeficientes = [2, +4, -6]
raiz = roots(coeficientes)

print (u'\n Raízes da função 2x**2 +4 x - 6.')
print (raiz)
```

Fonte: produzido pelo Autor, 2021.

Executando, para a saída do programa teremos algo como:

Figura 11 - Saída do programa

```
Raiz da função 2x + 1.
[-0.5]

Raízes da função 2x**2 +4 x - 6.
[-3.  1.]
```

Fonte: produzido pelo Autor, 2021.

Na sequência passamos a execução da construção dos gráficos da função de 1º grau.

Primeiro devemos importar as funções de elaboração de gráficos da biblioteca **pylab**; função **plot** para plotar o gráfico; a função **arange** para definir as faixas de valores; as funções **xlabel** e **ylabel** para incluir rótulos nos eixos; a função **title** para incluir nome do gráfico; a função **grid** para fazer a malha no gráfico; as funções **axhline** e **axvline** para incluir eixos nas origens. Em seguida definimos a Faixa de valores de x, que será de -1 à 3, e a própria equação como $y = 2 * x + 1$. Os demais comandos são para plotagens, e rotulagem. Os escritos em roxo são referentes ao nome e ao rótulo do gráfico.

Figura 12 - Código Python - Definição e formatação do gráfico

```
from pylab import plot, arange, xlabel, ylabel, title, grid, axhline, axvline

# Limpa a área de console para facilitar a visualização do resultado.
print ('\n' * 100)

# Define valores a serem plotados no gráfico.
x = arange(-1, 3, 1) # Faixa de valores para x.
y = 2 * x + 1

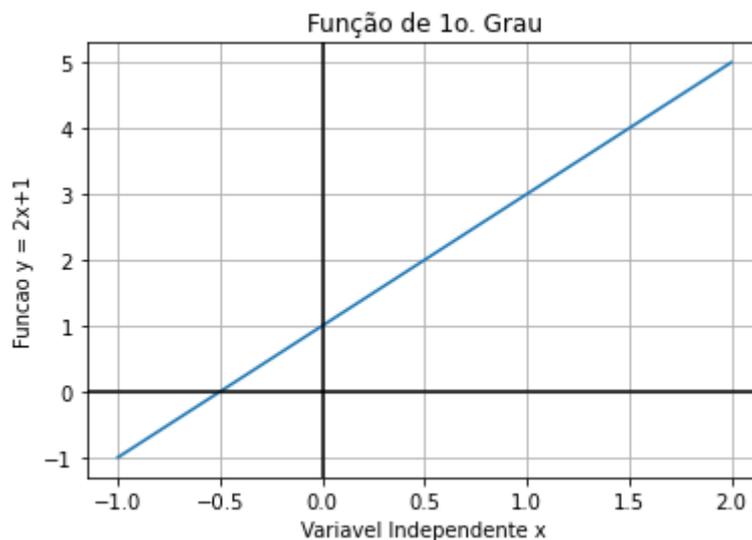
# Exibe gráfico.
plot(x, y)

xlabel('Variavel Independente x') # Inclui rótulo para eixo horizontal.
ylabel('Funcao y = 2x+1') # Inclui rótulo para eixo vertical.
title('Função de 1o. Grau') # Inclui nome no gráfico.
grid(True) # Inclui o grid no gráfico.

# Inclui os eixos na origem - x = 0 e y = 0.
axhline(y=0, color='k') # Traça uma linha horizontal em y = 0.
axvline(x=0, color='k') # Traça uma linha vertical em x = 0.
```

Fonte: produzido pelo Autor, 2021.

Gráfico 1 - Função de 1º Grau



Fonte: produzido pelo Autor, 2021.

Passamos agora para a execução da construção do gráfico da função do 2º grau.

Primeiro devemos importar as funções exatamente como no código acima. Em seguida definimos a faixa de valores de x (*arange*), que será de -10 à 10, e a própria função como

$y = 2x^2 + 4x - 6$. Os demais comandos são para plotagens, e rotulagem. Os escritos em roxo são referentes ao nome e ao rótulo do gráfico.

Figura 13 - Código Python - Definições do gráfico

```
from pylab import plot, arange, xlabel, ylabel, title, grid, axhline, axvline

# Limpa a área de console para facilitar a visualização do resultado.
print ('\n' * 100)

# Define valores a serem plotados no gráfico.
x = arange(-10, 10, 0.01) # Faixa de valores para x.
y = 2 * x ** 2 + 4 * x - 6

# Exibe gráfico.
plot(x, y)

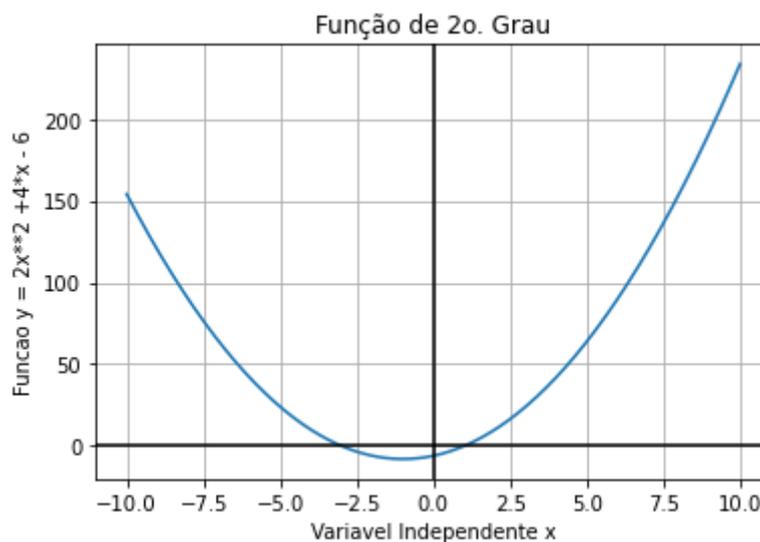
xlabel('Variavel Independente x') # Inclui rótulo para eixo horizontal.
ylabel('Funcao y = 2x**2 +4*x - 6') # Inclui rótulo para eixo vertical.
title('Função de 2o. Grau') # Inclui nome no gráfico.
grid(True) # Inclui o grid no gráfico.

# Inclui os eixos na origem - x = 0 e y = 0.
axhline(y=0, color='k') # Traça uma linha horizontal em y = 0.
axvline(x=0, color='k') # Traça uma linha vertical em x = 0.
```

Fonte: produzido pelo Autor, 2021.

Na sequência temos a saída esperada.

Gráfico 2 - Função de 2º Grau



Fonte: produzido pelo Autor, 2021.

4.4 SEQUÊNCIA DIDÁTICA: PA e PG

Esta atividade visa ensinar conceito de progressão aritmética e geométrica. Apreciamos a ideia da diferença significativa de uma progressão para outra, pelo resultado dos valores comparados. Também é mostrado como se pode construir gráficos com os valores resultantes.

1 - Objetivos específicos:

- ❖ Encontrar os termos de uma Progressão numérica: PA e PG;
- ❖ Determinar a razão e o termo geral de uma sequência: PA e PG.

2 – Metodologia:

A atividade será desenvolvida a partir das etapas seguintes:

Etapa 1: Escrever o pseudocódigo no papel, junto com algoritmo em Python

Etapa 2: Utilizar o compilador Python para a realização da proposta;

Etapa 3: Validação de acertos - as expressões são validadas tecendo **SHIFT+RETURN** ou **F5** e verificação de erros.

3 – Sequência Didática:

(1) Progressão Aritmética (Primeiros dez termos de uma P.A.)

Primeiramente, na Entrada definimos os valores para as variáveis. Neste caso, seja $a = 2$, $r = 3$ e $tam = 7$. Temos o primeiro termo sendo 2, a razão é igual a 3, e a quantidade de termos ou tamanho é igual a 10. Seguimos com a fórmula do termo geral:

$$a_n = a_1 + (n - 1) \cdot r \quad (1)$$

Em seguida, temos que definir a fórmula da P.A., que chamaremos de *partm*, de acordo como colocado anteriormente. Em seguida criamos o laço de repetição com o comando **for n in range()**, que fará o programa repetir a operação até se atingir o valor de $tam = 10$. Ficando da seguinte forma,

Figura 14 - Código Python - Progressão aritmética

```
# Entrada dos valores:
a = 2
r = 3
tam = 10

# Laço de repetição
partm = [ a+r*(n -1) for n in range(1, tam +1)]
print("Primeiro termo:",a)
print("Razão",r)
print("Os primeiros 10 termos da PA são:",partm)
```

Fonte: produzido pelo Autor, 2021.

A saída esperada é uma P.A. de 10 termos como segue,

Figura 15 - Saída do programa - Termos da PA

```
Primeiro termo: 2
Razão 3
Os primeiros 10 termos da PA são: [2, 5, 8, 11, 14, 17, 20, 23, 26, 29]
```

Fonte: produzido pelo Autor, 2021.

(2) Progressão Geométrica (Primeiros dez termos de uma P.G.). Primeiramente, na Entrada definimos os valores para as variáveis. Neste caso, seja $a = 7$, $r = 2$ e $tam = 10$. Temos o primeiro termo sendo 7, a razão é igual a 2, e a quantidade de termos ou tamanho é igual a 10. Seguimos com a fórmula do termo geral:

$$a_n = a_1 \cdot r^{n-1} \quad (2)$$

Em seguida temos que definir a fórmula da P.G., que chamaremos de pg , de acordo com o colocado acima. Em seguida criamos o laço de repetição com o comando **for n in range()**, que fará o programa repetir a operação até se atingir o valor de $tam = 10$. Resultando na seguinte forma,

A saída esperada é uma P.G com dez termos como segue,

Figura 16 - Código Python - Progressão geométrica

```
# Entrada dos valores:
a = 7
r = 2
tam = 10

# Laço de repetição
pg = [ a*r**(n -1) for n in range(1, tam +1)]
print('Primerio termo: ',a)
print('Razão: ',r)
print('PG: ',pg)
```

Fonte: produzido pelo Autor, 2021.

Figura 17 - Saída do programa

```
Primerio termo: 7
Razão: 2
PG: [7, 14, 28, 56, 112, 224, 448, 896, 1792, 3584]
```

Fonte: produzido pelo Autor, 2021.

- Representação gráfica dos termos da PA e PG.

Exercitando a capacidade de criação de gráficos, podemos construir um gráfico com os valores obtidos da P.A e P.G. Para tanto, necessitamos importar ***from pylab import plot, grid*** (biblioteca que nos dá a capacidade de construir um gráfico com uma malha). Esse código pode ser escrito e executado na sequência do algoritmo da P.A e da P.G, a partir da definição dos valores de x e de y. Neste processo irá o nome da variável entre parênteses: ***partm*** para a P.A, e ***pg*** para a P.G. O termo 'd' é a figura em forma de diamante utilizada na postagem.

Figura 18 - Código Python - Definição do gráfico da PA

```
from pylab import plot, grid

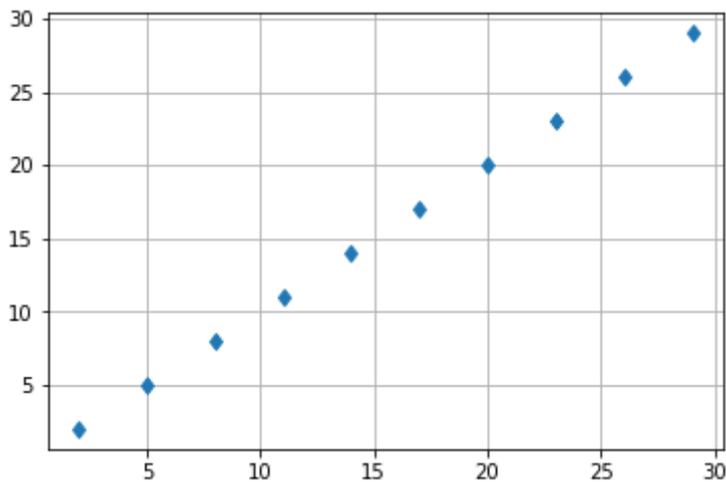
# Define valores a serem plotados no gráfico.
x = (partm)
y = (partm)

# Exibe gráfico.
# Valores de x apresentados no eixo horizontal.
# Valores de y apresentados no eixo vertical.
plot(x, y, 'd')
grid(True) # Inclui o grid no gráfico.
```

Fonte: produzido pelo Autor, 2021.

A saída é o gráfico esperado da P.A

Gráfico 3 - Progressão aritmética



Fonte: produzido pelo Autor, 2021.

De maneira semelhante, podemos construir um gráfico da P.G utilizando os valores calculados, escrevendo logo após o algoritmo o código como segue:

Figura 19 - Código Python - Definição do gráfico da PG

```
from pylab import plot, grid

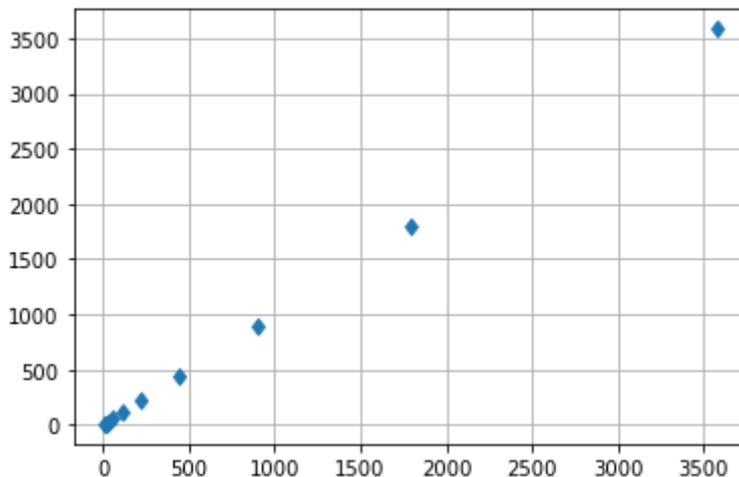
# Define valores a serem plotados no gráfico.
x = (pg)
y = (pg)

# Exibe gráfico.
# Valores de x apresentados no eixo horizontal.
# Valores de y apresentados no eixo vertical.
plot(x, y, 'd')
grid(True) # Inclui o grid no gráfico.
```

Fonte: produzido pelo Autor, 2021.

A saída é o gráfico esperado da P.G,

Gráfico 4 - Progressão geométrica



Fonte: produzido pelo autor, 2021.

4.5 SEQUÊNCIA DIDÁTICA: função exponencial

Esta última atividade visa a resolução de uma função exponencial, e a construção do seu gráfico correspondente. A visualização do gráfico desempenha um papel importante nesse caso ($3a^{x/2}$) e nos demais casos acima, para o estudante conseguir 'ver' a função em sua forma matemática com precisão e rigor.

1 - Objetivos específicos:

- Elaborar uma função exponencial;
- Construir o gráfico da função exponencial;

2 – Metodologia:

A atividade será desenvolvida a partir das etapas seguintes:

Etapa 1: Escrever o pseudocódigo no papel, junto com algoritmo em Python;

Etapa 2: Utilizar o compilador Python para a realização da proposta;

Etapa 3: Validação de acerto e verificação de erros.

3 – Sequência Didática:

Primeiramente definimos a função através do comando **def calcula_f(a, x)** e em seguida digitamos a função que desejamos, no caso $3 \cdot a^{x/2}$. Definimos para esse exemplo os valores de $a = 4$, $x = 4$, e y representa a própria função exponencial.

Figura 20 - Código Python - Definição da função

```
# Define uma nova função. Neste caso, uma Função Exponencial.
def calcula_f(a, x):
    return 3 * a**(x / 2)

# Limpa a área de console para facilitar a visualização do resultado.
print ('\n' * 100)

# Chama a função para calcular o valor quando a = 2 e x = 4.
a = 2
x = 4.0
y = calcula_f(a, x)

print (u'Resultado da função calcula_f de base a = 2 e x = 4.0.')
print ('y =', y)
```

Fonte: produzido pelo Autor, 2021.

A saída esperada é o resultado da função

Figura 21 - Saída do programa

```
Resultado da função calcula_f de base a = 2 e x = 4.0.  
y = 12.0
```

Fonte: produzido pelo Autor, 2021.

Na sequência apresentamos a construção do gráfico de uma função exponencial.

Podemos plotar o gráfico da nossa função exponencial, sem definir os valores de a e x como no exemplo anterior. Se faz necessário a importação da biblioteca que trabalha com gráficos, como os exemplos anteriores.

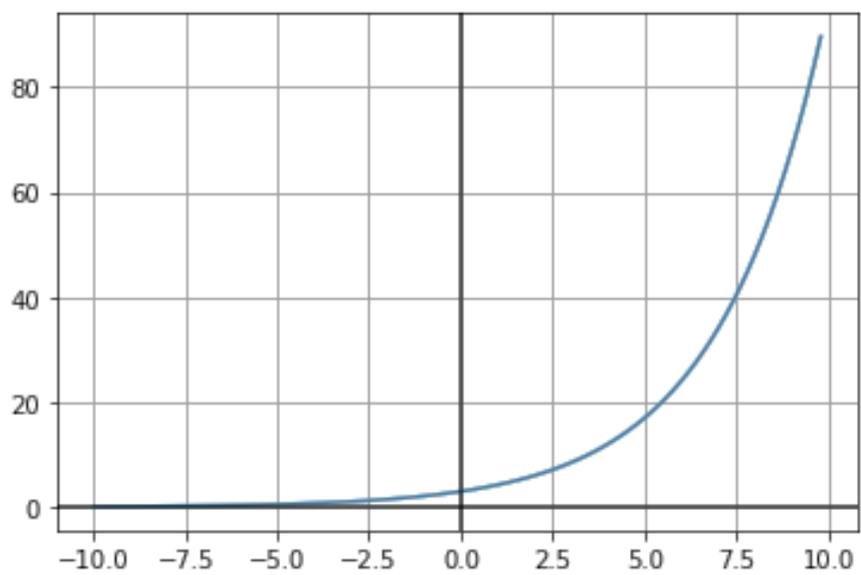
Figura 22- Código Python - Definição da gráfico da função

```
from pylab import plot, arange, grid, axhline, axvline  
  
# Limpa a área de console para facilitar a visualização do resultado.  
print ('\n' * 100)  
  
# Define valores a serem plotados no gráfico.  
x = arange(-10, 10, 0.2) # Faixa de valores para x.  
y = 3 * 2**(x / 2)  
  
# Exibe gráfico.  
plot(x, y)  
  
grid(True) # Inclui o grid no gráfico.  
  
# Inclui os eixos na origem - x = 0 e y = 0.  
axhline(y=0, color='k') # Traça uma linha horizontal em y = 0.  
axvline(x=0, color='k') # Traça uma linha vertical em x = 0.
```

Fonte: produzido pelo Autor, 2021.

A saída esperada é o gráfico da função exponencial como segue.

Gráfico 5 - Função exponencial



Fonte: produzido pelo Autor, 2021.

5 CONSIDERAÇÕES FINAIS

Assim, se pode verificar a eficiência da linguagem Python para o ensino da matemática de maneira plena, em todos os conteúdos propostos pela pesquisa. Os recursos disponíveis na verdade vão muito além, pois são utilizados em diversos ramos na atualidade, com enfoque científico. Os algoritmos, ou pseudocódigos são de fácil construção, se comparados a outras linguagens de programação.

As plataformas, ou diferentes *Softwares*, são ferramentas que podem ser utilizadas de forma livre e gradual; e a internet possui grande quantidade de tutoriais para àquele que deseja se aprimorar. Optamos pelo uso do *Spyder*, porém o *Google Colab* pode apresentar uma maior facilidade, porque não é necessária nenhuma instalação no computador, rodando a partir do site como já citado anteriormente. Em termos didáticos, possivelmente temos o *Jupyter Notebook* como melhor ferramenta, pois é mais didático em sua separação do código.

A sequência didática foi construída visando o aluno aprender matemática, trazendo seu conhecimento aprendido em sala de aula, e aplicando ao computador. Desenvolvendo assim, autonomia para saber o resultado correto dos problemas propostos.

A matemática utilizando o Python não irá parar de evoluir; e talvez o que precisamos, como professores, é introduzir esse assunto aos alunos, e apontar um caminho neste vasto oceano da tecnologia que se abre.

Todos os objetivos propostos foram alcançados. Pois, como foi criado por um matemático, o Python já possui todas as funcionalidades necessárias para os fins propostos. Podemos dizer, que cada biblioteca importada, traz consigo uma infinidade de possibilidades. E essas podem ser exploradas até com fins mais lúdicos, como é o exemplo da Biblioteca *Turtle* (sugestão para futuros estudos), que pode ser utilizada para construções geométricas.

REFERÊNCIAS BIBLIOGRÁFICAS

BIEMBENGUT, Maria Sallet; BIEMBENGUT, Nelson Hein. **Modelagem Matemática no Ensino**. São Paulo. 2000.

BEZERRA, P. S. O Ensino de Matemática nas séries iniciais: Desafios e Necessidades docentes. In: Educação Matemática na Contemporaneidade: desafios e possibilidades. 2016, São Paulo. **Anais do XII Encontro Nacional de Educação Matemática**, São Paulo, SBEM, 2016.

BURAK, D. **Modelagem Matemática** sob um olhar de Educação Matemática e suas implicações para a construção do conhecimento matemático em sala de aula. Revista de Modelagem na Educação Matemática (FURB), v. 1, n. 1, p. 10-27, 2010.

BURAK, D.; KLÜBER, T. E. **Considerações sobre modelagem matemática em uma perspectiva de Educação Matemática**. Margens (UFPA), v. 6, p. 33-50, 2013.

BURAK, D.; KLÜBER, T. E. **Educação Matemática**: contribuições para a compreensão de sua natureza. Acta Scientiae (ULBRA), v. 10, jul-dez, 2008. p. 93-106.

Dravis, P. **Open source software**: perspectives for development. Washington: Infodev. (2003).

FLEMMING, Diva Marília. **Fundamentos da Modelagem Matemática**. Palhoça: Unisul Virtual, 2019.

São Paulo: Atlas, 2002.. GIL, A. C. Como elaborar projetos de pesquisa. São Paulo: Atlas, 2002.

DORNELES, Pedro FT; ARAUJO, Ives S.; VEIT, Eliane A. **Simulação e modelagem computacionais no auxílio à aprendizagem significativa de conceitos básicos de eletricidade**: parte I-circuitos elétricos simples. Revista Brasileira de Ensino de Física, v. 28, n. 4, p. 487-496, 2006.

GODOY, Arilda Schimidt. **Pesquisa Qualitativa**: tipos fundamentais. Revista de Administração de Empresas, v.35, n.3, p.20-29. São Paulo. RAE Artigos. 1995.

LOPEZ. Oscar Ciro; TOLEDO, José Humberto Dias. **Informática aplicada à educação**: livro didático. 2ª Edição. Palhoça: Unisul Virtual, 2011.

MAIA, D. L.; BARRETO, M. C. **Tecnologias digitais na educação**: uma análise das políticas públicas brasileiras. In: Educação, Formação & Tecnologias, v. 5, no 1, p. 47-61, maio, 2012.

MARCONDES, Guilherme A. Barucke. **Matemática com Python**: Um guia prático. Primeira edição. São Paulo: Novatec Editora, 2018.

MIRANDA, Josete Barbosa; SENRA, Luciana Xavier. *Aquisição e desenvolvimento da linguagem*: contribuições de Piaget, Vygotsky e Maturana. 2012.

OLIVEIRA, João Lucas Melo de; Saron, Alexandre. Desenvolvimento de aplicativo educacional auxiliar em python para as disciplinas de mecânica dos fluidos e hidráulica. COBENGE 2020.

Parâmetros Curriculares Nacionais (**PCNs**). Ensino Médio. Brasília: MEC/SEF, 1998.

Parâmetros Curriculares Nacionais (**PCNs**). Ensino Fundamental. Terceiro e quarto ciclos. Brasília: MEC/SEF, 1998.

PESENTE, Guilherme Moraes. O ensino de matemática por meio da linguagem de programação Python. 2019. Dissertação (Mestrado em Ensino de Ciência e Tecnologia) - Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2019.

PALIERINI, Rafael Zacarias; CHINAGLIA, Eliane F.; MASUNAGA, Sueli HATSUMI. Simulações com python e videoanálise do movimento balístico com resistência do ar. COBENGE 2021.

RAMALHO, Geraldo Luis Bezerra. MOREIRA, André Pimentel. PLANTSIM - Uma Plataforma Em Código Livre Para Desenvolvimento De Práticas De Ensino De Automação Industrial. COBENGE, 2021.

Alyrio, Rovigati Danilo. **Métodos e técnicas de pesquisa em administração**. Rio de Janeiro: Fundação CECIERJ, 2009.

SILVA, Fábيا Valéria de Jesus. **Desenvolvendo animações em Python para explorar as ondas trigonométricas, usando Raspberry Pi como tecnologia de suporte**. Fábيا Valéria de Jesus Silva, 2018.

SOUZA, G. M. de O. (2008). **Navegar é preciso**: viagem nas políticas de adoção do software livre nas escolas municipais de Fortaleza. 2008. 162p. Dissertação (Mestrado Acadêmico em Educação – Universidade Estadual do Ceará), Fortaleza.

STADLER, Gesane; ROMANOWSKI, Joana P.; LAZARIN, Luciane; ENS, Romilda T.; VASCONCELLOS, Sílvia. *Proposta pedagógica interacionista*. PUCRS.

ANDRADE, Júnior. A matemática aplicada na Informática <<https://prezi.com/drdjxalj3f/a-matematica-aplicada-na-informatica/>> Acesso em: 19/11/2021.

CLEMENTE, Beatriz. SildePlayer. Acesso em 13/08/2021. Disponível: <https://slideplayer.com.br/slide/1800795/?_gl=1*18qlwjy*_ga*TTMzZ1JNdWxBM0psbWhBbF9SM1NCdmlORUxBT2NISHp1M2hje1kN1RUpxNG5qQkIIYkhaNIVRSHNiRlpYa25Meg> Acesso em: 19/11/2021.

BORGES, Luis Eduardo. **Python para desenvolvedores**. Novatec. São Paulo. 2014. Acesso 15/08/2021. E-book Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=eZmtBAAAQBAJ&oi=fnd&pg=PA14&dq=python+and+matem%C3%A1tica&ots=VENuwmBnft&sig=6XHIGQ7iCy_vBwsMUjCpK_MGrWQ#v=onepage&q=python%20and%20matem%C3%A1tica&f=false> Acesso em: 19/11/2021.

COELHO, Flávio Codeço. **Computação Científica com Python**. Rio de Janeiro. 2007. Acesso 16/08/2021. E-book. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=yHw25UKduJIC&oi=fnd&pg=PR10&dq=python+and+matem%C3%A1tica&ots=b3M1VzKq8p&sig=48BLvHf_iAWKXXwsgGyvjhllTBw#v=onepage&q=python%20and%20matem%C3%A1tica&f=false> Acesso em: 19/11/2021.

FOSSILE, Dieysa K. Construtivismo *versus* sociointeracionismo: uma introdução às teorias cognitivas. *Revista Alpha*, Patos de Minas, UNIPAM. 2010. Disponível em: http://alpha.unipam.edu.br/documents/18125/23730/construtivismo_versus_socio_interacionsimo.pdf. Acesso em: 19/11/2021.

Site Oficial Python. <<https://www.python.org>> Acesso em: 19/11/2021.

Site Compilador Ipython online: <<https://algoritmosempython.com.br/cursos/programacao-python/modulos-bibliotecas/>> Acesso em: 19/11/2021.

Site Tutoriar <<https://www.tutoriar.com.br/curso/matematica-para-1-ano-do-ensino-medio/>> Acesso em: 19/11/2021.

Site Anaconda Software: <<https://www.anaconda.com/>> Acesso em: 19/11/2021.

Site Google Colaboratory: <<https://colab.research.google.com/>> Acesso em: 19/11/2021.