

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS AGRÁRIAS
DEPARTAMENTO DE CIÊNCIAS DA NATUREZA, MATEMÁTICA E EDUCAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO EM CIÊNCIAS E
MATEMÁTICA

MATEUS XAVIER YAMAGUTI

**ENSINO DE FÍSICA POTENCIALIZADO PELA APRENDIZAGEM DA
LINGUAGEM DE PROGRAMAÇÃO PYTHON: UMA SEQUÊNCIA
DIDÁTICA BASEADA NOS TRÊS MOMENTOS PEDAGÓGICOS**

ARARAS -SP

2024

MATEUS XAVIER YAMAGUTI

**ENSINO DE FÍSICA POTENCIALIZADO PELA APRENDIZAGEM DA
LINGUAGEM DE PROGRAMAÇÃO PYTHON: UMA SEQUÊNCIA DIDÁTICA
BASEADA NOS TRÊS MOMENTOS PEDAGÓGICOS**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Educação em Ciências e Matemática da Universidade Federal de São Carlos, para obtenção do título de mestre em Educação em Ciências e Matemática.

Orientador: Prof. Dr. João Teles de Carvalho Neto.

ARARAS - SP

2024

Yamaguti, Mateus Xavier

Ensino de física potencializado pela aprendizagem da linguagem de programação python: uma sequência didática baseada nos três momentos pedagógicos / Mateus Xavier Yamaguti -- 2024.
174f.

Dissertação (Mestrado) - Universidade Federal de São Carlos, campus Araras, Araras
Orientador (a): João Teles de Carvalho Neto
Banca Examinadora: Tathiane Milaré, Nelson Canzian da Silva
Bibliografia

1. Linguagem de programação Python. 2. Ensino de Física. 3. Alfabetização Científica e Tecnol. I. Yamaguti, Mateus Xavier. II. Título.

Ficha catalográfica desenvolvida pela Secretaria Geral de Informática (SIn)

DADOS FORNECIDOS PELO AUTOR

Bibliotecário responsável: Maria Helena Sachi do Amaral - CRB/8
7083



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Agrárias
Programa de Pós-Graduação em Educação em Ciências e Matemática

Folha de Aprovação

Defesa de Dissertação de Mestrado do candidato Mateus Xavier Yamaguti, realizada em 04/07/2024.

Comissão Julgadora:

Prof. Dr. João Teles de Carvalho Neto (UFSCar)

Profa. Dra. Tathiane Milaré (UFSCar)

Prof. Dr. Nelson Canzian da Silva (UFSC)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Educação em Ciências e Matemática.

AGRADECIMENTOS

Gostaria de agradecer primeiramente a minha esposa Amanda, por todo apoio e paciência durante esse período dedicado à escrita de minha dissertação e ao longo de todo o mestrado. Além de me ajudar, também ofereceu diversas críticas construtivas em momentos de dúvidas e incertezas, me deu forças quando o desânimo e cansaço eram fortes demais para carregar sozinho. Aproveitei também para parabenizá-la pela recém conquista do seu mestrado e por todo empenho. Obrigado Amanda, por sempre me inspirar e dar forças.

Quero expressar minha gratidão ao meu orientador, professor João Teles, pela orientação excepcional, apoio e dedicação ao longo deste processo de pesquisa. Sua experiência e expertise foram fundamentais para a realização deste trabalho. Além de um orientador exemplar, não deixou de compartilhar conhecimento acadêmico, sugestões perspicazes, feedbacks construtivos e encorajamento. Sou imensamente grato por ter tido a oportunidade de trabalhar sob sua orientação.

Agradeço também aos membros da banca examinadora, professora Tathiane Milaré e professor Nelson Canzian, por dedicarem seu tão precioso tempo e experiência na avaliação deste trabalho e por seus comentários construtivos, sempre com a intenção de melhorar minha pesquisa.

À minha família, expressei minha gratidão pelo apoio desde a época da graduação e por todos esse percurso acadêmico, por compreender minha ausência nas festas e reuniões familiares. Mesmo assim, suas palavras de incentivo foram sempre uma fonte de inspiração.

Aos meus amigos, colegas de mestrado e corpo docente do Programa de Pós-Graduação em Educação em Ciência e Matemática (PPGE dCM) da UFSCar campus Araras, agradeço pelo incentivo e colaboração durante esse trajeto. Também à Fundação de Apoio Institucional (FAI) de desenvolvimento científico e tecnológico da UFSCar pelo auxílio financeiro que me permitiu uma dedicação integral no período final do mestrado e na realização dessa pesquisa.

RESUMO

Entendendo a necessidade da renovação do ensino de Física atual, com a utilização de estratégias inovadoras como a utilização de recursos digitais, esta pesquisa de cunho qualitativo e exploratório busca discutir quais as potencialidades que a linguagem de programação Python possui no ensino de Física aliada à metodologia de ensino dos Três Momentos Pedagógicos (3MP). Para isso foi utilizada a metodologia de pesquisa de estudo de caso, aplicada em grupo de alunos do ensino médio de uma escola pública do interior do estado de São Paulo no contraturno escolar de forma remota. Para tornar essa pesquisa possível, o professor pesquisador desenvolveu materiais e recursos didáticos digitais com o objetivo de potencializar o ensino e, também, alcançar o desenvolvimento da Alfabetização Científica e Tecnologia (ACT) dos alunos, relacionando os conceitos de Física e da linguagem de programação Python, sustentada pelos 3MP. A partir da análise de conteúdo dos dados constituídos na pesquisa foi possível estabelecer três categorias, indícios de aprendizagem em Física e/ou programação, problematização e/ou aplicação dos assuntos abordados em aula e indicativos de potencialidade ou limitação do ensino de Física com programação. Por fim, essas categorias revelaram que existem potencialidades como a promoção da compreensão dos conceitos físicos e visualização de seus fenômenos. Todavia, ainda há necessidade de criar recursos e estratégias que minimizem o impacto inicial da linguagem de programação e promovam maior afinidade com o público estudantil.

Palavras - chave: Linguagem de programação Python, Ensino de Física, Três Momentos pedagógicos.

ABSTRACT

Understanding the need to renew current physics teaching, with the use of innovative strategies such as the use of digital resources, this qualitative and exploratory research seeks to discuss the potential that the Python programming language has in teaching Physics combined with methodology of teaching the Three Pedagogical Moments (3PM). For this, the Case Study research methodology was used, applied to a group of high school students from a public school in the interior of the state of São Paulo during the after-school shift remotely. To make this research possible, the research professor developed digital teaching materials and resources with the aim of enhancing teaching, and also achieving the development of Scientific Literacy and Technology (SLT) in students, relating the concepts of Physics and programming language Python, supported by 3PM. From the content analysis of the data constituted in the research, it was possible to establish three categories, signs of learning in Physics and/or programming, problematization and/or application of the subjects covered in class and indications of potential or limitations of teaching Physics with programming. Finally, these categories revealed that there are potentialities such as promoting the understanding of physical concepts and visualization of their phenomena. However, there is still a need to create resources and strategies that minimize the initial impact of the programming language and promote greater affinity with the student audience.

Keywords: Python programming language, Physics Teaching, Three pedagogical moments.

LISTA DE QUADROS

Quadro 1 - Exemplo de tipos de objetos em Python.....	39
Quadro 2 - Exemplo de operações matemáticas feitas em Python.....	40
Quadro 3 - Exemplo operadores relacionais em Python.....	40
Quadro 4 - Ementa da sequência didática FIS_PY.....	64
Quadro 5 - Quadro de ocorrência de visualizações de atividades e recursos.....	80
Quadro 6 - Sintetização dos significados das categorias formuladas.....	87

LISTA DE TABELAS

Tabela 1 - Quadro de alunos inscritos.....	71
Tabela 2 - Frequência dos alunos.....	79
Tabela 3 - Distribuição dos tipos de questões por atividade e encontro.....	82
Tabela 4 - Nota das atividades por aluno.....	82
Tabela 5 - Detalhamento das questões corrigidas por atividades.....	83
Tabela 6 - Ranking crescente de índice de acerto por questões.....	84

LISTA DE GRÁFICOS

Gráfico 1 - Gráfico do MRU de uma partícula.....	36
Gráfico 2 - Deslocamento de uma partícula em MRUV.....	37

LISTA DE EQUAÇÕES

Equação 1 - Diferença espacial no eixo x.....	35
Equação 2 - Velocidade média no eixo x.....	35
Equação 3 - Aceleração média.....	36
Equação 4 - Velocidade da partícula em queda livre.....	36
Equação 5 - Variação espacial no eixo y de uma partícula em queda livre.....	37
Equação 6 - Posição vetorial de uma partícula em duas dimensões.....	37
Equação 7 - Equação do deslocamento vetorial de uma partícula em duas dimensões.....	37
Equação 8 - Equação da velocidade vetorial de uma partícula se deslocando em duas dimensões.....	37
Equação 9 - Variação espacial no eixo x em um movimento balístico.....	38
Equação 10 - Variação espacial no eixo y em um movimento balístico.....	38
Equação 11 - Velocidade da partícula no eixo x em relação a variação angular.....	38
Equação 12 - Velocidade da partícula no eixo y em relação a variação angular.....	38

LISTA DE FIGURAS

Figura 1 - Sentido do deslocamento de uma partícula.....	35
Figura 2 - Decomposição do lançamento oblíquo.....	38
Figura 3 - Página de login do AVA Moodle CodeGenius.....	51
Figura 4 - Painel principal com ênfase no calendário de atividades e recursos.....	51
Figura 5 - Página principal de cursos.....	52
Figura 6 - Modelo em linguagem HTML da questão padrão a ser utilizadas nas demais atividades CodeRunner.....	53
Figura 7 - Modelo ilustrativo da questão padrão resultado da renderização do exemplo da Figura 6 a ser usado nas demais atividades CodeRunner.....	54
Figura 8 - Ilustração da resposta à pergunta da Figura 7.....	55
Figura 9 - Ilustração da configuração de teste realizada no CodeRunner referente a figura 8.....	55
Figura 10 - Ilustração de questão CodeRunner da figura 8 sendo verificada automaticamente.....	56
Figura 11 - Exemplo de questão de arrastar e soltar no Moodle personalizada através da linguagem de marcação HTML.....	57
Figura 12 - Exemplo de questão de múltipla	58
Figura 13 - Imagem do slide da primeira aula do curso FIS_PY.....	63
Figura 14 - Imagem do slide da segunda aula do curso FIS_PY	64
Figura 15 - Mapa conceitual sobre o desenvolvimento da análise de conteúdo.....	77
Figura 16 - Curva da quantidade de visualizações das atividades por encontro.....	81
Figura 17 - Exemplo de questão A1Q6 com erro no ponto flutuante da casa decimal do número pi.....	85
Figura 18 - Exemplo de questão A2Q3 de sintaxe.....	86
Figura 19 - Imagem retirada após a execução do código Corrida de Alunos.....	92
Figura 20 - Imagem retirada após a execução do código Trabalho Curso.....	95

LISTA DE SIGLAS

3MP - Três Momentos Pedagógicos

AC – Alfabetização Científica

ACT - Alfabetização Científica e Tecnológica

AT - Alfabetização Tecnológica

AVA - Ambiente Virtual de Aprendizagem

BNCC – Base Nacional Comum Curricular

CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

CBEF - Caderno Brasileiro de Ensino de Física

CTS - Ciência, Tecnologia e Sociedade

CWI - Centro de Matemática e Informática

DERDND - Descrição - Execução - Reflexão - Depuração - Nova Descrição

EPEF - Encontro de Pesquisa e Ensino de Física

IA - Inteligência Artificial

IDE - Integrated Development Environment

MRU - Movimento Retilíneo e Uniforme

MRUV - Movimento Retilíneo e Uniformemente Variado

NSTA - National Science Teaching Association

PSF - Python Software Foundation

RBEF - Revista Brasileira de Ensino de Física

RCAAP - Repositório Científico de Acesso Aberto de Portugal

SGA – Sistema de Gestão de Aprendizagem

SNEF - Simpósio Nacional de Ensino de Física

SQL - Structured Query Language

TCLE - Termo de Consentimento Livre e Esclarecido

TDD - Test Driven Development

TDIC - Tecnologia Digitais da Informação e Comunicação

VM - Virtual Machine

SUMÁRIO

RESUMO.....	4
ABSTRACT.....	5
SUMÁRIO.....	13
APRESENTAÇÃO.....	14
INTRODUÇÃO.....	17
Questão de Pesquisa.....	19
OBJETIVOS DA PESQUISA.....	20
1. FUNDAMENTAÇÃO TEÓRICA.....	21
1.1 Tecnologias da informação e comunicação e a linguagem de programação Python....	21
1.2 Alfabetização Científica e Tecnológica no ensino de ciências.....	28
1.3 O ensino de Física a partir dos Três Momentos Pedagógicos.....	32
1.4 Conteúdo de Física.....	34
1.4.1 Movimento retilíneo e uniforme.....	35
1.4.2 Movimento retilíneo e uniformemente variado.....	36
1.4.3 Movimento balístico.....	37
1.5 Linguagem de programação Python.....	38
1.5.1 Variáveis.....	39
1.5.2 Operações matemáticas e relacionais.....	39
1.5.3 Funções em Python.....	41
1.5.4 Laços ou Loops.....	42
1.5.5 Testes if e else.....	43
1.5.6 Importar biblioteca e módulos.....	43
2. MATERIAIS E ESTRATÉGIA DE ENSINO.....	45
2.1 Moodle.....	50
2.2 Atividades.....	52
2.2.1 Questões CodeRunner.....	53
2.2.2 Atividade de arrastar e soltar e de múltipla escolha.....	57
2.2.3 Bibliotecas Gráficas em Python.....	59
2.3 Metodologia de Ensino.....	60
2.4 A sequência didática.....	61
2.4.1 Material Didático.....	63
2.4.2 Planos de aula FIS_PY.....	64
3. METODOLOGIA DE PESQUISA.....	69
3.1 Estudo de Caso.....	70
3.2 Unidade de análise.....	71
3.3 Instrumentos para constituição de dados.....	72
3.4 Metodologia de análise de dados.....	73
3.4.1 Análise de Conteúdo.....	74
4. ANÁLISE DOS DADOS.....	78

4.1	Frequência dos Alunos.....	78
4.2	Atividade e recursos.....	80
4.2.1	Atividades Questionário.....	81
4.3	Análises e Resultados.....	87
4.3.1	Indícios de aprendizagem em Física e/ou programação.....	88
4.3.2	Problematização e/ou aplicação do assuntos abordados em aula.....	96
4.3.3	Indicativos de potencialidade ou limitação do ensino de Física com programação	98
	CONSIDERAÇÕES FINAIS.....	104
	REFERÊNCIAS BIBLIOGRÁFICAS.....	107
	APÊNDICE A - DOCUMENTAÇÃO DA INSTALAÇÃO DA APLICAÇÃO MOODLE..	112
	APÊNDICE B - PROGRAMAS MRU FEITO PARA TESTE DE BIBLIOTECAS.....	119
	APÊNDICE C - SCRIPT PARA TRANSCRIÇÃO DE ÁUDIOS.....	122
	APÊNDICE D - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)....	123
	APÊNDICE E - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)....	126
	APÊNDICE F - CARTA DE AUTORIZAÇÃO ESCOLAR.....	128
	APÊNDICE G - TERMO DE CONFIDENCIALIDADE.....	129
	APÊNDICE H - TERMO DE AUTORIZAÇÃO PARA USO DE IMAGEM E SOM....	130
	APÊNDICE I - SLIDES UTILIZADOS EM AULA.....	132
	SLIDES AULA 01.....	132
	SLIDES AULA 02.....	140
	SLIDES AULA 03.....	147
	SLIDES AULA 04.....	151
	APÊNDICE J - ATIVIDADES.....	156
	QUESTÕES AULA 02.....	156
	QUESTÕES AULA 03.....	161
	QUESTÕES AULA 04.....	166

APRESENTAÇÃO

Meu nome é Mateus Xavier Yamaguti autor desta pesquisa. Dedico esse espaço para apresentar minha jornada acadêmica, profissional e pessoal que levaram meu profundo interesse pela interseção entre a educação em física e a linguagem de programação Python, o que me levou a explorar o uso dessas ferramentas como recurso potencializador de ensino e aprendizagem e também me fez escolher alguns recursos utilizados nesta pesquisa.

Sou graduado em licenciatura em Física pela UFSCar, onde tive a oportunidade de aprofundar meus conhecimentos em Física, ensino e aprendizado, além de adquirir parte das habilidades em programação. Durante a graduação, e principalmente nas disciplinas iniciais de laboratório que tem como base a física clássica, pude perceber que, ao realizar as experimentações, existe uma considerável necessidade de coleta de dados.

Que por sua vez, culmina em um grande trabalho braçal para elaborar os relatórios experimentais, com cálculos repetitivos a fim de se estabelecer desvio padrão, propagação de erro, além da tarefa de construir gráficos a punho, que consome um período grande de tempo da elaboração dos relatórios, que poderia ser destinado a outras atividades mais conceituais como análise e interpretação dos dados e do fenômeno estudado.

Como aluno, me senti frustrado com essa metodologia de ensino, por se tratar de uma sequência cansativa de cálculos repetitivos, mecânicos e que consumia uma considerável quantidade de tempo que poderia ser reduzida, a fim de aprender mais intensamente na aplicação das teorias estudadas, assim como, no entendimento do fenômeno estudado, nas implicações que esses fenômenos físicos podem ter na sociedade, e na problematização das situações reais no contexto da vida dos estudantes.

No entanto, felizmente esse cenário foi alterado na disciplina de Laboratório de Física Moderna que cursei durante minha graduação, uma vez que a proposta do professor foi diferente das demais disciplinas de laboratórios, principalmente por incorporar o uso da linguagem de programação Python, a fim de auxiliar na construção do relatório experimental e para realizar os cálculos.

Naquela disciplina os experimentos eram divididos em três etapas: I) Construção do Pré-relatório, II) Experimentação e III) Relatório Final. A etapa I consistia na pesquisa do fenômeno a ser estudado pelo aluno na construção do Pré-relatório com a fundamentação teórica do fenômeno, na elaboração do roteiro experimental a ser executado na etapa II, assim

como, na elaboração prévia dos programas em linguagem Python. E, por fim, é apresentado o Pré-relatório ao professor, no intuito de ajudar a mitigar qualquer equívoco experimental, refinar o entendimento sobre o fenômeno pesquisado e a refatorar pontos de melhorias nos códigos em Python.

Posteriormente, a etapa II compreendia a execução do roteiro experimental elaborado na etapa I, assim como na coleta de dados. Finalmente, na etapa III, eram utilizados os programas na linguagem Python para tratar todos os dados coletados, para posteriormente o estudante poder analisá-los e interpretá-los, gerando suas considerações finais sobre o fenômeno estudado, as conclusões sobre a prática laboratorial e a aplicação que esse fenômeno possui no contexto social.

A partir dessa experiência vivenciada no contexto da disciplina de Laboratório de Física Moderna, pude perceber que o uso da linguagem de programação Python potencializou dois pontos importantes no contexto geral da disciplina. Primeiramente, a potencialização do entendimento conceitual do fenômeno físico, pelo fato do estudante precisar desenvolver os cálculos em uma linguagem de programação e entender se o resultado do código em Python representa a realidade do fenômeno e, posteriormente, mas não menos importante, a utilização dos programas construídos para efetuar os cálculos necessários, que por sua vez diminui muito o tempo gasto nessa etapa se comparado com os cálculos mecânicos e repetitivos praticados nas disciplinas laboratoriais de física clássica.

Sendo assim, a otimização do tempo proporcionada pelo uso da linguagem de programação Python, proporcionou uma dedicação maior de tempo na análise e interpretação dos dados, assim como na compreensão do fenômeno e na sua aplicação social.

A partir daquele contexto surgiu a motivação que impulsionou a realização desta pesquisa, de proporcionar ao estudante maior entendimento sobre o conhecimento científico que está sendo estudado, como também o desdobramento que os conhecimentos científicos possuem no contexto social e suas aplicações.

Além disso, durante quase todo o período da minha graduação e início do mestrado, pude trabalhar em uma *startup* de tecnologia no interior do estado de São Paulo, onde fui responsável pela criação da universidade corporativa da empresa.

Nesta experiência, pude criar trilhas educacionais voltadas para o ensino de linguagem de programação Python e linguagem de consulta estruturada (SQL - *Structured Query Language* utilizada em banco de dados), conteúdo em formato de vídeo aula, exercícios e

trabalhar com pessoas de diversas etnias e idades. Durante esse projeto educacional, também pude desenvolver dois artigos baseado no estudo de caso da universidade corporativa e apresentar seus resultados em dois congressos (Yamaguti; Arle, 2022; Yamaguti *et al.*, 2022).

Além de criar recursos educacionais, também tive a oportunidade de trabalhar com desenvolvedores de *softwares* e aplicativos *backend*, *frontend* e *fullstack*, a fim de criar a aplicação do ambiente virtual de aprendizado baseado no Moodle, o me levou a desenvolver várias habilidades na área de tecnologia da informação e principalmente na linguagem de programação Python.

Motivado pelo desejo de explorar novas abordagens educacionais e tecnológicas, ingressei no PPGEdCM da UFSCar no campus da cidade de Araras, onde concentrei meus estudos na área de ensino de física aliada ao uso da linguagem de programação Python.

Esta dissertação não é apenas o resultado do meu compromisso com a academia, mas sim a aplicação do meu desenvolvimento profissional, pessoal e acadêmico em uma pesquisa que espero contribuir para o avanço do conhecimento científico e inspire outros estudantes e educadores a explorarem novas formas de ensino e aprendizagem.

INTRODUÇÃO

O desenvolvimento da Física muda o mundo, trabalha contra o senso comum e luta contra os apelos do misticismo e aceitações acríticas. O surgimento ou os primeiros relatos de estudos dos fenômenos naturais que se tem registro são dos gregos por volta do século VI antes de Cristo. Antes disso, os babilônios e os egípcios estudavam astronomia e matemática, porém para fins práticos, com o intuito de prever estações dos anos, inundações do Nilo e até mesmo para fins religiosos (Pires, 2011).

Foi na Grécia, no século VI antes de Cristo, que os primeiros filósofos começaram a se perguntar a causa e a origem dos fenômenos naturais e excluíram a explicação dos deuses, uma ruptura colossal para a sociedade da época. Também vale lembrar que o termo “físico” ainda não havia sido cunhado nesta época, apenas em 1840 o filósofo e cientista inglês William Whewell concebe a palavra “físico” em sua obra “The Philosophy of the Inductive Sciences, founded upon their history¹” (Pires, 2011).

Desde então o componente curricular de Física tem a intenção de ensinar os fenômenos naturais, logo, muitos dos conteúdos desta ciência estão diretamente ligados ao cotidiano, a uma série de acontecimentos que ocorrem em diferentes situações, dando significado para a vida, o que desempenha um papel fundamental na formação dos alunos, pois fornece uma compreensão fundamental das leis e princípios que governam o mundo físico.

No entanto, o ensino de Física no Brasil enfrenta desafios graves e estruturais devido a formação docente insuficiente, as más condições de trabalho, o reduzido número de aulas no ensino médio, da progressiva perda de identidade da Física no currículo e do aprendizado mecanizado e desatualizado de Física, visto que estamos no século XXI, mas a Física ensinada não passa do século XIX (Moreira, 2017).

Perante estes desafios, esta pesquisa busca encontrar um modelo inovador para superar o modo como o ensino de Física clássica é realizado atualmente no Brasil e até mesmo propor uma estratégia de ensino aliando recursos de Tecnologia Digitais da Informação e Comunicação (TDIC) buscando entender se o uso da linguagem de programação Python no ensino de Física aliada a uma metodologia de ensino contemporânea pode ser uma ferramenta potencializadora para o ensino de Física.

¹ “A Filosofia das Ciências Indutivas, fundada em sua história”. Livro não traduzido para o português.

Desta forma, esta pesquisa além de fazer o uso de recursos digitais e tecnológicos aliados ao ensino de ciências, também pode ser considerada um potencializador da Alfabetização Científica e Tecnológica (ACT), levando em conta o seu caráter ampliado como exposto por Auler e Delizoicov (2001) em compreender as interações entre Ciência, Tecnologia e Sociedade (CTS).

Essa visão parte do conceito de que para uma dinâmica social crescente é fundamental a democratização dos avanços científico-tecnológicos, visto que segundo o relatório apresentado pelo Fórum Econômico Mundial (World Economic Forum, 2023), até o ano de 2027 cerca de 23% de todos os trabalhos atuais serão alterados pela adoção de tecnologia e o aumento do acesso digital, o crescente uso de Inteligência Artificial (IA), cuidado com a sustentabilidade, aperfeiçoamento da inteligência de negócios e segurança da informação, com grande destaque nos setores de comércio digital, educação e agricultura. As mudanças nesses setores terão impacto significativo no mercado de trabalho, uma vez que o relatório indica que ocorrerá a criação de 69 milhões de novos empregos nessas áreas, o que ocasionará a eliminação de 83 milhões de empregos. O relatório também sugere a necessidade de uma possível requalificação dos trabalhadores que queiram transitar de emprego, o que torna ainda mais importante o desenvolvimento da ACT.

Com isso, podemos nos perguntar: como está sendo inserido o aluno do ensino médio brasileiro nesta transição do mercado de trabalho? Será que este estudante será capaz de sustentar o mercado de trabalho e suprir as necessidades das mudanças empregadas pela tecnologia? As habilidades requeridas estão realmente inseridas no contexto escolar? Talvez caiba uma reflexão para entender se as habilidades ensinadas atualmente no ensino básico estão alinhadas com as principais habilidades para o futuro do mercado de trabalho. Entre essas habilidades estão: pensamento analítico, pensamento criativo, uso das IA e *big data*, liderança e influência social, resiliência, flexibilidade e agilidade (World Economic Forum, 2023).

Porém, segundo Moreira (2017), o ensino de ciências no Brasil ainda continua treinando os estudantes para testes, centrado no docente, seguindo o modelo “bancário” de aprendizado elencado por Freire (2005) e não incorpora as TDIC. O que nos permite deduzir que o ensino de ciências no Brasil nas escolas de educação básica não valoriza as habilidades necessárias para a sustentação do mercado de trabalho e ainda desestimula o desenvolvimento de capacidades importantes para a sobrevivência em sociedade, criando a hipótese de um colapso futuro tanto no mercado de trabalho, quanto na educação e na economia.

Sendo assim, este estudo contou com o uso de recursos tecnológicos para promover o ensino de Física em um ambiente que também proporcione alfabetização tecnológica. Esses recursos consistem basicamente em um Ambiente Virtual de Aprendizagem (AVA) composto por ferramentas de ensino e atividades digitais, aulas síncronas remotas em sala de videoconferência e também o uso da linguagem de programação Python, que serão descritos com maiores detalhes no decorrer desta dissertação.

Aliado ao contexto tecnológico, foi escolhido como metodologia de ensino para a intervenção realizada nesta pesquisa os Três Momentos Pedagógicos (3MP) (Muenchen; Delizoicov, 2014), pois em sua concepção é possível problematizar temas e assuntos inerentes a fenômenos físicos e relacionar o ensino de Física com ACT e TICs de forma ampla, considerando os temas que fazem sentido na vida do estudante, a fim de proporcionar maior engajamento e motivação nas aulas e tornar o aprendizado mais próximo de suas vidas e conceitualmente mais aplicado. Essa aplicação refere-se ao uso de situações Físicas reais do cotidiano dos estudantes traduzidas em linguagem de programação Python no intuito de promover o aprendizado do conteúdo proposto, por meio da racionalização lógica, desenvolvimento de funções em Python, simulações digitais, criação e análise de gráficos digitais, entre outras aplicações.

Para compreender todo esse contexto, foi proposto pelo pesquisador professor um estudo de caso qualitativo de caráter revelador, ou seja, que busca estudar um fenômeno pouco explorado ou previamente inacessível, com ampla gama de constituição de dados, inserido em uma sequência didática autoral denominada de FIS_PY (nome escolhido para ajudar na estratégia de divulgação), aplicada em alunos do ensino médio de uma escola técnica no interior de São Paulo, compreendendo um total de cinco (5) encontros de uma hora e meia (1h 30 min) de duração cada, estruturados segundo os 3MP para ensinar o conteúdo de cinemática aliado ao uso da linguagem de programação Python.

Questão de Pesquisa

Diante dos desafios do ensino de Física no Brasil, da baixa utilização das TDIC, e da democratização dos conhecimentos científicos, esta pesquisa busca responder a seguinte questão: Quais as potencialidades e limitações da linguagem de programação Python para o ensino e aprendizagem de Física, organizados por meio de uma sequência didática baseada nos três momentos pedagógicos (3MP)?

OBJETIVOS DA PESQUISA

Esta pesquisa tem o objetivo geral de analisar as potencialidades e limitações da utilização da linguagem de programação Python como recurso didático para o ensino de Física a partir de uma sequência didática organizada com base nos Três Momentos Pedagógico (3MP).

Além disso, busca, como objetivos específicos, entender se o pensamento computacional necessário para traduzir os fenômenos físicos em um programa em Python que simule esses fenômenos pode potencializar o aprendizado; além de ajudar no desenvolvimento de habilidades cognitivas, racionalização, socialização, interpretação do mundo, criatividade e resolução de situações-problema de Física.

1. FUNDAMENTAÇÃO TEÓRICA

Neste primeiro capítulo, nos dedicaremos em apresentar as fundamentações teóricas desta pesquisa ao fazer um resgate histórico sobre os assuntos que a permeiam, um levantamento do que se tem pesquisado sobre elas, apresentação dos principais referenciais e também abordaremos o conteúdo de Física e de linguagem de programação Python ministrado pelo pesquisador professor na sequência didática FIS_PY.

Para tanto, o separamos em cinco subtópicos de modo que o primeiro busca apresentar o conceito de Tecnologias da Informação e Comunicação (TIC), como essas se aplicam ao ensino, e também, a apresentação da linguagem de programação Python escolhida como recurso desta pesquisa. Logo após, no segundo subtópico, nos debruçamos nos conceitos de Alfabetização Científica e Tecnológica (ACT) para o ensino de ciências nos baseando em alguns autores da área. Além disso, no terceiro subtópico, apresentamos a metodologia de ensino que utilizamos na intervenção realizada nesta pesquisa, denominada Três Momentos Pedagógicos.

Por fim, os dois últimos subtópicos desta fundamentação são destinados à explanação do conteúdo de Física e de linguagem de programação Python, uma vez que eles são referência para o capítulo sobre materiais e estratégia de ensino.

1.1 Tecnologias da informação e comunicação e a linguagem de programação Python

Desde o início da humanidade o domínio de algumas tecnologias e da informação diferencia os seres humanos dos demais animais (Kenski, 2009). Existem autores que definem o surgimento do homem racional a partir da criação da tecnologia (Ortega y Gasset, 1963). Na Idade da Pedra os homens puderam garantir sua sobrevivência mediante o domínio dos elementos da natureza, como estocar água, se aquecer com fogo, se proteger com pedaço de pau ou ossos de animais. Afinal, era necessário que o homem suprisse suas fragilidades naturais da espécie para manter sua sobrevivência e desenvolver sua supremacia (Kenski, 2009). Essa mesma relação pode ser constatada facilmente em nosso cotidiano, nas cadeias produtivas humanas, no formato com que nos comunicamos, nos padrões de habitação, dentre muitos outros modelos de vida humana que desenvolvemos com o uso, manejo e transformação das fontes e recursos naturais em comparação com as outras espécies.

Assim, o uso da tecnologia apenas se intensificou com o passar do tempo, potencializando ainda mais essa relação entre homem, tecnologia e supremacia. Hoje, as grandes potências e corporações multinacionais destinam parte de suas verbas para o desenvolvimento de inovações tecnológicas com o objetivo de manter seu domínio e hegemonia, e, em muito desses casos, as pesquisas que são realizadas levam à ampliação e produção de novos produtos que migram para as casas, alterando o modo de vida de toda a sociedade (Kenski, 2009).

Muitos desses desenvolvimentos foram resultados de pesquisas intensas realizadas durante os momentos mais conturbados da história mundial. Segundo o jornalista Fábio Reynold (2004) aparelhos automáticos como o aferidor de pressão arterial foram desenvolvidos primordialmente para atender a necessidade de aferir a pressão arterial dos astronautas de forma prática durante a corrida espacial, “a válvula de um novo tipo de coração artificial foi inspirada em uma bomba de combustível de foguetes. Marcapassos são monitorados graças à mesma tecnologia utilizada em satélites” (Reynold, 2004), e muitos outros, como a liga de Nitinol usada hoje em aparelhos ortodônticos, a própria asa-delta já foi um protótipo de guia para reentrada de espaçonaves na terra, além de outras inúmeras inovações como o isopor, o forno microondas, o relógio digital e o computador.

Os vínculos entre conhecimento, tecnologia e poder estão intimamente ligados em todas as épocas e influenciam diretamente as relações sociais. Isso implica mudanças gigantescas nas cadeias econômicas, no mercado de trabalho e na divulgação desses conhecimentos, alterando o modo como a educação é realizada. Vale lembrar, que a tecnologia, assim como a educação, não se desenvolvem de forma neutra, mas possuem influências que atendem os mais diversos interesses políticos, econômicos e sociais (Sahb; Almeida, 2016).

Desta forma, a escola também exerce seu poder sobre o conhecimento e o uso das tecnologias que podem mediar a relação de ensino e aprendizagem entre aluno e professor, logo, as escolas representam na sociedade moderna a oportunidade de educação em um momento em que a tecnologia altera de forma cada vez mais rápida nosso modo e qualidade de vida (Kenski, 2009).

Essa interação entre tecnologia e conhecimento, implica fortemente no modo como as informações são transmitidas. Para entender essa relação, é de interesse de vários autores desmistificar e definir o conceito de tecnologia o qual causa confusão entre leitores e, até

mesmo, uma profusão entre os termos de “tecnologia” e “técnica” usados na literatura (Lemos, 2013). De modo que,

Hoje compreendemos por tecnologia os objetos técnicos, as máquinas e seus respectivos processos de fabricação. Do mesmo modo, utilizamos o termo técnica para abranger áreas tão díspares como a dança, a economia, as atividades esportivas, ou mesmo objetos, instrumentos e máquinas. A confusão entre os termos é impressionante (Lemos, 2013, p. 26).

Por sua vez, Kenski (2009, p. 24) define tecnologia de uma forma mais pragmática e conceitual como sendo “o conjunto de conhecimentos e princípios científicos que se aplicam à construção e a utilização de um equipamento em um determinado tipo de atividade”, e também define técnica como, “as maneiras, jeitos ou habilidades especiais de lidar com cada tipo de tecnologia para executar ou fazer algo”. Uma definição importante para entender como a tecnologia influencia na educação e no modo com que compartilhamos as informações e conhecimentos.

Além disso, para possibilitar a comunicação, o homem criou um tipo especial de inteligência, a “tecnologia da inteligência”, o que é imaterial e não existe como objeto, mas sim como uma forma de linguagem, e os processos da indústria das telecomunicações e da informática vêm alterando o modo como produzimos esse tipo de tecnologia com novas maneiras de pensar, de conviver e de ver o mundo (Lévy, 1993).

As tecnologias que constituem a divulgação de informação baseados na linguagem oral, escrita e visual atribuímos o termo Tecnologia da Informação e Comunicação (TIC) (Kenski, 2009). No entanto, com o advento da propagação de dispositivos digitais e do uso da internet, há pesquisadores que utilizam o termo Tecnologias Digitais da Informação e Comunicação (TDIC) para se referir a tecnologias digitais conectadas à internet, como Kenski (2009), ou que refere-se às TDIC a partir da convergência de várias tecnologias digitais, ou seja, qualquer equipamento eletrônico, aplicativos, softwares que possam se conectar à internet, possibilitando o aumento de comunicação de seu usuário, como Valente (2014).

Com todos esses elementos, podemos inferir que o desenvolvimento tecnológico ocasionou uma nova forma do ser humano se comunicar, muito mais rápida do que um sinal de fumaça ou pelo envio de cartas. Onde a informação pode ser armazenada em *pendrives* menores do que um abridor de lata ou um chaveiro, e até mesmo dispor de informações que foram armazenadas a quilômetros de distância. Mas mesmo assim, podem ser armazenadas e

disseminadas pela internet em uma fração de segundos, o que potencializa e amplia a rede de comunicação humana.

Sendo então preciso aprender uma nova linguagem para se conversar com a tecnologia, ou, mais precisamente, com os computadores. Isso aumenta a abrangência do ensino de computação, passando a ser considerada uma ciência básica, ou seja, um subconjunto de competências e habilidades básicas que precisam ser desenvolvidas pelos estudantes da educação básica. Além disso, segundo Wing (2006), essas competências e habilidades podem ser denominado como “*Computational thinking*”, compreendendo assim os conhecimentos e domínios relacionadas à abstração e decomposição de problemas de forma a permitir sua resolução usando recursos computacionais e estratégias algorítmicas.

O pensamento computacional é uma habilidade fundamental para todos, não apenas para os cientistas da computação, pois ele refina as habilidades de leitura, a escrita e a aritmética, assim como a capacidade analítica de cada criança (Wing, 2006). Desta forma, para a pessoa “conversar” com o computador, é necessário utilizar uma linguagem de programação pela qual serão realizados comandos ou instruções algorítmicas para o computador executar uma ou várias tarefas, desenvolvendo assim na pessoa uma habilidade de descrição das ideias perante um linguagem formal e precisa, a construção de respostas específicas mediante a execução de um programa, a habilidade de refletir sobre o que foi executado e se o resultado foi satisfatório, assim como aprender a depurar a ideia e descrição inicial do seu algoritmo caso ele não tenha atingido o objetivo almejado (Valente, 1997).

E para aprender a falar a língua dos computadores, atualmente a linguagem de programação Python é indicada como uma das linguagem de programação mais popular em todo mundo, analisando a frequência com que tutorias de linguagem são pesquisados no Google (Github, 2023). E, segundo o Stack Overflow (2023) (site comunitário de perguntas e respostas para profissionais e entusiastas na área de programação de computadores), o Python ultrapassou o *Structured Query Language* (Linguagem de Consulta Estruturada - SQL) como a terceira linguagem mais usada no mundo, mas ficando em primeiro lugar para aqueles que não são desenvolvedores profissionais ou estão aprendendo a codificar.

Segundo Cruz (2022),

No que se refere à opção pela linguagem python, destaca-se a sua sintaxe, em geral, simples e robusta, sendo apropriada para aqueles com pouca ou nenhuma experiência em programação, o que permite compreender com maior facilidade

conceitos básicos da construção de algoritmos, como sequência, condição e repetição (Cruz, 2022, p. 207).

A linguagem de programação Python é uma das linguagens mais populares e amplamente utilizadas no mundo da programação. Ela foi criada por Guido van Rossum, um programador holandês, e a primeira versão, Python 0.9.0, foi lançada em fevereiro de 1991. A inspiração para criar o Python veio de Guido van Rossum enquanto trabalhava no Centro de Matemática e Informática (CWI) na Holanda. Ele queria criar uma linguagem que fosse fácil de ler, elegante, interpretável e que enfatizasse a produtividade do programador (Van Rossum, 1995). Logo, Python foi projetada para ser uma linguagem de alto nível, o que significa que os programas escritos nela são legíveis e próximos da linguagem humana, uma característica importante da linguagem a qual reflete na grande quantidade de alunos iniciantes que optam por utilizá-la.

Curiosamente, o nome "Python" não se refere à serpente, mas sim aos comediantes britânicos Monty Python, dos quais Guido van Rossum era fã (Lutz, 2007). Ao longo dos anos, Python evoluiu e ganhou popularidade rapidamente. Sua simplicidade e versatilidade a tornaram uma escolha ideal para uma variedade de aplicações, desde desenvolvimento web até análise de dados e automação. E para manter o desenvolvimento desta linguagem, a Python Software Foundation (PSF), uma organização sem fins lucrativos, foi fundada para apoiar seu desenvolvimento contínuo (PSF, 2023).

Além disso, segundo o GitHub (2023), um dos maiores sites de desenvolvedores computacionais, a linguagem de programação Python é a mais popularmente difundida, ultrapassando até mesmo a linguagem de programação Java, que liderava o ranking há mais de uma década (GitHub, 2023). Sua popularidade está aliada a extensa gama de tutoriais e vídeos conteúdos disseminados na internet que, por sua vez, aumenta a oportunidade de aprendizado na área de desenvolvimento de programas computacionais baseada na linguagem Python.

Ademais, Python é uma linguagem de código aberto, o que significa que qualquer pessoa pode contribuir para seu desenvolvimento. Isso levou ao surgimento de uma vasta biblioteca de módulos e *frameworks* que tornam Python ainda mais poderoso e útil em uma ampla gama de campos. Nos últimos anos, ela tem sido amplamente adotada na ciência de dados, computação científica, aprendizado de máquina, automação de tarefas, desenvolvimento web, entre outros. Sua popularidade crescente a torna uma das linguagens mais relevantes para aprender no mundo da programação (Menezes, 2010).

Em resumo, a linguagem de programação Python tem uma história fascinante de crescimento e evolução desde sua criação em 1991, e continua a desempenhar um papel fundamental na computação moderna. E na educação, Valente (1997) já indicava que o uso de linguagens de programação, como BASIC e Pascal, poderiam auxiliar no processo de ensino e aprendizado, no qual o aluno é um ser ativo que passa a informação para o computador como uma linguagem de texto logicamente executável, promovendo a resolução de problemas e tarefas.

Desta forma, escolhemos utilizar nesta pesquisa a linguagem de programação Python como recurso para auxiliar a estratégia de ensino e aprendizagem de Física, uma vez que a mesma é uma linguagem de alto nível, ou seja, muito próxima a linguagem de comunicação humana, com uma sintaxe projetada para ser muito mais legível do que as linguagens de programação mais tradicionais. Outro motivo, comentado anteriormente, é sua extensa gama de biblioteca de suporte e sua *Integrated Development Environment* (Ambiente de Desenvolvimento Integrado - IDE), criadas especificamente para a linguagem Python, o que tende a gerar um ambiente de trabalho menos complicado (Lutz, 2007).

Porém, mesmo com todo esse potencial, ainda existem muito poucos trabalhos que examinam seu uso no ensino de ciência, como apresentado por Cruz *et al.* (2022). Os autores realizaram um levantamento de trabalhos que abordam o uso e/ou a criação de simuladores de fenômenos físicos, considerando os artigos publicados nos último dez anos no Simpósio Nacional de Ensino de Física (SNEF), Encontro de Pesquisa e Ensino de Física (EPEF), na Revista Brasileira de Ensino de Física (RBEF) e no Caderno Brasileiro de Ensino de Física (CBEF).

A partir de uma análise, é possível afirmar que o número de trabalhos que abordam a construção de simuladores com o Python no ensino da Física é relativamente pequeno se comparado às vantagens proporcionadas por eles. Outrossim, nenhum trabalho discute movimentos da cinemática, o que demonstra a necessidade de investigar e explorar mais esses recursos. Pontuamos ainda que, mesmo que o levantamento realizado considerou apenas dois periódicos e duas revistas, é válido destacar a relevância desses locais de investigação para a área de Ensino de Física, pois recebem um grande número de trabalhos a nível nacional (Cruz *et al.*, 2019, p. 214).

A partir deste levantamento os autores encontraram 7 trabalhos que abordaram a aplicação do uso da linguagem Python, porém, em nenhum dos trabalhos foi apresentado de forma clara o seu uso como recurso para os alunos desenvolverem suas próprias simulações, mas sim uma aplicação direta de um recurso pronto. O que diferencia a abordagem desta

presente pesquisa, uma vez que se busca entender como a própria linguagem Python pode ajudar o ensino e a aprendizagem, tornando-a um recurso de inúmeras facilidades.

E, como abordado por Rodrigues e Machado (2022), ao realizar uma pesquisa bibliográfica em quatro bases de dados, sendo duas internacionais: Repositório Científico de Acesso Aberto de Portugal (RCAAP) e Scientific Electronic Library Online (SciELO), e duas nacionais: Portal de Periódicos da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e Catálogo de Teses e Dissertações da CAPES, no recorte temporal de 5 anos entre 2014 e 2018, sob a busca das seguintes palavras chaves: alfabetização científica e tecnologia. Com isso, os autores puderam constatar que os estudos contemporâneos sobre as possibilidades de articulação entre alfabetização científica (AC) e tecnologias digitais de informação e comunicação (TDIC) na educação em ciências, se dedicam à implementação dos processos de AC e não na articulação entre eles.

Enquanto isso, apenas uma pequena parte dos artigos da pesquisa de Rodrigues e Machado (2022) apresenta o estudo de potencialidades das TDIC para alcançar a AC dos estudantes, além disso, a mesma pesquisa indica que a grande parte de recursos tecnológicos são utilizados para otimizar as atividades em AC. Por fim, essa revisão salienta que ainda há poucos trabalhos associando de forma clara e intencional AC e TDIC o que aponta uma lacuna e pode contribuir para que se desenvolvam novas pesquisas que associam as TDIC ao processo de construção da AC (Rodrigues; Machado, 2022).

Além disso, recentemente Yamaguti e Carvalho Neto (2023) fizeram um levantamento bibliográfico em revistas sobre ensino de Física latino-americanas, categorizadas pela CAPES segundo o quadriênio de 2013 a 2016 com qualis A1, A2, B1 e B2, delimitado ao espaço temporal entre janeiro/2018 e julho/2022, a fim de entender o que se tem pesquisado sobre o tema “O uso da linguagem de programação no Ensino de Física”. Nesta pesquisa foram identificados 25 artigos por meio dos seguintes buscadores em português e em espanhol: programação, TIC e informática. Onde identificou-se que as frentes mais recorrentes de pesquisa são: criação de aparatos físicos ou virtuais, software ou aplicativo como ferramenta para o ensino e aprendizagem e outras tecnologias na educação. E destaca-se o uso dessas tecnologias em experimentos virtuais em disciplinas de laboratórios didáticos durante a pandemia como apoio para a realização das aulas remotas. Enquanto isso, o mesmo trabalho ainda mostra que existe apenas um artigo que apresenta o uso da linguagem de programação como recurso que permite potencializar a racionalização dos estudantes por

meio da lógica de programação para resolução de problemas ou conceituação de um fenômeno físico.

Desta forma, como destacado no trabalho de Machado e Rodrigues (2022), e nas outras pesquisas (Yamaguti; Carvalho Neto, 2023; Cruz; *et al.*, 2022), a bibliografia aponta uma carência em investigações pedagógicas que articulem o uso de TDIC visando potencializar o desenvolvimento do ensino e aprendizagem em Alfabetização Científica e Tecnológica (ACT). Assim como Cruz *et al.* (2022) também relata a incrível falta de trabalhos desenvolvidos com o auxílio da linguagem Python no ensino de Física mediante suas grandes potencialidades.

Logo, identifica-se uma lacuna na literatura expressa pela falta de pesquisas, investigações e trabalhos que visam utilizar a linguagem de programação como uma das ferramentas e ou recursos oriundos da TDIC, a fim de proporcionar o ensino e aprendizado nas ciências, articulando-se com metodologias de ensino e estratégias pedagógicas. Visto isso, é justificável pesquisar como as linguagens de programação podem ser usadas como recursos potencializadores para o ensino e aprendizado em ciências.

1.2 Alfabetização Científica e Tecnológica no ensino de ciências

Com o crescimento das TDIC e as facilidades de disseminação de informação, sua utilização incorreta também pode ocasionar disparidades e propiciar:

A difusão de informações falsas e sem fundamento científico por meio de plataformas cujos algoritmos podem gerar enviesamentos e ‘bolhas’ virtuais de informação não científica. Nesse contexto, reiteramos a necessidade de associação entre o acesso e a democratização das TDIC e o desenvolvimento da alfabetização científica na escola considerando o contexto da cultura digital (Rodrigues; Machado, 2023, p. 14-15).

Por conta disso, é necessário que as TDIC estejam aliadas à alfabetização, que é um processo de conhecimento do mundo a sua volta que permite a conexão dos saberes do mundo em que as pessoas vivem e a palavra, ou seja, a comunicação e informação. Logo,

A alfabetização é mais do que o simples domínio psicológico e mecânico das técnicas de escrever e de ler. É o domínio dessas técnicas, em termos conscientes. É entender o que se lê e escrever o que se entende. É comunicar-se graficamente. É uma incorporação. Implica, não uma memorização visual e mecânica de sentenças, de palavras, de sílabas, desgarradas de um universo existencial – coisas mortas ou semimortas – mas numa atitude de criação e recriação, implica numa autoformação

de que possa resultar uma postura interferente do homem sobre seu contexto (Freire, 2005, p. 118).

Desta forma, em sintonia com a abordagem de Freire (2005), a Alfabetização Científica (AC) pode ser descrita em três eixos; (i) compreensão básica de termos, conhecimentos e conceitos científicos fundamentais; (ii) compreensão da natureza das ciências e dos fatores éticos e políticos que circundam sua prática; e (iii) entendimento das relações existentes entre ciência, tecnologia, sociedade e meio-ambiente (Sasseron; Carvalho, 2011).

Nesta direção, as TDIC apresentam recursos importantes e capazes de criar um contexto de aprendizagem de competências múltiplas associadas à AC, como as comunicativas, tecnológicas e científicas, bem como o confronto dos discursos anti-científicos e negacionistas (Rodrigues; Machado, 2023).

Desta forma, podemos observar uma intrínseca relação entre a Alfabetização Científica (AC) e o desenvolvimento tecnológico, o que culmina na importância do indivíduo ser alfabetizado científica e tecnologicamente, em vista da necessidade de poder discernir as compreensões científicas em meio à disseminação das informações pela tecnologias digitais da informação e comunicação, tornando necessário o entendimento do uso racional e crítico das tecnologias, o que não é recente, mas os estudos vêm crescendo e ampliando-se nos últimos anos.

Milaré (2008), ao resgatar o conceito de AC e sua relação com a tecnologia, aponta alguns diferentes significados encontrados na literatura:

[...] a Alfabetização Tecnológica também possui muitos significados. Há aqueles que enfocam a criação e a construção de artefatos tecnológicos pelos alunos durante as aulas. Outros relacionam a Alfabetização com o estudo dos grandes desenvolvimentos tecnológicos da história da humanidade. A Tecnologia também é utilizada no ensino como uma forma de despertar o interesse dos alunos. Cada uma destas estratégias para a inclusão da Tecnologia no ensino possui sua validade considerando quais foram os objetivos traçados inicialmente e a formação do professor que irá desenvolvê-los em sala de aula (Milaré, 2008, p. 109).

Assim, a autora defende que é necessário entender a Alfabetização Tecnológica (AT) não apenas como uma simples aplicação dos conceitos de tecnologia ou uma instrumentalização por parte dos estudantes, mas sim uma busca pela compreensão dos mecanismos técnicos e autonomia dos estudantes perante as tecnologias que vêm sendo desenvolvidas.

O Ensino de Ciências pode auxiliar os estudantes a posicionarem-se diante da Ciência e da Tecnologia, considerando suas influências na vida dos cidadãos. Isso justifica a inclusão de aspectos tecnológicos no ensino e envolve capacidades como modelizar, abstrair, compreender códigos, antecipar e analisar situações, solucionar problemas e articular teoria e prática (Milaré, 2008, p. 110)

Desse modo, ao entendermos que a Ciência e Tecnologia são produtos sociais, desenvolvidas dentro de um contexto político e econômico, percebe-se a intrínseca relação entre AC e AT, que juntas buscam alfabetizar científica e tecnologicamente os estudantes, com o objetivo de que esses possam participar de debates, negociar e dialogar com outras pessoas, lidar com uma variedade de situações de forma lógica e racional, e entender como aplicar os conhecimentos na prática (Milaré, 2008)

O termo AC, segundo alguns autores, é citado desde o século XVI, enquanto as relações entre ciência e tecnologia que é o foco desta pesquisa surgem entre os séculos XIX e XX (Milaré; Richetti, 2021). Porém, a ACT ganha força na década de 1970, juntamente com o movimento Ciência, Tecnologia e Sociedade (CTS), quando as crises ambientais e sociais causadas devido ao grande desenvolvimento científico e tecnológico começam a ser incorporadas ao ensino de ciências.

Desde meados do século XX, tem se buscado uma revolução no ensino de ciências que estava em crise, principalmente após o final da Segunda Guerra Mundial e com a Guerra Fria, quando os países americanos começaram a buscar formas de ensinar ciências mais efetivas à população (Gaspar, 2014). Neste momento, a ACT surge como uma resposta a essa crise, além de outras metodologias, estratégias e recursos didáticos. Apesar de passados muitos anos, a ACT continua sendo uma metodologia muito utilizada para o ensino de ciências, com diversas pesquisas sendo realizadas sobre esse tema até hoje.

Principalmente por partir da necessidade de superar um ensino propedêutico, tradicional linear e reducionista, e estimular de forma didática-pedagógica um ensino que quebre os paradigmas da forma tradicional de se ensinar ciências. Logo, busca-se, a partir da ACT, a possibilidade de explorar uma educação temática e interdisciplinar (Auler, 2003).

Segundo Siemsen (2019), a ACT possui um potencial didático muito grande para o ensino de ciências, principalmente por ter um caráter interdisciplinar. Como exemplo, a autora em sua dissertação de mestrado, apresenta uma sequência didática com a proposta temática de ensino em astronomia, na qual verifica-se que existe um grande engajamento entre os alunos, promovendo debates e discussões a respeito dos temas escolhidos, envolvendo diversos outros

contextos históricos, filosóficos, geográficos, matemáticos entre outros, o que por sua vez possibilita um aprendizado científico e tecnológico significante.

Um outro ponto importante a fim de trazer clareza a perspectiva desta pesquisa, é o marco histórico muito próximo entre o surgimento da ACT e do movimento CTS, que são frequentemente tratados como sinônimos, todavia representam conceitos distintos. A ACT almeja um ensino de ciências que promova a educação científica para todos os cidadãos, no qual seja clara a relação entre Ciência e Tecnologia e suas implicações na sociedade. Desse modo,

Alfabetizar científica e tecnologicamente, implica em um processo de conscientização no uso dos conhecimentos científicos e tecnológicos, associados a outros tipos de conhecimentos e aspectos, que pode transformar as pessoas, e consequentemente, interferir nos contextos sociais, políticos e econômicos (Milaré; Richetti, p. 37, 2021).

Em seu livro, Fourez (2005) cita as habilidades consideradas necessárias para uma pessoa estar alfabetizada científica e tecnologicamente segundo os critérios propostos pela Associação de Professores de Ciências dos Estados Unidos (NSTA), como:

Utilizar os conceitos científicos e integrar valores e saberes para tomar decisões responsáveis no dia a dia; Compreender que a sociedade exerce um controle sobre as ciências e as tecnologias, e assumir que as ciências e as tecnologias imprimem seu selo na sociedade; Compreender que a sociedade exerce um controle sobre a ciência e as tecnologias por meio dos subsídios que lhes foram concedidos; Reconhecer tanto os limites como a utilidade das ciências e das tecnologias no progresso do bem estar humano; Conhecer os principais conceitos, hipóteses e teorias científicas e ser capaz de aplicá-los; Apreciar a ciência e as tecnologias pela estimulação intelectual que suscitam; Compreender que a produção do saberes científicos dependem ao mesmo de processos de pesquisa e conceitos teóricos; Saber reconhecer a diferença entre resultados científicos e opiniões pessoais; Reconhecer a origem da ciência e compreender que o saber científico é provisório de acordo com o grau de acúmulo de resultados; Compreender a aplicação das tecnologias e as decisões envolvidas em sua utilização; Possuir suficientes saberes e experiência para apreciar o valor da investigação e dos desenvolvimentos tecnológico; Extrair de sua formação científica uma visão do mundo mais rica e interessante; Conhecer as fontes válidas de informação científica e tecnológica e recorrer à elas quando tem que tomar decisões; Ter uma certa compreensão da maneira como a ciências e as tecnologias foram produzidas na história (Fourez, 2005, p. 25-36, tradução nossa).

Logo, podemos interpretar que a ACT capacita os indivíduos a compreender e aplicar os conceitos científicos e tecnológicos no cotidiano, ao diferenciar resultados científicos de opiniões pessoais. Em contraste, a abordagem CTS examina as interações entre ciência, tecnologia e sociedade, destacando o controle social sobre ciência e tecnologia, suas influências recíprocas, os limites e a utilidade das ciências e tecnologias no bem-estar

humano, e a importância histórica e investigativa dessas áreas. Enquanto a ACT foca na capacitação científica, CTS enfatiza a relação crítica entre a sociedade e as inovações científicas e tecnológicas.

Além disso, existem diferentes possibilidades de abordagens do movimento CTS, de acordo com a ênfase que é dada aos conhecimentos científicos em relação aos conceitos sociais, ou seja, os conhecimentos científicos podem ficar em segundo plano durante o ensino, enquanto numa abordagem da ACT isso não acontece, uma vez que os conhecimentos científicos e tecnológicos são imprescindíveis (Milaré; Richetti, 2021).

Portanto, quando falamos sobre ACT, devemos levar em conta que:

A ACT diz respeito mais do que ao conhecimento, ou igualmente, à aplicação de resultados científicos e tecnológicos. Tem como objetivo promover a consciência da ciência e da tecnologia como fenômenos da sociedade e da história. A produção do conhecimento científico não é um processo puramente teórico, nem mesmo teórico-experimental: põe em jogo seres humanos específicos, situados no interior de instituições, numa cultura e numa história. Assim, os 'métodos científicos' implicam, para além dos conceitos teóricos, o desenvolvimento de equipas de trabalho, a mobilização de recursos humanos e económicos, de capital humano e social, bem como a gestão de tudo isto - muitas vezes no meio de numerosos conflitos. (Fourez, 2005, p. 30-31, tradução nossa).

Logo, para atingir o objetivo destacado pela ACT, a computação, assim como as TDIC, têm interferido na forma como a sociedade se desenvolve e lida com o mundo, causando mudanças originárias da evolução tecnológica que repercutem no modo de vida do ser humano (Cassal, 2020).

1.3 O ensino de Física a partir dos Três Momentos Pedagógicos

Para chegar a um denominador comum entre o uso da linguagem de programação como recurso potencializador do ensino de Física, é necessária uma conexão metodológica que una esses dois objetos a partir de uma prática pedagógica que faça sentido para o indivíduo, desta forma, propomos essa união por meio da metodologia de ensino dos Três Momentos Pedagógicos (3MP) (Muenchen; Delizoicov, 2014), que tem origem na década de oitenta.

Entre os anos de 1985 e 1988 a Coordenadoria para Articulação com Estados e Municípios da Secretaria do Ensino de 2º Grau do MEC se propôs a desenvolver o “Projeto diretrizes gerais para o ensino de 2º grau: núcleo comum e habilitação magistério”

(Muenchen; Delizoicov, 2014), que trouxe subsídios para criação dos livros “Física” (Delizoicov; Angotti, 1990a) e “Metodologia do Ensino de Ciências” (Delizoicov; Angotti, 1990b), que contribuiu para a disseminação dos 3MP, o qual foi construído sobre uma abordagem

Vinculada a uma temática central, que será devidamente explicitada ao professor do 2º grau. Uma vez aceita a proposta, o professor poderá seguir as indicações, roteiros e instruções para o desenvolvimento de seu curso, sem contudo deixar de introduzir elementos que interessem ao seu grupo de alunos, determinados pelas condições locais e regionais onde esteja atuando. Queremos dizer que não se trata de um subsídio excessivamente diretivo, porém fundamentalmente um material didático que propõe diretrizes e alternativas, que simultaneamente pede e estimula a participação do professor quanto aos caminhos e possibilidades de se montar e prosseguir no curso. Esta proposta não é determinada por orientações de livros didáticos, mas ela se utilizará deles como instrumentos que contribuem significativamente para que o professor desenvolva seu trabalho (Delizoicov; Angotti, 1990a, p. 6).

Com isso, os 3MP foram caracterizados como metodologia de ensino por possuírem três etapas bem definidas pelas suas funções pedagógicas, sendo elas: (1) problematização inicial, (2) organização do conhecimento e (3) aplicação do conhecimento.

O primeiro momento apresenta situações reais que os alunos conhecem e estão envolvidos, mas que requerem uma introdução dos conhecimentos científicos para serem interpretadas. Logo,

Deseja-se aguçar posições contraditórias e localizar as possíveis limitações do conhecimento que vem sendo expressado pelos alunos, quando é cotejado, implicitamente, pelo professor, com o conhecimento científico que já foi selecionado para ser abordado. Em síntese, a finalidade deste momento é propiciar um distanciamento crítico do aluno ao se defrontar com as interpretações das situações propostas para discussão (Angotti, 2015, p. 16).

E também culmina com que o aluno sinta necessidade de buscar novos conhecimentos para resolver esse problemas (Angotti, 2015). Desta forma, esse momento oferece a oportunidade do professor incluir diversas problematizações, assim como o tema de tecnologia da informação que está disseminada em nossa sociedade, fazendo-se presente no dia a dia dos alunos, por meio dos aparelhos eletrônicos e mídias digitais.

O segundo momento corresponde à etapa em que o professor seleciona os conhecimentos necessários para a resolução das problemáticas expostas inicialmente. Neste momento, o professor orienta de forma sistemática os alunos, fundamentando os conhecimentos científicos por meio do desenvolvimento das mais variadas formas de atividades, recursos ou estratégias (Angotti, 2015). Com essa gama de diversidade de

atividades, o professor pode propor a relação dos conhecimentos científicos com atividades interdisciplinares com a linguagem de programação e o uso de TDIC. Por exemplo, quando é expressa a função horária dos espaços em um Movimento Retilíneo e Uniformemente Variado (MRUV), é possível relacionar conceitos básicos como as variáveis da função horária com as variáveis em linguagem de programação, a criação de funções e sua execução.

Por último, o terceiro momento

Destina-se, sobretudo, a abordar sistematicamente o conhecimento que vem sendo incorporado pelo aluno para analisar e interpretar tanto as situações iniciais que determinaram seu estudo como outras que, embora não estejam diretamente ligadas ao motivo inicial, podem ser compreendidas pelo mesmo conhecimento (Angotti, 2015, p.17).

Desta forma, a meta do ensino e aprendizagem das ciências tem seu ápice no potencial explicativo e conscientizador das teorias científicas que precisa ser explorado para melhor entender os conhecimentos científicos de forma significativa (Angotti, 2015). Com isso, surge a oportunidade de unir a utilização das tecnologias da informação para contribuir para o processo de aprendizado apresentando possibilidades e recursos computacionais no ensino de Física, proporcionando a interação entre o conteúdo e o aluno, a fim de possibilitar meios para a investigação e compreensão de fenômenos físicos (Cardoso; Dickman, 2012).

1.4 Conteúdo de Física

Essa pesquisa aborda com os alunos os fenômenos de cinemática no contexto de ensino e aprendizado, logo, faz-se necessário a apresentação do conteúdo utilizado na sequência didática dessa pesquisa para conhecimento do leitor e também para referenciar os materiais que foram utilizados no capítulo seguinte.

Segundo Halliday *et al.* (2012) um dos objetivos da física é estudar o movimentos dos corpos, ou seja, a rapidez com que um objeto se move, a distância que ele percorre em um determinado período de tempo, a variação de sua velocidade e até mesmo a composição destes movimentos em mais de uma dimensão. Para isso, esse subtópico é dividido na explanação do fenômeno do movimento retilíneo e uniforme (MRU), movimento retilíneo e uniformemente variado (MRUV) e movimento balístico.

1.4.1 Movimento retilíneo e uniforme

Para compreender o MRU, precisamos entender a posição que a partícula se encontra neste fenômeno, para assim determinar seu deslocamento, logo, a posição x de uma partícula em um eixo x indica a distância da partícula em relação à origem, ou ponto zero do eixo. Essa posição pode ser expressa como um valor positivo ou negativo, dependendo da localização da partícula em relação à origem. Isto é, o sentido positivo do eixo é definido como o sentido em que os valores aumentam ao longo do eixo a partir da origem, e o sentido negativo é o oposto (Halliday, 2012).

Figura 1 - Sentido do deslocamento de uma partícula.

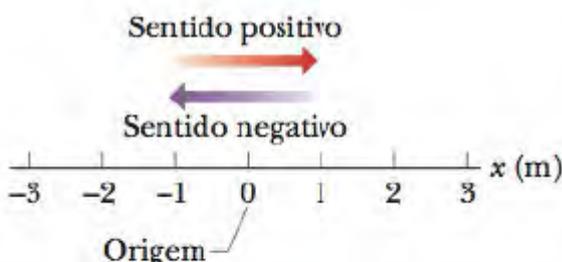


Figura 2-1 A posição é assinalada em um eixo marcado em unidades de comprimento (metros, por exemplo), que se estende indefinidamente nos dois sentidos. O nome do eixo, x , por exemplo, aparece sempre no lado positivo do eixo em relação à origem.

Fonte: Halliday *et al.* (2012, p. 14).

Logo, o deslocamento de uma partícula no eixo x é representado pela expressão,

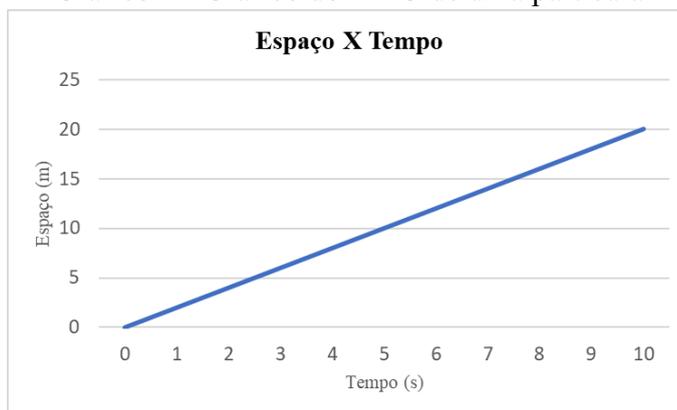
$$\Delta x = x_2 - x_1. \quad (1)$$

Sabendo que a partícula se desloca no espaço durante um intervalo de tempo, podemos determinar sua velocidade média como:

$$v_{med} = \frac{\Delta x}{\Delta t} = \frac{x_2 - x_1}{t_2 - t_1}. \quad (2)$$

Desta forma, podemos observar o exemplo do deslocamento de uma partícula que possui velocidade igual a 2 m/s no gráfico 1.

Gráfico 1 - Gráfico do MRU de uma partícula



Fonte: Elaborado pelo autor.

1.4.2 Movimento retilíneo e uniformemente variado

Além do caso MRU é comum estudarmos na Física fenômenos que possuem uma variação constante de sua velocidade, como por exemplo o movimento de queda livre, onde uma partícula aumenta sua velocidade constantemente conforme a gravidade da Terra a puxa para o centro do planeta. Para esses tipos de movimento retilíneo onde a variação da velocidade é praticamente constante, damos o nome de movimento retilíneo e uniformemente variado ou MRUV (Halliday, 2012).

Neste tipo de fenômeno, a taxa com que a velocidade varia durante um intervalo de tempo é denominada aceleração média, e podemos expressá-la da seguinte forma:

$$a = a_{med} = \frac{\Delta v}{\Delta t} = \frac{v - v_f}{t - 0} \quad (3)$$

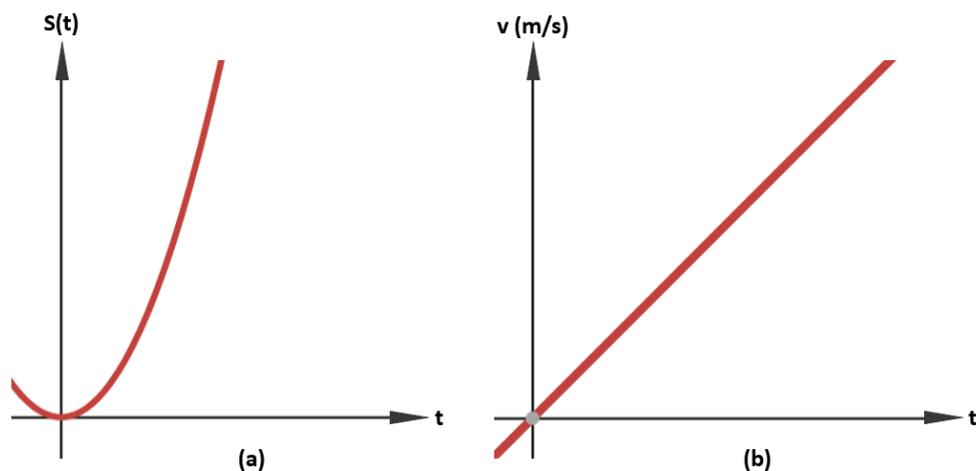
No caso especial do exemplo de queda livre quando a partícula parte do repouso, as equações podem ser expressas para descrever seu movimento, mas é preciso fazer três mudanças na notação: (1) o movimento deve ser descrito em relação a um eixo vertical y , e com o sentido positivo verticalmente para cima; e (2) a aceleração deve ser substituída por $-g$, onde o valor do módulo da aceleração da gravidade próxima a superfície da Terra é de $g = 9,8 \text{ m/s}^2$; e (3) como a partícula parte do repouso, sua velocidade inicial é considerada nula (Halliday, 2012). Sendo assim, as demais equações referentes ao movimento da partícula ficam da seguinte forma:

$$v = gt \quad (4)$$

$$\Delta y = \frac{1}{2}gt^2 \quad (5)$$

O fenômeno de queda livre pode ser exemplificado graficamente pelas representações do gráfico X.Xa em que é apresentado o deslocamento da partícula em relação ao tempo e no gráfico X.Xb, onde pode ser observado o aumento constante da velocidade em relação ao tempo.

Gráfico 2 - Deslocamento de uma partícula em MRUV



Fonte: Elaborado pelo autor.

1.4.3 Movimento balístico

Tanto o conceito de aceleração, como o de velocidade não se aplicam apenas em casos retilíneos, ou em um único eixo. Esses fenômenos se estendem a movimentos bidimensionais como o caso do lançamento de uma bola de basquete (onde desprezamos a resistência do ar), em que no eixo vertical existe influência da gravidade em sua trajetória, e no eixo horizontal o movimento é retilíneo e uniforme.

Para isso podemos utilizar o conceito de vetor de forma genérica para simbolizar o deslocamento e a velocidade da partícula

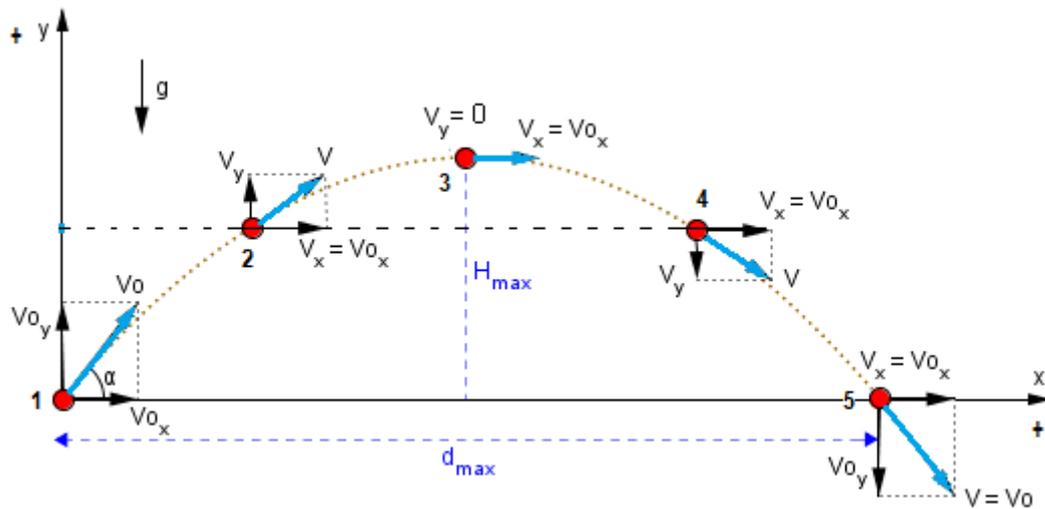
$$\vec{r} = x\hat{i} + y\hat{j} \quad (6)$$

$$\Delta\vec{r}(t) = \vec{r}(t) - \vec{r}_0 \quad (7)$$

$$\vec{v}_{med} = \frac{\Delta\vec{r}}{\Delta t} \quad (8)$$

A partir das equações vetoriais, podemos decompor o movimento balístico tendo como base que sua velocidade inicial v varia em toda trajetória, onde a aceleração horizontal da partícula é zero e a aceleração vertical de queda livre é $-g$. Portanto, é possível expressar a decomposição de v_0 pelo seu ângulo, conforme a representação da figura 3.

Figura 2 - Decomposição do lançamento oblíquo.



Fonte: Manetta, 2013, np.

Desta forma chegamos nas seguintes equações do movimento da partícula ao longo do eixo horizontal x e do eixo vertical y .

$$\Delta x = (v_0 \cos \theta_0) t \quad (9)$$

$$\Delta y = (v_0 \sin \theta_0) t - \frac{1}{2} g t^2 \quad (10)$$

$$v_x = v_{0x} = v_0 \cos \theta \quad (11)$$

$$v_y = v_0 \sin \theta_0 - g t \quad (12)$$

1.5 Linguagem de programação Python

Como já citado na fundamentação teórica, a linguagem de programação escolhida para essa pesquisa é a linguagem Python. E, a fim de situar o leitor sobre os conteúdos que serão ensinados sobre Python, como também, referenciar o conteúdo utilizado na sequência didática FIS_PY, esse subtópico é dedicado em abordar os principais conceitos utilizados no contexto

de ensino e aprendizado desta pesquisa divididos em seis partes: variáveis, operações matemática e relacionais, funções em Python, importação de bibliotecas e módulos, laços ou *loops* e teste *if* e *else*.

1.5.1 Variáveis

As variáveis em Python servem para armazenar dados que podem ser utilizados e manipulados ao longo da execução de um programa. Elas desempenham um papel fundamental na escrita de um código, permitindo que os programadores gerenciem informações de forma dinâmica e eficiente.

Lutz (2007) aborda a ideia que as variáveis em Python são como "etiquetas" que referenciam objetos na memória do computador, ou seja, no código é possível atribuir um tipo valor a uma variável. Além disso, vale ressaltar que em Python tudo pode ser considerado um objeto, e existem diversos tipos de objetos, dos quais podemos destacar alguns no quadro 1.

Quadro 1 - Exemplo de tipos de objetos em Python

Tipos de objetos (tipo em python)	Exemplos de literais / criação
Número inteiros (int)	3, 7, 1245, 18
Números flutuantes (float)	3.14, 7.45286, 2.71828
String (str)	“Olá, mundo!”, “erro”, “Física”
Listas (list)	[1, [2, “três”], 4]

Fonte: Elaborado pelo autor.

1.5.2 Operações matemáticas e relacionais

As operações matemáticas em Python são utilizadas por uma variedade de razões em programas, entre elas temos: cálculos numéricos, manipulação de dados, controle de fluxo, modelagem matemática, visualização de dados entre outras razões (Lutz, 2007). Para isso, destaca-se as operações matemáticas fundamentais no quadro 2.

Quadro 2 - Exemplo de operações matemáticas feitas em Python

Operação	Operador	Explicação
Adição	+	Utilizada para somar dois valores.
Subtração	-	Utilizada para subtrair um valor de outro.
Multiplicação	*	Utilizada para multiplicar dois valores
Divisão	/	Utilizada para dividir um valor por outro. A divisão retorna um resultado de ponto flutuante, mesmo se os operandos forem inteiros
Divisão Inteira	//	Utilizada para realizar a divisão e retornar apenas a parte inteira do resultado, descartando qualquer parte fracionária.
Módulo	%	Utilizada para calcular o resto da divisão entre dois números.

Fonte: Elaborado pelo autor.

Além dos operadores de matemática, também são muito importantes na programação os operadores relacionais, eles são fundamentais na programação Python, pois são usados para realizar comparações de valores, tomar decisões, validar dados, ordenar dados e controlar o fluxo de execução do código (Lutz, 2007). Eles são uma ferramenta essencial para expressar lógica condicional e realizar operações de comparação em programas Python. Os operadores relacionais são:

Quadro 3 - Exemplo operadores relacionais em Python

Operação	Operador	Explicação
Igualdade	==	Verifica se dois valores são iguais.
Desigualdade	!=	Verifica se dois valores são diferentes.
Maior que	>	Verifica se o valor da esquerda é maior que o valor da direita.

Menor que	<	Verifica se o valor da esquerda é menor que o valor da direita.
Maior ou igual a	>=	Verifica se o valor da esquerda é maior ou igual ao valor da direita.
Menor ou igual a	<=	Verifica se o valor da esquerda é menor ou igual ao valor da direita.
Pertence a	in	Verifica se um valor está contido em uma sequência, como uma lista, tupla ou string.
Não pertence a	not in	Verifica se um valor não está contido em uma sequência.

Fonte: Elaborador pelo autor.

1.5.3 Funções em Python

As funções desempenham um papel essencial na estruturação, organização e reutilização de código em programas Python. Elas promovem a modularidade, a abstração, a reutilização de código e a facilidade de manutenção, tornando o desenvolvimento de software mais eficiente e legível.

Para construir uma estrutura básica de função em Python sem passar argumentos, basta criar um arquivo de extensão *.py* e iniciar os passos a seguir:

1. Declarar *def* é a palavra-chave utilizada para definir uma função em Python.
2. Declarar o nome_da_funcao, é o nome escolhido para a função. Indica-se que ele representa a operação que irá realizar.
3. Os parênteses vazios () indicam que a função não recebe argumentos.
4. Os dois pontos : indicam o início do bloco de código da função.

Assim como nas funções matemáticas é possível passar um ou mais argumentos nas funções em linguagem de programação Python. Como por exemplo, criar uma função que some dois valores numéricos e retorne o valor da soma.

```
def soma(x, y):  
    # Corpo da função  
    resultado = x + y  
    return resultado
```

Neste caso, *soma* é o nome da função, *x* e *y* são os parâmetros da função, isto é, os valores que serão passados para a função quando ela for executada. O conteúdo da função é indentado, e contém uma instrução que calcula a soma dos parâmetros *x* e *y* que por fim retorna o resultado da função.

1.5.4 Laços ou *Loops*

Os laços ou loops são uma parte essencial da programação Python e são amplamente utilizados em muitos tipos de programas para realizar tarefas repetitivas. Eles permitem que os códigos sejam mais conciso e eficiente, automatizando processos repetitivos e simplificando a interação sobre grandes conjuntos de dados (Lutz, 2007).

Em Python existem dois principais tipos de laços, o *while* e o *for*. O laço *while* executa um bloco de código enquanto uma condição especificada for verdadeira, e continua a executar o conteúdo do laço repetidamente até que a condição se torne falsa. Por exemplo:

```
contador = 0  
while contador < 5:  
    print(contador)  
    contador += 1
```

Este laço *while* imprimirá os números de 0 a 4, pois o conteúdo dele será executado enquanto o valor do contador for menor que 5.

Enquanto isso, o laço *for* é usado para iterar sobre um objeto sequencial como uma lista, tupla, string ou dicionário. E por fim, executa o conteúdo do laço para cada item na sequência. Por exemplo, no caso abaixo, o laço *for* imprimirá cada item na lista, um de cada vez.

```
for item in lista:  
    print(item)
```

1.5.5 Testes *if* e *else*

Os testes *if* e *else* em Python permitem tomada de decisões, controlar o fluxo de execução do programa, validar dados de entrada e lidar com exceções de maneira controlada. Eles são uma parte essencial da estruturação e lógica de muitos programas em Python.

O teste *if* é usado para verificar se uma condição é verdadeira. Se a condição for verdadeira, o conteúdo dentro do teste é executado, se a condição for falsa, o teste *if* é ignorado. O teste *else* é usado em conjunto com o teste *if* para executar um bloco de código alternativo caso a condição do teste *if* não seja verdadeira. O conteúdo dentro do teste *else* é executado apenas se a condição do teste *if* for falsa. Além disso, em Python, também existe o teste *elif* (uma abreviação de *else if*) para verificar múltiplas condições em sequência. O teste *elif* é executado apenas se a condição do teste *if* anterior for falsa e a condição do próprio teste *elif* for verdadeira. Conforme podemos observar neste exemplo:

```
idade = 18
if idade < 18:
    print("Você é menor de idade.")
elif idade == 18:
    print("Você acabou de atingir a maioridade.")
else:
    print("Você é maior de idade.")
```

Neste exemplo, o teste *elif* é usado para verificar se a variável *idade* é igual a 18. Se for verdadeiro, o programa imprimirá "Você acabou de atingir a maioridade." Se a idade for menor que 18, imprimirá "Você é menor de idade." Caso contrário, imprimirá "Você é maior de idade."

1.5.6 Importar biblioteca e módulos

As bibliotecas e módulos em Python são uma parte essencial na programação, pois fornecem recursos, ferramentas e funcionalidades adicionais que ampliam as capacidades básicas da linguagem e permitem o desenvolvimento de aplicações mais eficientes, robustas e escaláveis (Lutz, 2007).

Geralmente, uma biblioteca em Python é composta por vários módulos, que oferecem funcionalidades específicas. Por exemplo, a biblioteca NumPy é utilizada para computação

numérica, enquanto a biblioteca Pandas é utilizada para manipulação de dados e a biblioteca VPython é utilizada para criação e manipulação de objetos 3D.

Além disso, existem bibliotecas nativas do Python, ou seja, um conjunto de módulos que são distribuídos junto com a instalação padrão do Python. Uma dessas bibliotecas é a *math*, utilizada para realizar cálculos numéricos. Para isso é preciso importar a biblioteca no código Python por meio da declaração *import* seguida pelo nome da biblioteca.

```
import math
```

Após importar a biblioteca *math*, pode-se acessar suas funcionalidades usando a sintaxe *nome_da_biblioteca.nome_da_função*. Por exemplo, a função *sqrt()* da biblioteca *math* calcula a raiz quadrada de um número, conforme exemplo abaixo.

```
import math

# Calculando a raiz quadrada de 16
raiz_quadrada = math.sqrt(16)
print("A raiz quadrada de 16 é:", raiz_quadrada)
```

Portanto, as bibliotecas são partes essenciais na construção de códigos pois fornecem conjuntos de funções e métodos pré-escritos que facilitam a realização de tarefas complexas. Além de economizar tempo de desenvolvimento, promover a reutilização de código, garantir a padronização e reduzir a probabilidade de erros.

2. MATERIAIS E ESTRATÉGIA DE ENSINO

Aliada à metodologia de ensino utilizada na intervenção pedagógica, descrevemos nesta seção os materiais e recursos utilizados para elaboração da sequência didática, atividades e tarefas, assim como as estratégias de ensino que foram utilizadas. Para tal, apresentamos a seguir uma breve contextualização da origem dos recursos, com o objetivo de familiarizar o leitor com os conceitos de cada recurso e estratégia, além de justificar os procedimentos e recursos escolhidos pelo autor desta pesquisa, para então detalhá-los nos subtópicos deste capítulo.

Primeiramente, no contexto de Ambiente Virtual de Aprendizagem (AVA), recurso essencial para viabilização das interações e dos usos dos recursos tecnológicos, escolhemos utilizar o Moodle como Sistema de Gestão de Aprendizagem (SGA ou LMS na sigla em inglês), por ser um dos recursos disponíveis pela Universidade Federal de São Carlos. Além disso, o Moodle também é o SGA mais utilizado do mundo, contando hoje com cerca de 156 mil sites baseados em sua plataforma, 44 milhões de cursos em 42 idiomas, 2.5 bilhões de inscrições e mais de 367 milhões de usuários (Moodle, 2023b).

Vale lembrar que o Moodle também é uma ferramenta que, quando usada de forma consciente pelo docente, pode trazer diversos benefícios na prática de ensino, pois pode facilitar a apresentação de conteúdos e conhecimentos, gerar atividades didáticas que potencializam o ensino, incentivar o trabalho autônomo e em grupo. Logo, não há dúvidas que os AVAs podem auxiliar e potencializar o desempenho das aulas, no entanto, é necessário se precaver para que eles não sejam subutilizados, relegando suas ações a tarefas administrativas e raramente a uma ferramenta de apoio ao aprendizado (Valenzuela-Zambrano; Perez-Villalobos, 2013).

Além disso, o Moodle possui diversas extensões (*plugins*) que podem ser incorporados ao Ambiente Virtual de Aprendizagem, e um deles é o CodeRunner, um *plugin* do tipo pergunta, de código aberto, criado inicialmente para ser usado no ensino de programação, mas que pode ser facilmente adaptado para outros contextos, além de estar disponível em diversos idiomas e suportar uma grande gama de linguagens de programação. O CodeRunner permite que o docente crie perguntas baseadas em alguma linguagem de programação e, após o aluno finalizar sua resposta, o próprio *plugin* avalia a resposta e emite um *feedback* (Lobb; Hunt, 2022).

De acordo com o site institucional do CodeRunner (2022), atualmente são mais de 2795 sites que utilizam seu *plugin* e, mesmo que existam outros softwares que exerçam a mesma tarefa, o CodeRunner se demonstrou uma ferramenta educacional com o propósito de correção autônoma de exercícios computacionais muito mais flexível, podendo ser utilizado em diferentes contextos na computação, desde os mais simples até os mais complexos (Benevides, 2019).

Outro aspecto importante desta pesquisa é o fato dos alunos poderem observar os fenômenos físicos de forma digital, como simulações pelo próprio código que eles criaram. Para isso, foi necessário pesquisar bibliotecas gráficas que contemplavam tanto os aspectos visuais de simulações Físicas, quanto características gráficas, como a demonstração de movimentos cinemáticos por meio de gráficos e que possuíam baixo grau de complexibilidade. Para tal, foi testado pelo pesquisador professor três bibliotecas: Turtle (Python, 2023), PyGame (PyGame, 2023) e VPython (GlowScript, 2017).

Para realizar esse teste, o pesquisador professor escreveu três programas de baixa complexibilidade com cada uma dessas bibliotecas, com a finalidade de simular um movimento retilíneo e uniforme de uma partícula e também apresentar de forma simultânea o gráfico do deslocamento em metros pelo tempo em segundos, esses programas podem ser encontrados no GitHub do pesquisador professor² (Yamaguti, 2023).

Tanto a biblioteca Turtle como a PyGame possuíam suporte para criação e movimentação de objetos 2D de forma simples, com ligeira vantagem para a Turtle por ser uma biblioteca nativa do Python. Porém, para um maior o grau de complexibilidade dos movimentos e interações entre as partículas é possível perceber pela documentação da biblioteca PyGame que ela possui métodos mais robustos para simular fenômenos físicos como os de colisões. No entanto, ambas as bibliotecas não possuem suporte para construção de gráficos, sendo necessário utilizar outras bibliotecas para construção de gráficos

Enquanto isso, a biblioteca VPython se demonstrou mais versátil, pois além de possuir métodos de criação de figuras 2D e 3D (as duas primeiras permitem apenas 2D), também possui suporte para construção de gráficos. Não bastando, essa biblioteca também dispõe de uma IDE online chamada GlowScript³, o que aumenta o arcabouço de ferramentas para realizar um ensino de Física e programação que alcance todos os tipos de público, uma vez

² Disponível em: <https://github.com/mateusyamaguti/Testes-de-Bibliotecas-Para-Animacao-Fisicas>

³ Disponível em: <https://www.glowscript.org/>

que essa ferramenta permite que se construa programas tanto pelo smartphone como em computadores com menor capacidade de processamento.

Com isso, segundo Cruz *et al.* (2022), a linguagem Python destaca-se pela sua sintaxe, em geral, simples e robusta, sendo apropriada para aqueles com pouca ou nenhuma experiência em programação e a escolha pela biblioteca VPython ocorre, principalmente, pois o programador pode, de forma prática e simples, criar elementos geométricos tridimensionais em sua interface, atribuindo-lhes cor, tamanho, posição, massa e velocidade.

Optou-se também nesta pesquisa pelo emprego da técnica de desenvolvimento guiado por testes (TDD, do inglês *Test Driven Development*), a qual consiste em criar testes unitários que validem automaticamente o código principal em pequenos ciclos, com o intuito de impedir o desenvolvimento de programas grandes com falhas estruturais que podem vir a ser detectadas apenas na fase de produção ou execução (Beck, 2010).

O TDD na linguagem de programação Python é usualmente implementado por meio da biblioteca PyTest⁴, a qual possibilita que o programa a ser testado retorne uma mensagem de erro ou acerto para o desenvolvedor do programa, orientando-o se as condições expressas nos testes foram aceitas ou se o código precisa ser corrigido, e para essa pesquisa o pesquisador professor adaptou o seu uso para incorporação do *plugin* CodeRunner.

A técnica TDD usualmente utilizada no desenvolvimento de softwares e programas, consiste em um ciclo composto por três etapas. A primeira etapa é a criação de um teste que falhe, ou seja, antes de começar a desenvolver o código, é criado um teste que o programador precisa resolver. Para isso, é crucial compreender os requisitos da situação em questão com profundidade suficiente do cenário que o programa está inserido, para escrever um teste que avalie o que se espera ser implementado (Beck, 2010). Uma vez que esse teste avalia um código que ainda não existe, é natural que inicialmente ele falhe.

Posteriormente, a segunda etapa é fazer o teste que está falhando “passar”. Nesta etapa, é importante que o programador concentre-se exclusivamente em escrever o código necessário para fazer com que o teste que estava falhando comece a passar. Ao fazer isso, é fundamental garantir que os testes anteriores, se houver, continuem a ser aprovados (Beck, 2010). Durante essa etapa, o objetivo é fazer com que todos os testes sejam bem-sucedidos.

Em seguida, a terceira etapa é refatorar o código desenvolvido. Com o código escrito e os testes com êxito, é o momento de alocar um tempo para refatorar o código. Isso envolve a

⁴ Disponível em: <https://docs.pytest.org/en/7.4.x/>

eliminação de práticas ruins e a simplificação do código, tornando-o mais legível e sustentável para quem futuramente irá trabalhar nele (Beck, 2010). Essa etapa é conhecida como "refatoração".

Com isso, essa técnica pode ser conceitualmente aproximada ao que Valente (2014) chama de ciclo de “descrição - execução - reflexão - depuração - nova descrição” (DERDND):

Primeiro, a descrição de ideias pode ser entendida como a representação dos conhecimentos que o aprendiz possui. Nessa representação é possível identificar, do ponto de vista cognitivo, os conceitos e as estratégias que o aprendiz utiliza para resolver um problema ou projeto. Segundo, as TDIC executam as instruções fornecidas, o que não acontece com os objetos tradicionais da nossa cultura. Tal execução permite verificar se os conceitos e estratégias utilizadas são adequados ou merecem ser depurados. Terceiro, pelo fato de estar trabalhando com o digital, as alterações a serem realizadas nas atividades são facilmente implementadas, o que facilita a realização do ciclo de ações descrição-execução-reflexão-depuração-nova descrição (Valente, 2014, p. 146).

Esse ciclo consiste, primeiramente, na descrição da resolução do problema em termos da linguagem de programação, o que implica em utilizar todo o conhecimento estrutural disponível, incluindo os conceitos relacionados ao problema, estratégias para aplicação desses conceitos, compreensão do funcionamento do computador e da linguagem de programação em uso (Valente, 1999). O objetivo é representar e explicar os passos da resolução do problema na linguagem de programação escolhida.

A segunda etapa é a execução da descrição pelo computador, estágio em que a descrição de como o problema é resolvido na linguagem de programação é executada pelo computador. A execução fornece um *feedback* do programa, isto é, uma resposta ou retorno de informação do computador para o usuário aluno. O resultado obtido é uma consequência direta do que foi instruído ao computador (Valente, 1999). Por exemplo, em casos gráficos o aluno pode observar a figura se formando na tela, avaliar o produto final e refletir sobre as informações apresentadas.

A terceira etapa corresponde à reflexão sobre o que foi produzido pelo computador. Nesta etapa, o processo de reflexão sobre os resultados do programa pode levar a algumas alternativas. O aluno pode optar por não modificar o procedimento, se os resultados obtidos correspondem às suas intenções iniciais para a resolução do problema, considerando o problema como resolvido (Valente, 1999). Por outro lado, se o resultado difere das intenções originais, o aprendiz depura o procedimento, ajustando-o para atingir os resultados desejados.

E a quarta etapa, depuração dos conhecimentos por intermédio da busca de novas informações ou do pensar. Aqui, o aluno tem a opção de buscar informações adicionais, como conceitos específicos em uma determinada área, métodos da linguagem de programação, estratégias de computação ou técnicas para a resolução de problemas. Essas informações são assimiladas na estrutura mental do aluno, tornando-se parte de seu conhecimento (Valente, 1999). Ele então utiliza esse conhecimento para aprimorar o programa, iniciando assim um novo ciclo de descrição - execução - reflexão - depuração - nova descrição.

Desta forma, foi criado pelo pesquisador professor atividades no *plugin* CodeRunner, relacionando o conteúdo de cinemática com os conhecimentos de programação em Python, com o objetivo do aluno também ter um aprendizado guiado automaticamente pelos *feedbacks* dos testes criados a partir da aproximação da técnica TDD presente nos exercícios do CodeRunner, chegando aos critérios físicos estabelecidos nas questões, ou seja, o programa criado pelo aluno deve satisfazer o fenômeno físico representado pelas situações problema.

Sendo assim, a primeira etapa do TDD foi criada pelo pesquisador professor utilizando o *plugin* CodeRunner, o qual tem o objetivo de criar testes que orientem os alunos na primeira etapa do ciclo DERDND, a descrição, a qual é similar a segunda etapa do TDD que tem por objetivo fazer os testes terem êxito. Tendo concluído essa etapa o aluno pode prosseguir para a execução da descrição pelo computador, ou seja, o aluno executa o código. Nesta etapa o aluno recebe o *feedback* tanto da execução do código como também dos testes criados pelo pesquisador professor, nesse processo o *feedback* criado pela técnica do TDD pode devolver aos alunos orientações sobre os pontos de falha e acerto nos códigos de forma objetiva e precisa, enquanto o programa que foi executado retorna uma resposta que pode ser interpretada pelo aluno tanto de forma objetiva, como de forma interpretativa, como em casos gráficos ou de simulações. Logo essas etapas do ciclo DERDND de reflexão sobre o que foi produzido e, também, de depuração dos conhecimentos por intermédio da busca de novas informações ou do pensar. Após refletir sobre o que foi produzido e compreender os erros e acertos, a etapa de depuração uni-se ao processo de refatoração do código em TDD, buscando melhorar o programa que foi criado e reiniciar ambos os ciclos.

Logo, essa aproximação consiste no fato do professor ter conhecimentos sobre os potenciais educacionais da linguagem de programação Python e ser capaz de traduzi-las adequadamente às práticas de ensino e aprendizagem sobre os conhecimentos científicos, que não podem se limitar o processo do uso da linguagem de programação apenas a características instrumentistas, visto que é nítido um processo de diálogo e comunicação entre a máquina e o

aluno quando observamos que ao executar o código o computador devolve *feedback* de erros e acertos criados tanto pelo professor, quanto pelo próprio aluno.

2.1 Moodle

Para esta pesquisa, o pesquisador professor criou um ambiente Moodle, ou seja, um site personalizado intitulado CodeGenius⁵ no intuito de angariar maior impacto na divulgação da pesquisa pensando no público infantojuvenil. Essa decisão também foi tomada pois o sistema AVA Moodle da UFSCar que permite a inscrição de participantes externos não possui o *plugin* CodeRunner, o qual será detalhado ainda neste capítulo e é um dos recursos fundamentais para realização dessa pesquisa. Assim, foram disponibilizados no Moodle as aulas, os recursos de aprendizagem e as atividades avaliativas. Além disso, o design do ambiente foi adaptado para atender às necessidades específicas dos participantes da pesquisa e incluiu elementos de interatividade para promover a participação ativa dos alunos.

Desta forma, para garantir a acessibilidade dos alunos e o desempenho da plataforma Moodle, foi preciso realizar sua aplicação em um servidor web, para isso foi escolhido o servidor Contabo por possuir uma VM (*Virtual Machine* - Máquina virtual) de baixo custo e com todos os parâmetros de segurança de dados necessários para garantir a proteção dos participantes, atendendo assim às necessidades desta pesquisa.

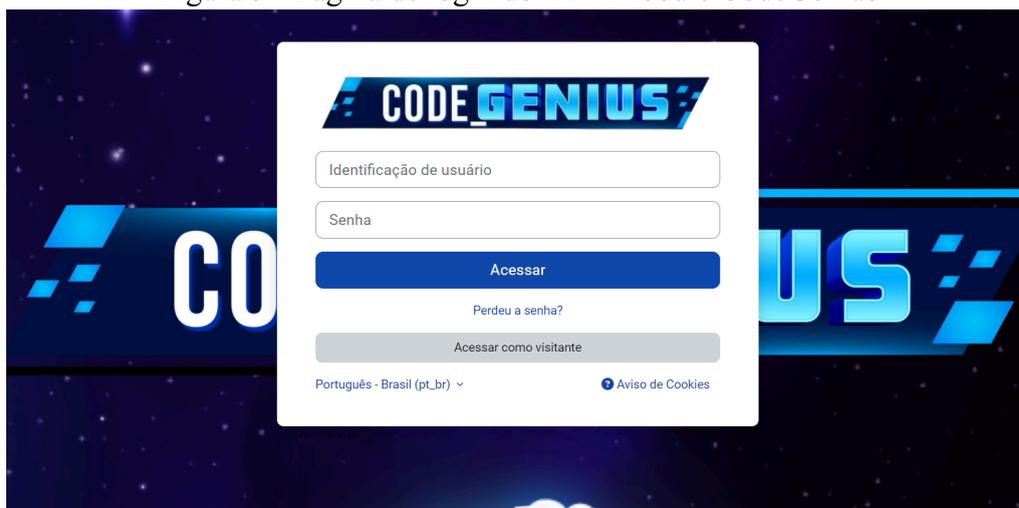
Para que a aplicação Moodle funcionasse corretamente foi necessário a configuração dos seguintes pré-requisitos como orientados pelo site oficial do Moodle (2023a): Um servidor web funcionando (p.ex. Apache), um banco de dados (p.ex. MySQL, MariaDB ou PostgreSQL) que tenha o PHP configurado. Com isso, fechamos a composição estrutural com estes itens relacionados: Moodle (v. 4.2), Nginx, PHP 8, Postgre, SQL 15, CloudFlare DNS + Certificado, Backup S3 AWS e Contabo (VM).

Não sendo o bastante, também deixamos no apêndice desta pesquisa a documentação utilizada para se realizar essa aplicação (APÊNDICE A). Vale ressaltar que muitas dessas ações e escolhas de recursos foram motivadas pela experiência em tecnologia do pesquisador professor, autor desta pesquisa, uma vez que não cabe a essa pesquisa detalhar o processo de instalação do AVA, e sim buscar utilizá-lo como ferramenta estratégica e metodológica para entender as potencialidades do uso da linguagem Python no ensino de Física aliada aos 3MP.

⁵ Disponível em: <https://codegeniuskids.com.br/>

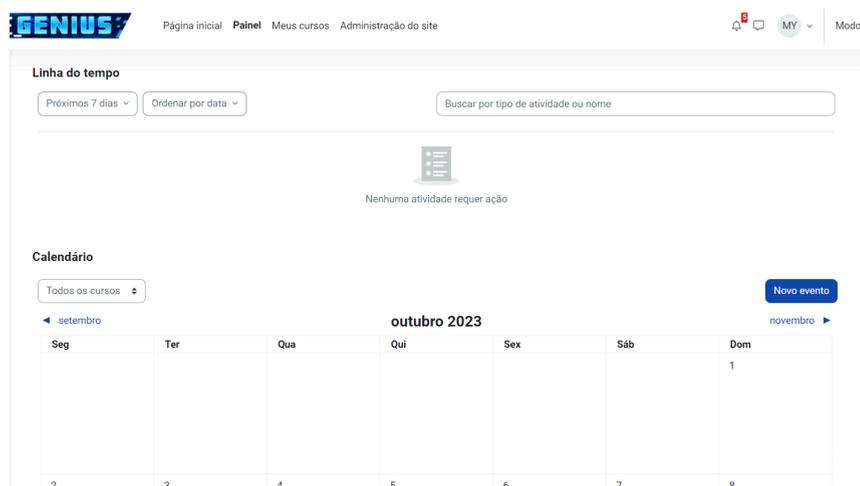
E com o objetivo de deixar o design do AVA mais agradável foi instalado o *plugin* Moove⁶ (Mano, 2023) cujo tema de sua aparência busca tornar os ambientes online mais limpos e intuitivos para alunos e educadores em comparação com a interface padrão do Moodle, como pode se observar na figura 1, 2 e 3.

Figura 3 - Página de login do AVA Moodle CodeGenius



Fonte: Elaborado pelo autor.

Figura 4 - Painel principal com ênfase no calendário de atividades e recursos.



Fonte: Elaborado pelo autor.

⁶ Disponível em: https://moodle.org/plugins/theme_moove

Figura 5 - Página principal de cursos.



Fonte: Elaborado pelo autor.

Por fim, foi necessário instalar o *plugin* CodeRunner⁷ para encerrar a configuração de todo Ambiente Virtual de Aprendizagem (AVA), visto que o CodeRunner não é nativo do Moodle. Sua instalação pode ser facilmente realizada pela guia de *plugins* presente na administração do site, desde que o diretório moodle/question/behaviour esteja liberado no servidor.

2.2 Atividades

Este tópico tem o objetivo de apresentar as atividades que foram desenvolvidas pelo pesquisador professor para serem trabalhadas pelos alunos após a realização de cada aula da sequência didática elaborada.

⁷ Documentação para instalação do Plugin CodeRunner:
https://github.com/trampgeek/moodle-qtype_coderunner/blob/master/Readme.md#installing-coderunner

2.2.1 Questões CodeRunner

Nesta pesquisa buscou-se planejar e estruturar todas as atividades previamente antes da aplicação da sequência didática FIS_PY, mesmo que essas atividades pudessem ser alteradas e ajustadas durante o período de execução das aulas.

Desta maneira, foram criadas 21 questões CodeRunner divididas entre as aulas (segunda, terceira e quarta aula) que compreendem o segundo momento pedagógico e oportunizam mais uma vez a aprendizagem dos conceitos e conhecimentos trabalhados durante esses encontros.

Além disso, como o Moodle e o CodeRunner não oferecem ferramentas de design de texto avançadas para usuários comuns, para confecção do padrão de texto das questões foi desenvolvido um template padrão em linguagem de marcação HTML, a fim de apresentar as questões em um formato mais limpo e com um aspecto que remetesse a linguagem de programação conforme pode ser observado na figura 4, a qual após ser renderizada gera o modelo de questão representado pela figura 5.

Figura 6 - Modelo em linguagem HTML da questão padrão a ser utilizadas nas demais atividades CodeRunner

```
<p style="font-family: Cascadia Code;">Escreva uma
fun&ccedil;&atilde;o de nome SubMultMetro, que receba como
par&acirc;metro um n&uacute;mero que represente a quantidade de
metros e retorne seu submultiplo em cent&iacute;metros, conforme
exemplo abaixo:</p>
<p dir="ltr">&nbsp;</p>
<div style="background-color: #1e1e1e; font-family: Cascadia Code;
color: white; padding: 1rem;">
<p><span style="font-size: 0.9375rem; color: springgreen;">In [
]:</span><span style="font-size:
0.9375rem;">SubMultMetro(15)</span></p>
<p><span style="font-size: 0.9375rem; color: red;">Out[
]:</span><span style="font-size: 0.9375rem;">1500</span></p>
</div>
```

Fonte: Elaborado pelo autor.

Figura 7 - Modelo ilustrativo da questão padrão resultado da renderização do exemplo da figura 06 a ser usado nas demais atividades CodeRunner

CR **000_Question_Model** **Versão 1 (mais recente)**

Questão 1
Incompleto
Vale 1,00 ponto(s).

Escreva uma função de nome `SubMultMetro`, que receba como parâmetro um número que represente a quantidade de metros e retorne seu submúltiplo em centímetros, conforme exemplo abaixo:

```
In [ ]:SubMultMetro(15)
Out[ ]:1500
```

Answer: (penalty regime: 0 %)

1

Fonte: Elaborado pelo autor.

Além disso, todas as questões construídas por meio deste plugin foram desenvolvidas no formato de função pelo pesquisador professor, o que também facilita a construção dos testes no CodeRunner, por conta de testar suas aplicações de forma única, o que tornou o processo de testagem da função mais claro, uma vez que basta testar o retorno da função e não especificamente a semelhança das linhas de código, o que gera a flexibilidade do estudante construir códigos diferente mas com a obtenção do mesmo resultado.

Sendo assim, podemos visualizar a construção de uma questão CodeRunner no exemplo da figura 6, onde no campo *Answer*, o pesquisador professor implementa uma função que tem o objetivo converter a unidade de medida metro para seu submúltiplo centímetro. Depois desta etapa é realizada a criação do teste da questão dentro do próprio *plugin* CodeRunner, conforme apresentado na figura 7.

Figura 8 - Ilustração da resposta à pergunta da Figura 7

✓ Answer

```
Answer ?  
1 def SubMultMetro(metro):  
2     sub_mult_metro = metro*100  
3     return sub_mult_metro
```

Fonte: Elaborado pelo autor.

Figura 9 - Ilustração da configuração de teste realizada no CodeRunner referente a figura 8

✓ Test cases

Test case 1 ?	<input type="text" value="print(SubMultMetro(15))"/>
Standard Input ?	<input type="text"/>
Expected output ?	<input type="text" value="1500"/>

Fonte: Elaborado pelo autor.

Pode-se observar na Figura 7, que no campo *Test Case 1* foi utilizada a função *print()* responsável por gerar o *output*, ou seja, a saída da informação da função que foi criada na etapa anterior que será comparada de forma automática pelo *plugin* CodeRunner, o qual verifica se a informação inserida no campo *Expected output* é igual ao teste realizado no campo *Test Case 1*.

Todas essas configurações foram realizadas previamente pelo pesquisador professor ao criar as questões CodeRunner, enquanto isso, para resolver essas questões os alunos precisariam acessar a atividade criadas pelo pesquisador professor no Moodle onde consta

para cada questão uma IDE que o aluno pode escrever a solução do problema, conforme apresentado na figura 8.

Por fim, as questões CodeRunner ainda contemplam a possibilidade da verificação das respostas elaboradas pelos alunos, mostrando as diferenças e erros existentes em comparação com a saída esperada, dando ao aluno a oportunidade de depuração de seu código e consequentemente a refatoração do mesmo.

Essa situação pode ajudar na resolução final do exercício, além de se relacionar com a prática da programação TDD e a prática de ensino citada por Valente (2014); como sendo o ciclo de ações referente a descrição - execução - reflexão - depuração - nova descrição.

Figura 10 - Ilustração de questão CodeRunner da figura 8 sendo verificada automaticamente

Escreva uma função de nome SubMultMetro, que receba como parâmetro um número que represente a quantidade de metros e retorne seu submúltiplo em centímetros, conforme exemplo abaixo:

```
In [ ]: SubMultMetro(15)
Out [ ]: 1500
```

Answer: (penalty regime: 0 %)

```
1 def SubMultMetro(metro):
2     sub_mult_metro = metro*100
3     return sub_mult_metro
```

Verificar

	Test	Expected	Got	
✓	print(SubMultMetro(15))	1500	1500	✓

Run using the University of Canterbury's Jobe server. This is for initial testing only. Please set up your own Jobe server as soon as possible. See here.

Passou em todos os teste! ✓

Correto
Notas para este envio: 1,00/1,00.

Fonte: Elaborado pelo autor.

2.2.2 Atividade de arrastar e soltar e de múltipla escolha

Para contemplar uma maior gama de evidências que destaquem as possíveis potencialidades do tema desta pesquisa e também enriquecer a diversidade de atividades a fim de proporcionar aos alunos diferentes formas de exercitar seu conhecimento e consolidar seu aprendizado, foram desenvolvidas pelo pesquisador professor outras duas modalidades de atividade: questões de arrastar e soltar e de múltipla escolha.

As questões de arrastar e soltar foram criadas no intuito de apresentar alguns exemplos de estrutura de código para os alunos, assim como reforçar os pontos-chaves sobre as estruturas de código em programação e os termos de sintaxe padrão da linguagem Python, os quais seguem um padrão de escrita com alguns detalhes de pontuação que podem fugir aos olhos dos programadores até mais experientes, como por exemplo o uso de palavras reservadas pela linguagem, a tabulação ou indentação correta e até mesmo os “dois pontos” no final das estruturas condicionais, função ou repetições.

Para isso foram criados seis exercícios de arrastar e soltar distribuídos entre o terceiro e quarto encontro, em que o pesquisador professor apresentou aos alunos durante as aulas como realizar essas atividades de acordo com o conteúdo abordado durante as aulas online. De forma semelhante ao CodeRunner, o padrão de texto das questões também foram desenvolvidos em HTML para constituir uma aparência mais clara e próxima das IDE de programação, resultando no exemplo da figura XX.

Figura 11 - Exemplo de questão de arrastar e soltar no Moodle personalizada através da linguagem de marcação HTML

Preencha as lacunas abaixo para que a função crie a esfera com sucesso e retorne a posição do eixo x.

```
from vpython import *  
  
def CriarEsfera(raio):  
    esfera = sphere(pos=vector(0, 0, 0), [ ] =raio, color=color.cyan)  
    return esfera.pos.[ ]
```

Fonte: Elaborada pelo autor.

Além disso, foram criadas outras duas questões de múltipla escolha de caráter interpretativo, em que o estudante pode interpretar o código relacionado com os conceitos de cinemática para chegar ao resultado solicitado ou até mesmo digitar o código em alguma IDE para analisar o resultado final e atribuir a resposta correta. Em segundo plano, esses exercícios também davam subsídios aos alunos para realizar a construção da atividade final a ser apresentada no terceiro momento pedagógico que corresponde ao quinto encontro desta sequência didática.

Figura 12 - Exemplo de questão de múltipla escolha

Um estudante desejou encapsular o deslocamento de uma esfera em uma função, que resultou no código abaixo. O código finaliza retornando a posição final da esfera. Qual seria essa posição?

```
from vpython import *

def DeslocamentoDaEsfera():
    esfera = sphere(pos=vector(-5, 0, 0), radius=0.5, color=color.cyan)
    parede = box(pos=vector(32.5, 0, 0), size=vector(75,12,0.2),
color=color.green)
    esfera.velocity = vector(25, 0, 0)
    deltat = 0.005
    t = 0
    while t < 2.995:
        rate(100)
        t = t + deltat
        esfera.pos = esfera.pos + esfera.velocity*deltat
    return esfera.pos.x

DeslocamentoDaEsfera()
```

- a. 65
- b. 69
- c. 70
- d. 71
- e. 75

Fonte: Elaborado pelo autor.

2.2.3 Bibliotecas Gráficas em Python

Antes de iniciar a criação de quaisquer conteúdos didáticos, foi necessário testar algumas bibliotecas da linguagem Python que possuem a função de criar objetos 2D ou 3D com a finalidade de realizar animações gráficas capazes de demonstrar os fenômenos físicos de cinemática abordados nesta pesquisa.

Sendo assim, o pesquisador professor escolheu três bibliotecas Python para testar as potencialidades do seu uso em aulas. A escolha dessas bibliotecas se deu inicialmente pela experiência do pesquisador professor e por algumas características motivacionais específicas de cada uma delas. Logo, a biblioteca Turtle foi escolhida por ser nativa da linguagem Python, não tendo a necessidade de instalação por parte dos usuários iniciantes. A biblioteca PyGame, foi escolhida por conta de sua vasta coletânea de jogos que foram desenvolvidos por ela e por ser popularmente conhecida. E a biblioteca VPython, foi selecionada por possuir a capacidade de realizar simulações com objetos 3D e construir gráficos, além disso, em seu repositório no site GlowScript também existem vários exemplos de simulações físicas prontas que podem servir de exemplos para os alunos.

A partir disso, o pesquisador criou três códigos simples para avaliar as bibliotecas que poderiam ter o melhor desempenho a fim de representar a simulação de um MRU de acordo com os critérios de: criação de objetos 2D e 3D, facilidade de escrita do código, capacidade de representação das simulações e de criação de gráficos. Os códigos que foram construídos encontram-se no Apêndice B.

Com isso, observou-se que o código MRU_PyGame (PyGame, 2023) criado a partir da biblioteca PyGame possui os elementos necessários para criação de animações 2D de boa qualidade. Com poucas linhas de código é possível criar a janela de animação e seus elementos, todavia ela não permite a criação de gráficos. De forma semelhante, o código MRU_Turtle (Python, 2023) também foi capaz de criar animações 2D por meio da biblioteca Turtle a qual é nativa da linguagem Python, mas também não possui suporte para criação de gráficos. Sendo assim, a única biblioteca que compreendeu todos os critérios necessários foi a VPython, apresentada pelo código MRU_VPython (GlowScript, 2017), onde foi possível observar a criação de objetos 3D os quais possuem funções específicas com argumentos internos que oportunizam a configuração das características dos objetos, diferente das outras bibliotecas que era necessário usar algum método para alterar as características dos objetos que foram criados. Além disso, a biblioteca VPython demonstrou um suporte amplo na criação de gráficos, uma vez que possui dois métodos de construção. Um que é mais rápido

(atualmente baseado em Flot) e foi escolhido para esta pesquisa, e o outro que oferece recursos interativos avançados, como zoom e panorâmica, mas é mais lento (atualmente baseado em Plotly).

Por esses motivos foi escolhida a biblioteca VPython como mais um recurso da linguagem de programação Python a fim de potencializar a visualização dos fenômenos físicos de cinemática a serem estudados, a criação de gráficos digitais, a racionalização do pensamento computacional e a lógica matemática.

2.3 Metodologia de Ensino

A BNCC (BRASIL, 2017) norteia que o Ensino Médio contemporâneo tem por responsabilidade garantir aos estudantes a preparação básica para o trabalho e a cidadania, assegurando a compreensão dos fundamentos científico-tecnológicos dos processos produtivos. Assim sendo, esta pesquisa busca interligar a linguagem de programação com os meios de produção tecnológica, bem como o desenvolvimento de conhecimentos científicos de Física para o ensino médio.

Desta forma, para atender a BNCC, escolheu-se utilizar os 3MP como metodologia de ensino conforme exposto na fundamentação teórica desta pesquisa, em que o primeiro momento pedagógico se refere à etapa de problematização inicial. Nesta etapa, são expostas aos alunos situações reais e problemas desafiadores que contemplam o conteúdo da aula, no intuito do professor entender o conhecimento de cada aluno e como eles interpretam o mundo. Além disso, segundo os proponentes do modelo,

A finalidade desse momento é propiciar um distanciamento crítico do aluno ao se defrontar com as interpretações das situações propostas para discussão, e fazer com que ele sinta a necessidade da aquisição de outros conhecimentos que ainda não detém (Muenchen; Delizoicov, 2014, p. 620).

Desta forma, a fim de introduzir os conhecimentos de cinemática, como por exemplo: o tempo gasto pelo aluno no percurso entre a escola e sua casa com diferentes meios de transporte, a velocidade média dos jogadores de futsal no campeonato interclasse e até mesmo a velocidade média necessária para o aluno chegar à sala de aula nos minutos finais do intervalo.

Em seguida, o segundo momento pedagógico é a organização do conhecimento, destinado ao ensino dos conhecimentos de Física necessários orientado pelo professor, para que ocorra a compreensão do tema da aula, assim como a resolução dos problemas expostos (Muenchen; Delizoicov, 2014). Essa etapa utilizou três das cinco aulas da sequência didática desta pesquisa, uma vez que foi necessário que o pesquisador professor promovesse tanto o ensino de Física como o ensino da linguagem de programação Python. Desta forma, em cada um dos encontros referente ao segundo momento, o pesquisador professor organizou os conhecimentos de Física e cinemática de tal forma, que ambos se relacionassem em busca de mediar a aplicação sistemática dos conhecimentos de linguagem de programação com aplicações Físicas. Tal organização está apresentada no subtópico “Planos de aula FIS_PY”.

O terceiro momento pedagógico é a aplicação do conhecimento, etapa destinada para apresentar sistematicamente os conhecimentos incorporados pelos alunos, seja em uma única aula, ou no final de uma sequência didática como realizado nesta pesquisa, a fim de analisar e interpretar as situações reais e os problemas desafiadores que foram exposto no primeiro momento dos 3MP, assim como outros temas e situações que surgiram durante o período da aplicação da sequência didática, em virtude de poder gerar aplicações desses conhecimentos (Muenchen; Delizoicov, 2014). Sendo assim, o propósito desta etapa foi compreender se os alunos conseguiram aprender os conhecimentos físicos e computacionais ensinados de forma a aplicá-los em uma atividade prática, a qual pedia a interpretação de uma situação Física que precisava ser representada por um programa em Python.

2.4 A sequência didática

Para realização desta pesquisa, foi dado o apelido de “Curso FIS_PY” à sequência didática no intuito de tornar mais atrativa a divulgação do curso. Além disso, conforme exposto no subtópico “Moodle” e “Atividades”, para realização das atividades e incorporação das aulas, foi escolhido o ambiente de aprendizado Moodle, como também, utilizamos os *plugins* o CodeRunner, que permite a criação de perguntas, as quais o professor tem a possibilidade de criar um programa para avaliar automaticamente a resposta do aluno (Lobb; Hunt, 2022).

Para captação audiovisual e realização das aulas, foi utilizada a ferramenta Google Meet, pois oferece a facilidade de gravação audiovisual, armazenamento dos dados no *chat box* da reunião, além de renderizar e armazenar automaticamente a captação audiovisual em

resolução HD. As gravações das aulas foram anexadas a uma pasta privada no Google Drive em formato de acesso restrito criado apenas para o fim desta pesquisa, lembrando que as aulas foram incorporadas ao ambiente Moodle, com a possibilidade de acesso restrito somente aos participantes dessa pesquisa, para que o aluno possa rever a aula caso necessário.

Os alunos inscritos na pesquisa foram cadastrados no AVA Moodle CodeGenius, como também, foi encaminhado um cronograma prévio por e-mail para todos os alunos com as orientações iniciais para participação da primeira aula.

Como orientado pela metodologia dos Três Momentos Pedagógicos, a primeira aula foi destinada ao primeiro momento. O objetivo foi apresentar questões sobre situações reais que se aplicam às leis básicas de cinemática, assim como as possíveis relações que a tecnologia pode ter com essas situações. As perguntas foram previamente estruturadas pelo pesquisador professor e podem ser encontradas no apêndice I, subtópico Slides Aulas 01. Todavia, pode ser que, durante o diálogo entre todos os participantes da pesquisa, surjam outras questões desafiadoras. Com isso, essa etapa tem o intuito de aproximar os participantes das aplicações reais que a cinemática possui, assim como as implicações tecnológicas que podem existir para resolver as questões e desafios expostos.

Posteriormente, seguindo a metodologia dos 3MP, o pesquisador professor analisou as questões expostas na primeira aula, a fim de concluir um tema que envolve tanto o ensino de cinemática, quanto a relação que a linguagem de programação possa ter com a resolução dos problemas expostos durante o primeiro momento. Desta forma, o pesquisador professor organizou os conhecimentos de cinemática e sobre a linguagem de programação Python a partir dos temas debatidos, bem como adaptou e criou exercícios para fundamentar os conhecimentos necessários para embasar uma aplicação no terceiro momento pedagógico.

Em decorrência disso, foi realizado o segundo momento em que, sob orientação do pesquisador professor, foram apresentados aos participantes tanto os conhecimentos em cinemática como em linguagem de programação Python, fazendo relação com o que foi debatido no primeiro momento.

Enfim, no terceiro momento, foi exposto pelo pesquisador professor, problemas reais que foram ou não levantados durante as aulas e que os alunos tinham que analisar e chegar a conclusões sobre eles, aplicando os conhecimentos aprendidos em linguagem de programação Python para solucionar essas situações problema.

Portanto, seguindo a linha de raciocínio dos 3MP, realizamos essa pesquisa em 5 encontros virtuais de duração de uma hora e meia (1h e 30min) cada, sendo um encontro destinado ao primeiro momento dos 3MP, para apresentação da oficina e familiarização de todos os participantes; três encontros para o segundo momento dos 3MP; e um encontro para o terceiro momento dos 3MP. Vide quadro 4 (pág. 64).

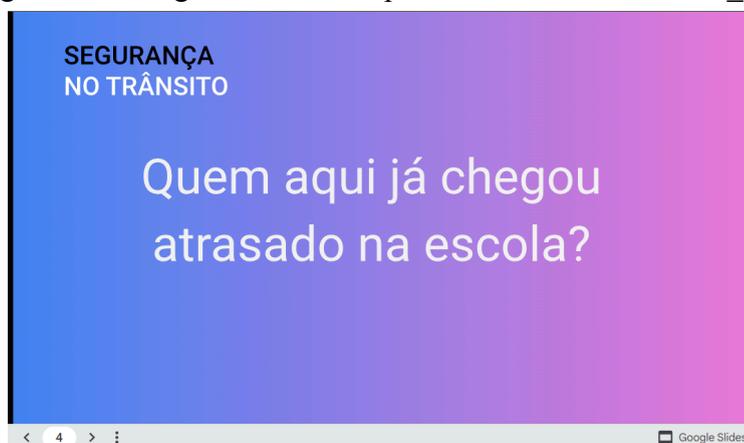
2.4.1 Material Didático

Para apoiar o pesquisador professor e também servir como material de consulta para os alunos, tendo em vista os extensos conteúdos abordados tanto de Física como de linguagem de programação Python, o pesquisador professor criou um material didático em formato de slides para apoiá-lo durante as aulas. O material foi anexado no moodle CodeGenius e liberado semanalmente para os alunos e pode ser encontrado no apêndice I.

Todo material foi criado previamente antes da realização do curso com perguntas introdutórias a partir de assuntos abrangendo a rotina de um estudante do ensino médio e exemplos de resolução de situações físicas resolvidas por meio da aplicação da linguagem de programação Python. Houve espaço para alteração dos assuntos e problemas, uma vez que as aulas foram liberadas semanalmente e, caso os assuntos sugeridos fugissem da problematização inicial realizada com os alunos, o pesquisador professor poderia ajustá-los.

Como exemplo, a figura 11 apresenta uma pergunta inicial sobre o cotidiano escolar dos alunos a fim de introduzir o conteúdo de cinemática, mais especificamente os conceitos de MRU. E na figura 12, pode-se observar a aplicação dos conceitos de MRU no desenvolvimento de uma função que calcula a velocidade média de uma partícula.

Figura 13 - Imagem do slide da primeira aula do curso FIS_PY



Fonte: Elaborado pelo autor.

Figura 14 - Imagem do slide da segunda aula do curso FIS_PY

Física

Por fim, Gabriel teve a surpreendente ideia de calcular a velocidade média de seus passeios para otimizar o tempo gastos em suas próximas viagens. Para isso, ele calculou a velocidade média de seus passeios que corresponde a **variação da distância percorrida pelo período de tempo gasto**.

$$v_m = \frac{\Delta S}{\Delta t}$$

Python

Como foram muitos passeios Gabriel criou um função em Python para retornar a velocidade média dos passeios e assim também agregou uma tolerância de duas casas decimais para ter mais precisão.

```
def VelocidadeMedia(deslocamento, tempo):
    v_medio = deslocamento / tempo
    return round(v_medio, 2)
print(VelocidadeMedia(150, 33))
```

Google Slides

Fonte: Elaborado pelo autor.

2.4.2 Planos de aula FIS_PY

Os planos de aula desta pesquisa foram construídos de forma que os conhecimentos abordados principalmente no segundo momento dos 3MP estejam organizados de forma que o aprendizado de cinemática ocorra concomitantemente com o aprendizado da linguagem de programação Python. Como exemplo, quando foi exposto o assunto de variáveis em Python, as variáveis declaradas são as mesmas utilizadas em cinemática, como tempo, espaço e velocidade. Seguindo essa abordagem, a estrutura dos planos de aula foi feita para relacionar constantemente os temas de Física e de linguagem de programação Python, no intuito de ambos os conhecimentos evoluírem gradativamente juntos.

Para uma compreensão melhor dos planos de aula, pode-se observar no Quadro 4 a forma resumida dos assuntos debatidos em cada encontro.

Quadro 4 - Ementa da sequência didática FIS PY

Encontros	Objetivos	Conhecimento em Física	Conhecimento em Python
1º Encontro (1º MP)	Problematização do conceito de cinemática e suas implicações na vida cotidiana com o uso de tecnologia.	<ul style="list-style-type: none"> Relacionar o conceito de cinemática com as situações da vida cotidiana; Entender as aplicações que o conceito de cinemática pode ter na vida cotidiana; Compreender como a tecnologia está inserida no 	<ul style="list-style-type: none"> Entender como a tecnologia se comunica com os seres humanos, com outras tecnologias e qual é essa linguagem; Aprender as orientações básicas para a utilização do portal AVA Aprender a utilização e instalação básica dos recursos

		contexto da vida cotidiana.	Python, PyCharm e GlowScript
2º Encontro (2º MP)	Aprender o que são grandezas escalares, o conceito e as características do MRU e os fundamentos de Python, inseridos na temática de uma viagem de passeio.	<ul style="list-style-type: none"> • Relacionar o conceito de cinemática com as situações da vida cotidiana; • Entender o conceito de grandezas Físicas e especificamente grandezas escalares; • Desenvolvimento do conceito e das equações do MRU (deslocamento, tempo e velocidade média). 	<ul style="list-style-type: none"> • Relacionar o conceito de variáveis na programação com a Física e a matemática; • Executar programas no Modo Script; • Aprender as operações matemáticas básicas na linguagem Python • Desenvolver e relacionar o conceito e importância de funções na programação com suas implicações na Física e na vida real.
3º Encontro (2º MP)	Aprender as características de um Movimento Retilíneo e Uniformemente Variado e suas respectivas funções, correlacionando com animações digitais em Python, inseridos na temática de uma viagem de passeio.	<ul style="list-style-type: none"> • Relacionar o conceito de grandezas vetoriais lembrando as aplicações de trigonometria; • Desenvolvimento do conceito e das equações do MRUV e suas possíveis aplicações na vida cotidiana. 	<ul style="list-style-type: none"> • Aprender a importar bibliotecas, módulos ou frameworks; • Criar objetos 3D com a biblioteca VPython; • Compor funções em python relacionando com MRUV; • Aprender o conceito e a aplicação de laços (<i>loops</i>) na programação relacionando com animações em VPython e sua variação temporal;
4º Encontro (2º MP)	Aprender as características do fenômeno lançamento oblíquo, correlacionando com decomposição dos movimentos e animações digitais em Python, inseridos na temática do lançamento perfeito no jogo de basquete.	<ul style="list-style-type: none"> • Relacionar o uso de gráficos em Física com a interpretação dos movimentos que eles representam; • Aprender o conceito e dedução dos movimentos em: queda livre, em duas dimensões e o movimento balístico. 	<ul style="list-style-type: none"> • Relacionar o conceito de estruturas condicionais (teste <i>if</i> e <i>else</i>) a situações Físicas reais; • Ensinar a criação de gráficos com a biblioteca VPython e unir simulações Físicas.
5º Encontro (3º MP)	Apresentar em grupo as resoluções das situações problemas propostas em aula pelo professor e debater os conhecimentos de Física aliados à tecnologia e suas implicações, a fim de compreender se o	Não se aplica a este encontro.	Não se aplica a este encontro.

	aprendizado foi potencializado pela estratégia de ensino proposta na metodologia do curso FIS_PY.		
--	---	--	--

Fonte: Elaborado pelo autor.

Sendo assim, de forma mais detalhada, no primeiro encontro o pesquisador professor orientou os alunos a como utilizar o AVA Moodle e apresentou o cronograma da sequência didática. Também foi ensinado como realizar a instalação da linguagem de programação Python e do Software PyCharm, assim como cadastrar e utilizar a IDE online GlowScript. Nesse mesmo encontro, o pesquisador professor também iniciou a problematização sobre o conceito de cinemática ao expor questões (Apêndice I) que fazem parte do cotidiano dos alunos para então debater possíveis aplicações.

Dentre as questões abordadas durante esse encontro pelo pesquisador professor a fim de introduzir o assunto de cinemática, temos por exemplo: Quem aqui já chegou atrasado na escola? E por que vocês chegaram atrasados? Como vocês fariam para calcular o tempo mínimo para chegar na escola sem ultrapassar a velocidade permitida? A partir da problematização da velocidade máxima permitida, também foi abordado o assunto sobre as tecnologias que estão inseridas no trânsito, como semáforos e radares. Para que pudessemos relacionar a ideia da comunicação entre máquinas, regras e condições além da programação das máquinas.

No segundo encontro foram abordados pelo pesquisador professor os conhecimentos de Física e de programação fundamentais para resolução das situações reais exploradas no primeiro encontro. Desta forma, foi discutido nesse encontro inicialmente o conteúdo sobre medição de grandezas escalares, sistema internacional de medidas, mudança de unidade, comprimento, tempo e massa. Posteriormente, foram utilizadas essas mesmas unidades de medida para apresentar os conteúdos de programação sobre variáveis, atribuição de valores a variáveis, boas práticas para dar nome às variáveis, expressões e instruções básicas a serem realizadas no terminal e no modo *script*. Posteriormente a essa explicação, foram apresentados pelo pesquisador professor os conhecimentos físicos sobre posição e deslocamento de uma partícula, velocidade média, velocidade instantânea e aceleração, assim como suas respectivas funções horárias, conforme conteúdo exposto no subtópico 1.4.1. Com esses conhecimentos apresentados, o pesquisador professor pôde utilizar os mesmos exemplos

para explorar os seguintes conhecimentos em linguagem de programação Python: ordem das operações matemáticas, operações com *strings* e comentários, função, sem parâmetros, com parâmetros, com parâmetros mais instrução retorno e, por fim, como implementar as funções no *script*.

No terceiro encontro, o pesquisador professor iniciou o conteúdo de Física descrito no subtópico 1.4.2 a partir da problematização de uma viagem de passeio à Itália para ver a Torre de Pisa (apêndice I - slide aula 02). A partir disso, foi trabalhada a importação da biblioteca *math*, com o intuito de calcular as dimensões da Torre de Pisa e aprender como se importa biblioteca e módulos. Logo após, foi abordado a criação e configuração dos atributos e características de objeto 3D por meio da biblioteca VPython. Em seguida, para aplicar os conceitos de MRU, foi introduzido o conteúdo sobre laços na programação conforme exposto no subtópico 1.5.4 para ilustrar o deslocamento de uma partícula (apêndice I - slides aula 03 - slide 7). Ao final desse encontro, foi abordado o conceito de MRUV discutido no subtópico 1.4.2, em que o pesquisador professor deduziu as equações espaciais com os alunos e apresentou uma aplicação do conceito de aceleração ao laço na programação a fim de fazer o objeto 3D ganhar velocidade (apêndice I - slide aula 03 - slide 10).

Em seguida, no quarto encontro, foi levantado o assunto sobre o arremesso em um jogo de basquete e como a física poderia ajudar a descobrir o arremesso perfeito. Para tal, foi discutido com os alunos a aplicação dos conhecimentos adquiridos sobre MRU e MRUV no fenômeno do arremesso de uma bola de basquete. A partir da contextualização do movimento balístico (subtópico 1.4.3) e como podemos decompor o movimento no eixo horizontal e vertical de forma independente, foi abordado primeiramente a decomposição do movimento no eixo horizontal por meio da comparação com um corpo em queda livre. Com esse exemplo, também abordou-se o conteúdo sobre condições *if* e *else*, para que o objeto em queda não ultrapasse o objeto 3D que simboliza o chão na animação em VPython (apêndice I - slide aula 04 - slide 11). Também realizou-se a união do movimento na vertical e na horizontal a fim de apresentar um gráfico por meio da utilização da biblioteca VPython que descreva o movimento da partícula (apêndice I - slide aula 04 - slide 13).

Ao final do quarto encontro, o pesquisador professor solicitou aos alunos que eles se dividissem em dois grupos para desenvolverem uma simulação em Python que demonstrasse alguma aplicação dos conceitos que foram aprendidos durante as aulas. Para isso o pesquisador professor também abordou alguns exemplos de situações reais que poderiam ser solucionadas ou ilustradas pelos alunos mediante a aplicação dos conhecimentos físicos e de

linguagem de programação aprendidos até o quarto encontro, para ser apresentado na atividade final de criação de código do quinto encontro, como por exemplo: corrida de carro, lançamento de uma bola de canhão, espaço de segurança entre carros para frenagem segura, entre outros.

Por fim, o quinto encontro finalizou a sequência didática, com o intuito de que os estudantes pudessem apresentar as simulações aplicadas em situações problemas apresentados no quarto encontro mediante o debate dos conhecimentos de Física aliados às aplicações da linguagem de programação Python, para resolver esses problemas e potencializar o aprendizado. O objetivo deste encontro foi também ouvir os estudantes e poder constituir evidências que demonstrassem se o aprendizado foi ou não potencializado pelo uso da linguagem de programação Python.

3. METODOLOGIA DE PESQUISA

A metodologia científica não é simplesmente um conjunto de regras, mas sim uma lente que nos permite ver o mundo de uma nova maneira, de forma científica, indagadora e curiosa. Sendo que nas pesquisas qualitativas, a maior preocupação não é a representatividade numérica do grupo pesquisado, mas sim, a profundidade da compreensão de um grupo social mediante as características e relações que se deseja pesquisar (Goldenberg, 2004).

A forma de compreender a realidade pela imersão no mundo de outros indivíduos é certamente uma prática que busca compreender os fenômenos sociais, simplesmente definida como abordagem qualitativa, que fundamenta-se no princípio de que a realidade só existe a partir da visão da pessoa, ou seja, que a realidade é a interpretação que o indivíduo faz de um fenômeno (Malheiros, 2011).

E o fenômeno educacional trata de um universo multifacetado e complexo que envolve as relações de como o indivíduo se relaciona entre seus iguais e com o mundo. Para dar suporte a esse quadro educacional, segundo Panozzo (2013), a pesquisa qualitativa é excepcional, pois traz subsídios diferenciados para entender como crianças, jovens e adultos se formam, além de propor de forma investigativa melhorias para essa formação.

Por conta disso, escolhemos nesta pesquisa utilizar uma abordagem qualitativa, a qual norteou a forma pela qual o pesquisador professor enxerga o fenômeno, assim como a metodologia de pesquisa, constituição de dados e análise de dados que foram feitas. Ademais, foi utilizado o estudo de caso como metodologia de pesquisa, a qual será apresentada ainda neste capítulo, como também a unidade de análise, os instrumentos para constituição de dados e a metodologia de análise de dados.

Além disso, por conta desta pesquisa envolver seres humanos, ela foi previamente submetida para apreciação ética do Comitê de Ética em Pesquisa (CEP) e pela Conep seguindo as diretrizes da Plataforma Brasil⁸. Após análise do CEP a pesquisa foi aprovada para ser desenvolvida e pode ser consultado seu projeto pelo número CAAE 67749423.0.0000.5504.

⁸ Plataforma Brasil: <https://plataformabrasil.saude.gov.br/>

3.1 Estudo de Caso

Originalmente utilizado em medicina e em psicologia, o estudo de caso se tornou uma das principais metodologias em pesquisas qualitativas em ciências sociais. O estudo de caso é uma análise holística que considera a unidade a ser estudada como um todo, seja ela um único indivíduo ou uma organização social, com o objetivo de compreendê-la em seus próprios termos (Goldenberg, 2004).

Além disso, como na fundamentação teórica, o tema que está sendo pesquisado que corresponde a junção de ensino de Física com o auxílio de linguagem de programação aliada a alguma metodologia de ensino é inovador e carece de estudos, sendo este mais um atributo de impacto para utilização do estudo de caso como metodologia de pesquisa. Uma vez que o estudo de caso busca, de forma empírica, ampla, profunda e investigativa a relação do contexto e do fenômeno caso os limites não estejam estritamente claros, onde pode existir múltiplas variáveis e fontes a serem evidenciadas (Yin, 2015).

Isso beneficia esta pesquisa, visto que a carência de estudo deste tema turva a visão que temos da fronteira do caso que desejamos estudar, logo a metodologia de estudo de caso nos auxiliará a entender as particularidades do fenômeno, correlacionando as diversas variáveis que ele pode possuir.

Nas pesquisas de estudo de caso, geralmente é realizado um projeto piloto, ou seja, é aplicada uma pesquisa preliminar com objetivo de mitigar possíveis desvios e falhas em uma amostragem pequena, para fortalecer e consolidar as bases da pesquisa principal (Yin, 2015). Todavia, por conta do curto período de tempo disponibilizado para aplicação da pesquisa, mediante a duração que o mestrado permite e a flexibilidade de tempo do pesquisador professor, foi proposta nesta pesquisa a aplicação de um caso único que, segundo Yin (2015), não deixa de ser importante e de trazer *insights* inestimáveis, assim como dados suficientes para serem analisados – basta que o pesquisador siga o rigor estabelecido pelo método.

Além disso, este estudo de caso possui um caráter *revelador*, visto que o tema da pesquisa, além de ser inovador e possuir uma carência de estudo na área, permitirá analisar um fenômeno previamente não investigado ou descrito na bibliografia atual.

Desta maneira, conforme descrito por Yin (2015) em seu livro “Estudo de Caso: Planejamento e Métodos”, o estudo de caso precisa contemplar as seguintes etapas: (I) formulação do problema, descrito nesta pesquisa na parte introdutória; (II) definição da unidade de caso ou unidade de análise, como apresentado no capítulo de metodologia da

pesquisa deste trabalho, que consiste basicamente nos critérios de seleção do caso e do sujeito da pesquisa; (III) definição do número de casos, como supracitado o estudo de caso pode ser aplicado em um único caso, como o desta pesquisa, ou em múltiplos casos; (IV) elaboração do protocolo de constituição dos dados, responsável por aumentar a confiabilidade da constituição dos dados da pesquisa e orientar a conduta do pesquisador, sendo que nesta pesquisa ele é representado pela *sequência didática e estrutura das aulas*; (V) coleta de evidências ou constituição de dados, etapa que promove a obtenção de dados que garantirá a confiança e a qualidade da pesquisa; (VI) análise das evidências do estudo de caso ou simplesmente análise de dados, que consiste em organizar, padronizar, categorizar e atribuir *insights* mediante a metodologia de análise de dados escolhida; e, por fim, a última etapa é a (VII) consolidação do relatório do estudo de caso, que consiste na confecção do trabalho completo, composto por todas as etapas anteriores e permitindo compartilhar todos os dados, resultados e procedimentos da pesquisa.

3.2 Unidade de análise

O grupo pesquisado ou a unidade social, como descreve Goldenberg (2004), foram os alunos do ensino médio de uma escola técnica do interior do estado de São Paulo. A escolha da referida escola se deu pela facilidade de comunicação do pesquisador professor com a direção escolar e com a comunidade docente. Os alunos que se interessaram em participar desta pesquisa se inscreveram por meio de um formulário eletrônico. Estes alunos são adolescentes que compreendem a faixa etária dos 14 aos 18 anos, conforme apresentado na Tabela 1 referente aos inscritos.

Tabela 1 - Quadro de alunos inscritos

Aluno	Série do E.M.	Idade	Sexo (M/F)
A	3º ano	18	M
B	3º ano	17	F
C	2º ano	16	F
D	3º ano	17	M
E	1º ano	15	M
F	1º ano	15	M
G	3º ano	17	M
H	1º ano	15	M

Fonte: Feito pelo autor.

Para convidar os alunos a participar desta pesquisa, o pesquisador professor fez uma visita presencial na escola, na qual foi exposto o objetivo da pesquisa, assim como foi aberto o convite a todos os alunos interessados.

Para a inscrição, foi disponibilizado um QR Code vinculado a um formulário eletrônico do Google Forms que ficou exposto nos murais da escola para ser acessado pelos alunos que tivessem interesse em participar. Após o período de inscrição, o pesquisador professor retornou à escola para entregar aos alunos participantes o Termo de Assentimento de Menor, o Termo de Cessão para uso de Imagem e Voz e o Termo de Consentimento Livre e Esclarecido (TCLE), que foram assinados pelos pais dos participantes menores de idade, para que ficassem cientes dos riscos e benefícios da pesquisa e que seus filhos poderiam optar por deixar a pesquisa em qualquer momento.

3.3 Instrumentos para constituição de dados

As evidências em um estudo de caso podem vir de diversas fontes, e as mais usuais são: documentação, registros em arquivos, entrevistas, observação direta, observação participante e artefatos físicos. Sendo possível utilizar todas essas fontes em um único estudo, tendo o cuidado de não negligenciar os quatro princípios da coleta de dados: uso de múltiplas fontes, criação de banco de dados para a pesquisa, manutenção e encadeamento das evidências e cuidado com fontes eletrônicas ou de mídia social (Yin, 2015).

Nesta seção, apresentaremos as evidências que constituíram o estudo de caso, começando com os dados e registros decorrentes das aulas síncronas remotas realizadas através da ferramenta Google Meet. Esses dados foram originalmente arquivos de mídia em formato de vídeo gravados automaticamente pela ferramenta Google Meet e salvo no google drive restrito ao pesquisador professor, e para realizar sua análise foi realizada a transcrição do áudio das aulas remotas gerando assim um arquivo documental. Além disso, o Google Meet também permite o envio de mensagens no *chat box*, as quais também foram armazenadas, complementando o registro documental das aulas remotas.

Aliada à sequência didática, a pesquisa também propôs aos alunos a resolução de diversos exercícios. Estas resoluções foram compostas por atividades autocorrigíveis tanto pelo *plugin* CodeRunner discutido na seção deste trabalho, como também por questionários, que constituíram os registros de dados do tipo levantamento. Além dessas questões, também foram propostos exercícios abertos para criação de códigos que buscam resolver um problema

proposto relacionado aos temas das aulas, como por exemplo a criação de funções que indicam a velocidade limite que um carro pode transitar em vias públicas, finalizando assim a constituição dos dados documentais.

Por fim, foi aplicada uma entrevista aberta, ou seja, um questionário em que os alunos tinham total liberdade para expressar suas opiniões sobre o curso e o desenvolvimento do seu aprendizado, para que isso fosse realizado, foi aberto pelo pesquisador professor uma atividade de questionário discursivo no Moodle. Em resumo, esta pesquisa conta com seis tipos de fontes, sendo elas três documentais, duas do tipo registro e uma fonte de entrevista.

3.4 Metodologia de análise de dados

Neste tópico pretende-se apresentar a estratégia geral que norteou a análise dos dados desse estudo de caso, assim como a metodologia utilizada. Segundo Yin (2015), muitos pesquisadores de estudo de caso não dão o devido valor ao desenvolvimento da estratégia utilizada na análise dos dados, e acabam sofrendo no estágio analítico da pesquisa.

Para que isso não ocorra, foi escolhido para esta pesquisa uma metodologia de análise que trata os dados “a partir do zero”, pois como apresentado na fundamentação teórica foi identificado uma escassez de referencial, o que dificulta basear-se em proposições teóricas. Além de utilizar um modelo lógico organizacional, que busca investigar os padrões de causa-efeito-causa-efeito (Yin, 2015).

Essa escolha estratégica conecta a natureza reveladora e investigativa desta pesquisa com a busca a partir dos dados brutos constituídos, a fim de evidenciar padrões lógicos que indiquem hipóteses e resultados sobre a problemática desta pesquisa, sendo elas as potencialidades que a linguagem de programação Python aliada a uma sequência didática baseada nos três momentos pedagógicos (3MP) pode trazer para o ensino e aprendizagem dos conhecimentos de Física.

Sendo assim, para analisar os dados documentais, escolhemos a metodologia de análise de conteúdo, pois, quando bem aplicada, suas técnicas são capazes de encontrar padrões lógicos que podem ser traduzidos em categorias, mesmo em dados brutos com ou sem proposições teóricas (Mozzato; Grzybovski, 2011).

Por fim, para compor o início da análise optamos por apresentar um quadro geral do cenário em que o curso FIS_PY ocorreu, apresentando dados quantitativos que conferem a

quantidade dos alunos inscritos, frequência dos alunos, resolução das atividades, visualizações dos recursos e das atividades do curso e também o índice de acerto de cada questão.

3.4.1 Análise de Conteúdo

A metodologia de análise de conteúdo se originou nos Estados Unidos para analisar as comunicações e discursos realizados principalmente por políticos, a fim de quantificar e qualificar estudos empíricos apoiados na utilização de técnicas de classificação e categorização, que passou por vários aperfeiçoamentos durante o séculos XX, no intuito de levantar hipóteses sobre a propensão heurística de um determinado material, discurso, texto entre outros materiais (Bardin, 2002).

Com isso a análise de conteúdo ganhou destaque pelo fato dela constituir uma metodologia de análise:

Usada para descrever e interpretar o conteúdo de toda classe de documentos e textos. Essa análise, conduzindo a descrições sistemáticas, qualitativas ou quantitativas, ajuda a reinterpretar as mensagens e a atingir uma compreensão de seus significados num nível que vai além de uma leitura comum (Moraes, 1999, p. 2).

Com isso, foi escolhida a análise de conteúdo de Bardin (2002), pois entendemos que por ela ser capaz de analisar os mais diversos conteúdos de comunicação, se torna aplicável tanto no material de diálogo que será redigido neste trabalho, como nos códigos escritos na linguagem de programação Python, que é de alto nível, ou seja, próxima à comunicação humana. Logo, podemos utilizar a análise de conteúdo para interpretar o código que foi criado pelos alunos, uma vez que esta é capaz de capturar as expressões contidas em conteúdos criados pelos indivíduos. Partindo dessa lógica, os códigos criados pelos alunos são conteúdos que podem ser categorizados nesta pesquisa que revelam padrões, resultados e hipóteses.

Para que isso ocorra, a análise de conteúdo é organizada em fases em torno de três pólos cronológicos: (I) a pré-análise, (II) a exploração do material e (III) o tratamento dos resultados, a inferência e a interpretação (Bardin, 2002).

A (I) pré-análise tem o objetivo de sistematizar as ideias iniciais, a fim promover um desenvolvimento claro e operacional para garantir o decurso conciso da análise e também permitir que novos processos possam ser incorporados de forma harmoniosa. Além disso, a

pré-análise possui três tarefas iniciais, a escolha dos documentos, a formulação das hipóteses e dos objetivos e a elaboração de indicadores que fundamentam a interpretação.

Essas tarefas não ocorrem obrigatoriamente em ordem cronológica, mas estão intimamente ligadas, sendo que inicialmente a exploração sistemática do conteúdo é realizada por uma leitura flutuante dos documentos, com o objetivo de conhecer o conteúdo e permitir invadir-se de primeiras impressões que busca aos poucos hipóteses emergentes.

Adiante, a escolha dos registros desta pesquisa foi feita a priori constituindo os seguintes documentos já mencionados, que foram os dados redigidos decorrentes das aulas remotas: mensagens em *chat box*, atividades autocorrigíveis (questões feitas pelo plugin CodeRunner, questões de arrastar e soltar sobre o texto, questões de múltipla escolha e entrega de tarefa), entrevistas aberta e os registros dos códigos fontes das atividades problemas propostas aos alunos.

A formulação das hipóteses e dos objetivos consiste em recorrer a afirmações provisórias que norteiam a busca do objetivo da pesquisa. Sendo assim, a hipótese inicial é que o uso de linguagem de programação Python organizada em uma sequência didática baseada nos 3MP, possa trazer benefícios significativos para o desenvolvimento do ensino e aprendizado de Física, a fim de atingir o objetivo principal da pesquisa que é entender as potencialidades que a linguagem de programação Python poder implicar no desenvolvimento do ensino e aprendizagem de Física.

Mesmo que esta pesquisa seja de caráter posteriori para a construção da categorias do trabalho, a referenciação dos índices ou indicadores de narrativas também são importantes (BARDIN, 2002) para identificar padrões e fazer recortes de conteúdos mais precisos. Por conta disso, nesta pesquisa utilizamos como indicadores iniciais padrões que indiquem, de forma geral nos registros documentais, repetições de frases e argumentos de natureza científica ou pessoal que indique possível construção de conhecimento, assim como padrões de código que apresentem clareza em sua construção e efetividade na resolução de problemas. Além disso, buscamos padrões que apresentem, ou não, marcas sobre a importância e impacto do uso da programação aliada aos 3MP no ensino de Física.

Assim como indicado pela metodologia de estudo de caso, a análise de conteúdo também requer uma preparação do material. Por conta disso, se fez necessário criar uma base de dados para constituir a transcrição das gravações das aulas remotas, assim como dos exercícios, tarefas e entrevistas disponibilizados na sequência didática. As gravações das aulas

remotas foram armazenadas no Google Drive privado do pesquisador professor e os demais dados no Ambiente de Aprendizagem Virtual (AVA) da UFSCar (moodle). Após a finalização da sequência didática, demos início à transcrição dos dados a partir de um *script* criado na linguagem de programação Python pelo pesquisador professor (Apêndice C), o qual utiliza a biblioteca *Whisper* (<https://openai.com/research/whisper>), que consiste em um conjunto de ferramentas com o objetivo de transcrever áudios por meio de uma inteligência artificial. Essas transcrições foram conferidas pessoalmente pelo pesquisador professor para confirmação das informações e garantia da fidelidade dos dados.

Após a pré-análise inicia-se a (II) exploração do material, uma etapa de aprofundamento do conteúdo da pesquisa, que busca penetrar ainda mais no material da pesquisa e refinar o gerenciamento das ideias, consistindo em uma fase

De procedimentos aplicados manualmente ou de operações efetuadas pelo ordenador, o decorrer do programa completa-se mecanicamente. Essa fase longa e fastidiosa consiste essencialmente de operações de codificação, desconto ou enumeração, em função de regras previamente formuladas (Bardin, 2002 p. 101).

Nesta etapa, iniciamos a leitura dos dados transcritos, explorando os significados e buscando estabelecer relações entre os textos. Assim, durante a exploração do material, foi necessário aplicar o desenvolvimento da análise inicial da fase de codificação. Segundo Bardin (2002), a codificação corresponde à transformação por intermédio de regras precisas dos dados brutos. Ou seja,

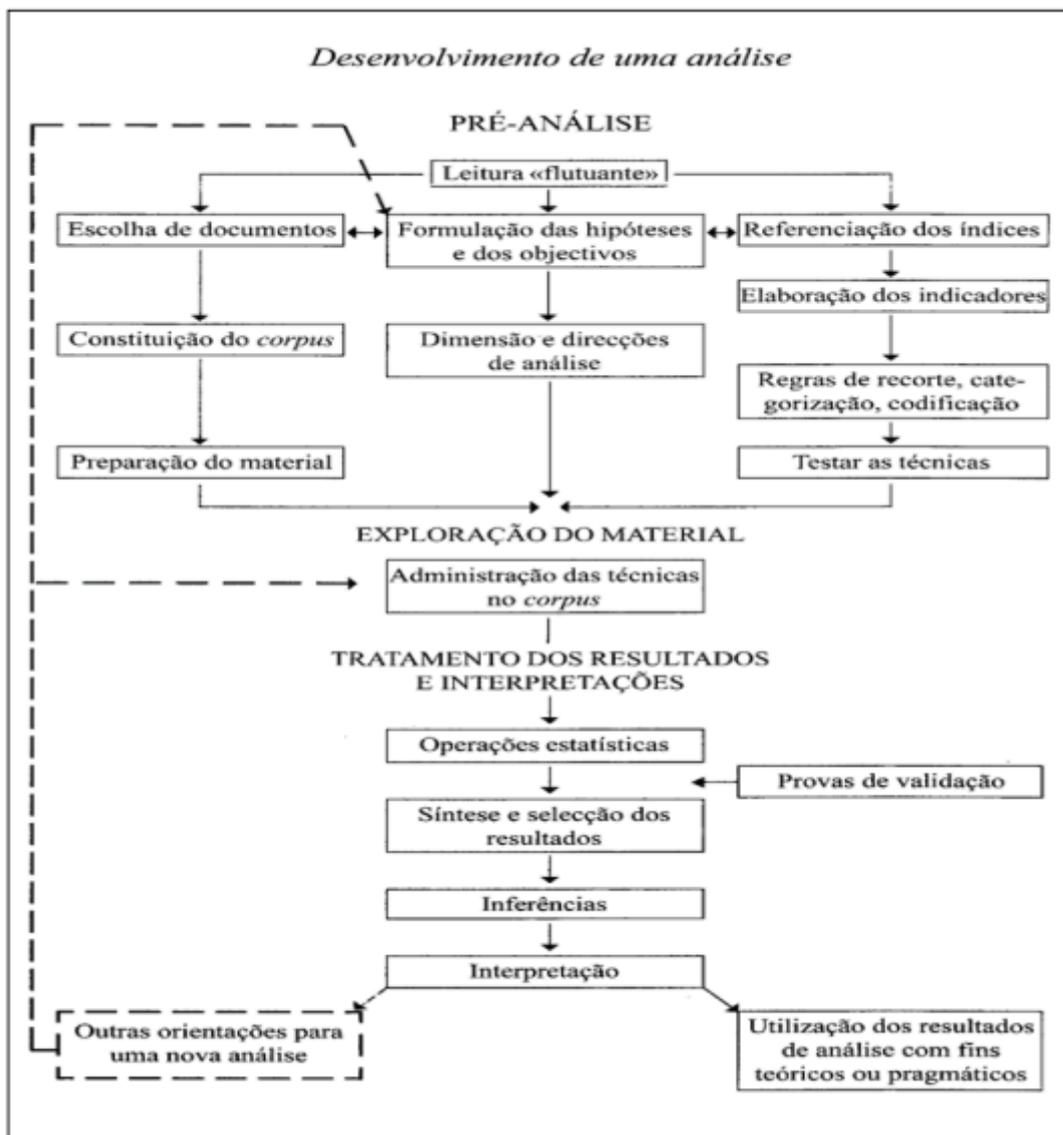
A codificação é o processo pelo qual os dados brutos são transformados sistematicamente e agregados em unidades, as quais permitem uma descrição exata das características pertinentes do conteúdo (Holsti, 1969 apud Bardin, 2011, p. 103).

Desta forma, a organização da codificação compreende três escolhas (no caso de uma análise quantitativa e categorial): (I) o recorte, designa a escolha das unidades, (II) a enumeração, aplica-se a escolha das regras de contagem e a (III) a classificação e agregação da escolha das categorias.

Por fim, a etapa final de (III) tratamento dos resultados obtidos e interpretados, tem o objetivo de dar voz aos dados brutos de forma significativa e válida com o teor da pesquisa, trazendo informações estatísticas simples ou complexas de forma gráfica, tabelada ou por meio de diagramas que forneçam informações que condensam a análise dos dados.

Conceitualmente, as etapas da análise de conteúdo podem ser visualizadas de forma mais sintetizada na figura 13:

Figura 15 - Mapa conceitual sobre o desenvolvimento da análise de conteúdo.



Fonte: Bardin (2002, p. 102).

4. ANÁLISE DOS DADOS

Este capítulo é responsável por apresentar a análise de dados, a qual desempenha um papel importante na exploração do tema da pesquisa sobre o uso da linguagem de programação Python no ensino de Física aliada aos Três Momentos Pedagógicos. Em resumo, a análise dos dados coletados pode fornecer insights cruciais sobre como a linguagem de programação Python influenciou o processo de ensino e aprendizagem, bem como sua relação com cada um dos 3MP.

Para isso, inicia-se esse capítulo a partir de uma apresentação global dos dados quantitativos da pesquisa, a fim de demonstrar o desenvolvimento da pesquisa ao longo da aplicação do curso FIS_PY. Desta forma, o início desse capítulo é composto por: frequência dos alunos, entrega de exercícios, visualizações de atividades e recursos além do índice de acerto das questões. Posteriormente, como já descrito na metodologia, será abordada a análise de conteúdo com o objetivo de responder às questões de pesquisa e avaliar o impacto do uso da linguagem de programação Python no contexto do ensino de Física.

A análise foi guiada pelos objetivos da pesquisa, que visam aprofundar nosso entendimento sobre o ensino de Física com a linguagem de programação Python relacionado aos 3MP, e também oferecer *insights* práticos para educadores, formuladores de políticas educacionais e pesquisadores da área. Ao abordar essas questões, buscamos destacar padrões que demonstrem argumentos de natureza científica ou pessoal que indiquem possível construção de conhecimento, assim como padrões de código que apresentem clareza em sua construção e efetividade na resolução de problemas, visando contribuições aos desafios do uso da tecnologia de programação no contexto educacional, bem como identificar estratégias que podem ser aprimoradas para melhorar a eficácia do ensino de Física.

Por fim, para conduzir essa análise, recorreremos ao método da análise de conteúdo (Bardin, 2002) como apresentado na seção tal e iniciaremos a análise a partir da apresentação de dados quantitativos sobre o curso FIS_PY.

4.1 Frequência dos Alunos

Após a divulgação da pesquisa na escola parceira, se inscreveram no curso FIS_PY doze alunos, os quais estão representados nesta pesquisa por meio das letras do alfabeto para

garantir o anonimato dos mesmos. Desta forma, pode-se observar na Tabela 2 a frequência dos alunos nas aulas, a disposição das séries do ensino médio de cada um e também seu nome simbólico.

Tabela 2 - Frequência dos alunos

Aluno	Série do E.M.	Encontro 1	Encontro 2	Encontro 3	Encontro 4	Encontro 5
A	3° ano					
B	3° ano					
C	2° ano					
D	3° ano					
E	1° ano					
F	1° ano					
G	3° ano					
H	1° ano					
Legenda		Presença				
		Ausência				

Fonte: Elaborado pelo autor.

A respeito das inscrições por série do ensino médio, a terceira série compreende 4 alunos do total dos inscritos, ou seja, o maior número de interessados em participar desta pesquisa. Enquanto isso, houve apenas 1 aluno inscrito que estava na segunda série do EM e 3 alunos do primeiro ano do EM que se interessaram em participar da pesquisa, totalizando 8 alunos.

Para efeitos de se alcançar os objetivos desta pesquisa, o pesquisador professor optou por não buscar as justificativas dos alunos que desistiram do curso, por conta desta pesquisa não ter um caráter explorador sobre o tema de evasão escolar ou de cursos, visto que o objetivo central de nossa pesquisa é entender quais as potencialidade do uso da linguagem de programação Python no ensino de Física por meio dos 3MP, através dos encontros que foram realizados com alunos que permaneceram interessados no curso FIS_PY.

4.2 Atividade e recursos

Cabe explicar ao leitor, que os registros quantitativos constituídos neste subtópico são dados contabilizados automaticamente pelo AVA Moodle denominado como CodeGenius nesta pesquisa, que representam as atividades de questionário (questões de CodeRunner, múltipla escolha e arrastar e soltar) e tarefas, assim como os recursos (slides e videoaula) disponibilizados no curso FIS_PY. Esses registros gerais do curso apresentam a quantidade de visualizações por atividade ou recurso, que analisaremos inicialmente para detalhar as estatísticas de cada atividade.

Sobre as visualizações das atividades ou recursos, o Moodle reconhece como visualização ações do usuário como atualização da página, troca de questão, entrar e sair da atividade, verificação da resposta (para questões CodeRunner), abrir uma nova guia do navegador entre outras opções que se assemelhem a essas. Sendo assim, podemos observar no Quadro 5 a ocorrência das visualizações em cada encontro.

Quadro 5 - Quadro de ocorrência de visualizações de atividades e recursos

Atividade / Recurso	Encontro 01	Encontro 02	Encontro 03	Encontro 04	Encontro 05
Slides	16 visualizações por 5 alunos	17 visualizações por 5 alunos	8 visualizações por 4 alunos	13 visualizações por 4 alunos	NA
Vídeoaula	3 visualizações por 1 aluno	9 visualizações por 5 alunos	4 visualizações por 2 alunos	10 visualizações por 5 alunos	1 visualizações por 1 aluno
Atividades	N/A	568 visualizações por 7 alunos	312 visualizações por 6 alunos	474 visualizações por 6 alunos	157 visualizações por 6 alunos
NA - Não se Aplica					

Fonte: Elaborado pelo autor.

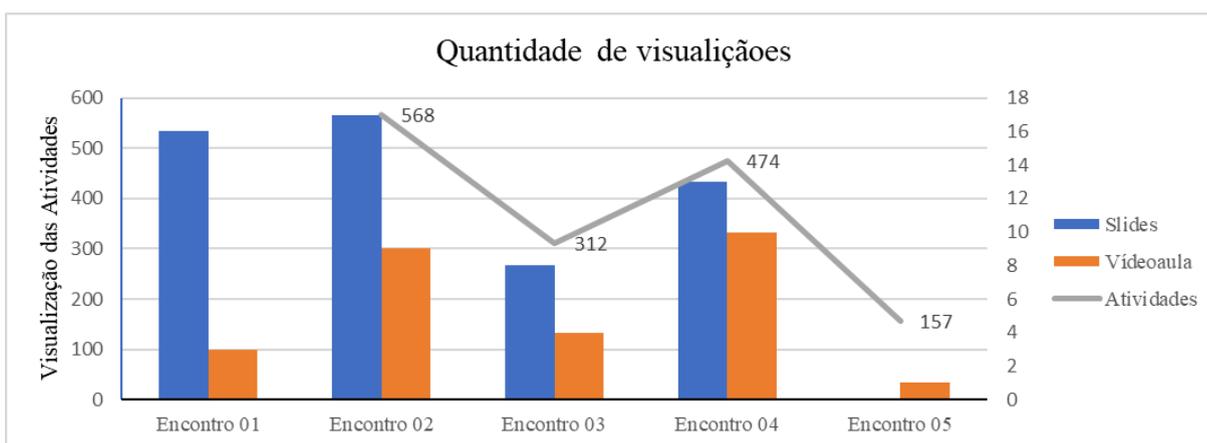
No primeiro encontro com os alunos não houve a aplicação de nenhuma atividade, visto que ele se referia ao momento de problematização do curso. De forma semelhante, no último encontro não houve apresentação de slides, uma vez que foi a etapa de aplicação dos conhecimentos onde conforme exposto no final do subtópico 2.4.1, os alunos apresentariam a aplicação de seus conhecimentos em alguma situação real. Com isso, para a atividade do encontro 01 e para os slides do encontro 05, aplica-se o conceito de NA (ou seja, Não se Aplica).

O curso apresentou uma média de 13,5 visualizações dos slides acessados por uma média de 4,5 alunos, em comparação com 6,5 visualizações das videoaulas gravadas que correspondem ao acesso em média por alunos de 3,25. Ou seja, o material em slides foi visualizado cerca de duas vezes mais que o material de consulta em videoaula.

Sobre as visualizações das atividades temos a ocorrência de 1511 visualizações realizadas por uma média de 6,5 alunos por encontro, o que corresponde aproximadamente ao total de alunos que finalizaram o curso. Além disso, foi constatada uma média de 378 visualizações por encontro dessas atividades. Levando em conta que a atividade do quinto encontro foi a entrega e apresentação de um programa que simulasse um fenômeno físico, temos um total de perguntas nas atividades, que compreendem o segundo, terceiro e quarto encontro, de 29 questões. Logo, temos uma média de 47 visualizações por questões, o que evidencia algum tipo de ação do estudante durante o processo de resolução das atividades.

Ademais, podemos evidenciar a curva de visualização das atividades e recursos no gráfico na figura 14, onde é destacado um percurso semelhante entre as curvas, reforçando ainda mais a correlação entre a quantidade de visualizações do material de consulta com o as ações realizadas para solucionar as atividades de cada encontro.

Figura 16 - Curva da quantidade de visualizações das atividades por encontro.



Fonte: Elaborado pelo autor.

4.2.1 Atividades Questionário

Como abordado anteriormente, a atividade questionário é uma forma de avaliação do AVA Moodle que permite a inclusão de diversos tipos de questões a serem respondidas e

avaliadas em um único local. Com isso, o curso FIS_PY fez o uso de três atividades em formato de questionário distribuído conforme tabela 2:

Tabela 3 - Distribuição dos tipos de questões por atividade e encontro.

Tipo de questões	Encontros		
	Atividade 01 / Encontro 02	Atividade 02 / Encontro 03	Atividade 03 / Encontro 04
CodeRunner	10	5	6
Arrastar e soltar	0	3	3
Múltipla escolha	0	2	0

Fonte: Elaborado pelo autor.

A escala de pontuação varia de 0 a 10 pontos para cada atividade questionário como estabelecido por padrão pelo Moodle, sendo distribuídos igualmente para cada questão, ou seja, uma questionário com 10 questões cada uma vale 1 (um) ponto, enquanto um questionário de 5 (cinco) questões, cada questão vale 2 pontos. Desta forma, pode-se observar a quantidade de acertos de cada aluno nas respectivas atividades que foram entregues.

Tabela 4 - Nota das atividades por aluno

	Aluno A	Aluno B	Aluno C	Aluno D	Aluno E	Aluno F	Média Atividades
Atividade 01	7	9	8	8	10	10	8,67
Atividade 02	6,6	10	7	7	10	8	8,10
Atividade 03	4,44	8,89	10	10	9,52	NE	8,57
Média Alunos	6,01	9,30	8,33	8,33	9,84	9,00	

NE - Não realizou a atividade

Fonte: Elaborado pelo autor.

Levando em conta apenas as atividades que foram feitas ou finalizadas pelos alunos, ou seja, que os alunos chegaram até o final do questionário online e apertaram o botão enviar, temos uma média de pontos de todas as atividades igual a 8,45. Como também, a turma de alunos apresentou uma média geral de 8,47 pontos por aluno. Usando a fórmula do desvio

padrão $\sigma = \sqrt{\sum(x - \bar{x})^2/N}$ temos uma variação de 1,22 pontos entre os alunos. De forma semelhante, entre a média geral do total de atividades, temos um ligeiro desvio de 0,25 pontos.

Os dados da Tabela 4 revelam que a menor média das atividades corresponde ao questionário 02 que continha os conteúdos de grandezas vetoriais, relações trigonométricas, Movimentos Retilíneo e Uniforme Variado (MRUV), uso da biblioteca VPython, composição de funções e criação de laços (loops). Enquanto isso, as outras duas atividades apresentaram uma média mais próxima entre si. Mesmo assim, para poder detalhar a pontuação de cada questionário e identificar a maior ocorrência de erros por questões, segue o resumo de cada questionário.

Tabela 5 - Detalhamento das questões corrigidas por atividades

Atividade 01 - Aula 02											
Nome	Média por Aluno	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
C	7	1	1	0	0	1	0	1	1	1	1
D	9	1	1	0	1	1	1	1	1	1	1
F	8	1	1	1	1	1	0	0	1	1	1
G	8	1	1	1	1	1	0	0	1	1	1
K	10	1	1	1	1	1	1	1	1	1	1
L	10	1	1	1	1	1	1	1	1	1	1
Média geral	8,67	1	1	0,67	0,83	1	0,5	0,67	1	1	1
Atividade 02 - Aula 03											
Nome	Média por Aluno	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
C	6,6	1	1	0	1	1	0	0,6	1	1	0
D	10	1	1	1	1	1	1	1	1	1	1
F	7	0	1	0	1	0	1	1	1	1	1
G	7	0	1	0	1	0	1	1	1	1	1
K	10	1	1	1	1	1	1	1	1	1	1
L	8	1	1	1	1	1	0	1	1	1	0
Média geral	8,1	0,67	1	0,5	1,00	0,67	0,67	0,93	1	1	0,67
Atividade 03 - Aula 04											

Nome	Média por Aluno	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
G	10	1,11	1,11	1,11	1,11	1,11	1,11	1,11	1,11	1,11
K	9,52	1,11	1,11	1,11	0,63	1,11	1,11	1,11	1,11	1,11
F	10	1,11	1,11	1,11	1,11	1,11	1,11	1,11	1,11	1,11
C	4,44	0	0	0	1,11	1,11	1,11	0	0	1,11
D	8,89	1,11	1,11	1,11	1,11	1,11	0	1,11	1,11	1,11
Média geral	8,57	0,89	0,89	0,89	1,02	1,11	0,89	0,89	0,89	1,11

Fonte: Elaborado pelo autor.

Ao aprofundar a análise sobre as atividades questionário inicialmente pelas questões com menor índice de acerto, podemos separar as questões que não tiveram 100% de acerto dos alunos em busca de identificar os possíveis erros, e compreender se esses erros foram provocados por questões conceituais, analíticas, operacional matemática ou oriunda da lógica de programação.

Desta forma, na Tabela 6 pode ser observado o índice de acerto de forma crescente das atividades, esse índice é calculado automaticamente pelo AVA Moodle, em que consiste em dividir o total de acertos por questão, pela quantidade de alunos que a realizaram, isto é, uma questão solucionada corretamente por todos os alunos do curso possui o índice de 100% igual a 1, enquanto uma questão que foi solucionada corretamente apenas por metade dos alunos possui o índice 0,5.

Tabela 6 - Ranking crescente de índice de acerto por questões

Atividade / Questão	Índice de acerto
A1Q6	0,5
A2Q3	0,5
A1Q3	0,67
A1Q7	0,67
A2Q1	0,67
A2Q5	0,67
A2Q6	0,67
A1Q4	0,83
A3Q1	0,89
A3Q2	0,89

A3Q3	0,89
A3Q6	0,89
A3Q7	0,89
A3Q8	0,89
A2Q7	0,93
A3Q4	1,00
A - Atividade	
Q - Questão	

Fonte: Elaborado pelo próprio autor

Segundo o ranking crescente da Tabela 6 as questões A1Q6 e A2Q3 apresentam o menor índice de acerto. Analisando as respostas incorretas da questão A1Q6 é possível compreender que todas as inconsistências são referentes a um erro de sintaxe, ou seja, uma falha na interpretação do código por conta de palavras-chaves, argumento e até mesmo pontuações estruturais. No caso da questão A1Q6, a falha se encontra na vírgula da casa decimal representada pelo número pi, visto que é padrão nas linguagem de programação que os números com os pontos flutuantes ou casas decimais sejam representados por ponto final (“3.14”).

Figura 17 - Exemplo de questão A1Q6 com erro no ponto flutuante da casa decimal do número pi

Crie uma função de nome `CalcularAreaDaEsfera`, que receba como parâmetro um número que representa o valor do raio da esfera em centímetros e retorne o valor da sua área superficial em centímetro, conforme exemplo abaixo (obs: Sendo π igual a 3,14 com apenas duas casas de tolerância)

```
In [ ]:CalcularAreaDaEsfera(4)
Out [ ]:200.96
```

Answer: (penalty regime: 0 %)

```
1 def CalcularAreaDaEsfera(raio):
2     area = 4 * 3,14 * (raio ** 2)
3     return area
```

	Test	Expected	Got	
✘	<code>print(CalcularAreaDaEsfera(4))</code>	200.96	(12, 224)	✘

Run using the University of Canterbury's Jobe server. This is for initial testing only. Please set up your own Jobe server as soon as possible. See here.

Show differences

Fonte: Elaborado pelo autor.

A resolução da questão A2Q3 feita pelos alunos, também apresentou erro de sintaxe nas três respostas incorretas, oriunda da escrita inconsistente das variáveis, visto que a linguagem de programação Python aceita apenas variáveis que contemplem nomes sem espaço, e os alunos incluíram nomes de variáveis internas das função com espaço, e em uma dessas questões incorretas também houve um erro de indentação, ou seja, na construção da resposta faltou incluir a tabulação necessária para o conteúdo do programa ser inserido na função.

Figura 18 - Exemplo de questão A2Q3 de sintaxe.

Crie uma função de nome `CalcularDiferancaVelocidadeEntrePontos` que receba como parâmetros dois valores numéricos que representem a velocidade do móvel no ponto inicial e ponto final, e retorne o valor da diferença de velocidade entre os pontos em m/s com uma casa decimal se houver necessidade, conforme exemplo abaixo.

```
In [ ]: CalcularDiferancaVelocidadeEntrePontos(90, 100)
Out[ ]: 10.0
```

Answer: (penalty regime: 0 %)

```
1 def CalcularDiferancaVelocidadeEntrePontos(velocidade inicial, veloci
2     velocidade = velocidade final - velocidade inicial
3     return velocidade
```

Syntax Error(s)

File "__tester__.python3", line 7
 def CalcularDiferancaVelocidadeEntrePontos(velocidade inicial, velocidade final)
 ^^^^^^^

SyntaxError: invalid syntax

► Show/hide question author's solution (Python3)

Incorreto
 Notas para este envio: 0,00/1,00.

Fonte: Elaborado pelo autor.

Com isso, os erros no uso da linguagem de programação nas duas questões com menor índice de acerto foram causadas por inconsistências de sintaxe na linguagem de programação, o que indica um ponto relevante, uma vez que os programas construídos pelos alunos demonstram que os conceitos de programação e física foram compreendidos, todavia uma falha estrutural no código causou o erro em sua execução.

4.3 Análises e Resultados

Nesta seção, apresentamos a análise qualitativa dos dados constituídos durante a pesquisa, utilizando a metodologia de análise de conteúdo proposta por Bardin (2002). Para tal, faz-se necessários alguns esclarecimentos para o bom entendimento do leitor sobre a análise feita nesta pesquisa.

Como já descrito no capítulo de metodologia de pesquisa, as transcrições das aulas foram realizadas pelo pesquisador professor e armazenadas em um documento tipo *docx* no *google drive*, essas mesmas transcrições estão destacadas em *itálico* neste trabalho para melhor identificação do leitor. Além disso, como durante as aulas vários diálogos foram realizados por meio do *Chat Box*, foi necessário incluir as conversas em formato de mensagem de texto nas transcrições dos áudios para que os diálogos pudessem fazer sentido.

Desta forma, todos os diálogos iniciados com a abreviação **CB** (*Chat Box*) em negrito são oriundos das mensagens de texto. Além disso, como nesta pesquisa também foi registrado entrevista aberta, para melhor identificação desses dados foi utilizado a abreviação **EA** (entrevista aberta) em negrito. Vale ressaltar, que o nome dos alunos foi mantido em anonimato sendo substituído por letras.

Após a organização das transcrições das aulas, mensagens de textos e das entrevistas em uma base de dados foi possível iniciar a análise de conteúdo. E, a partir de uma leitura em busca de padrões que indiquem impactos sobre o uso da linguagem de programação Python como recursos potencializador ou não de ensino e aprendizagem em Física, foi possível estabelecer três categorias que estão sintetizadas no Quadro 6 e seus respectivo significados a fim de apresentar suas principais características.

Quadro 6 - Sintetização dos significados das categorias formuladas.

Categoria	Significados
Indícios de aprendizagem em Física e/ou programação	Categoria que agregou relatos e diálogos que apresentam indícios de aprendizagem dos fenômenos físicos trabalhados em aula, dos conceitos de programação ou ambos.
Problematização e/ou aplicação do assuntos abordados em aula	Categoria na qual constituímos relatos ou diálogos que demonstre que aluno soube problematizar o fenômenos físicos, conseguiu

	aplicar algum tipo de resolução com ou sem o auxílio de linguagem de programação.
Indicativos de potencialidade ou limitação do ensino de Física com programação	Categoria em que reunimos todos os relatos e diálogos que apresentam indícios que a linguagem de programação ajuda ou não no entendimento dos fenômenos físicos, assim como indícios de potencialidades e limitações.

Fonte: Elaborado pelo autor.

4.3.1 Indícios de aprendizagem em Física e/ou programação

Baseado na fundamentação teórica desta pesquisa (Milaré; Richetti, 2021; Siemsen, 2019; Rodrigues; Machado, 2023), foi possível entender que a ACT tem por objetivo promover o uso consciente dos conhecimentos científicos e tecnológicos, por meio de práticas de ensino e aprendizagem que se relacionam com outros tipos de conhecimento, a fim de atribuir sentido ao mundo em que vivemos a partir da linguagem científica e tecnológica, para que possamos tomar decisões claras que conseqüentemente tenham interferência nas vidas das pessoas nas esferas política, econômica e sociais.

Ao considerar que os conhecimentos científicos abordados nesta pesquisa são os de Física, mais especificamente sobre o conteúdo de cinemática, e que o recurso utilizado para compor a estratégia de ensino e aprendizagem é a linguagem de programação Python, buscou-se na transcrição das falas dos alunos, *chat box* e entrevistas, evidências que demonstrem a relação com a obtenção de dados, a estruturação do pensamento e suas relações, como proposto por Silva e Lorenzetti (2020), ao associar grupos de indicadores de AC que apresentem organização do conhecimento, raciocínio lógico, levantamento de hipóteses e explicações.

Com isso, criou-se a categoria de indícios de aprendizagem em Física e/ou programação, a fim de reunir evidências que apresentem se o aluno chegou ao aprendizado dos conceitos propostos na sequência didática. Desta forma, percebemos esses indicadores nas falas dos alunos destacadas nesse subtópico.

Aluno G: Primeiramente, a gente importou todas as bibliotecas. Então, a gente criou uma cena, um quadradinho, com 80 por 400 pixels. Com o nome que eu falei. O nome comparando alunos. E a gente criou todos os nossos objetos. O ponto de referência foi o corredor, que é o quadradão aqui, o retângulo azul. Que, por mais que seja uma simulação

em 3D, tá bem 2D isso aqui. É melhor vindo de lado mesmo. Depois, a gente criou a primeira bolinha, que é o aluno 1. E a segunda bolinha, que é o aluno 2. Aqui, a velocidade do aluno 1 é constante. De 1 metro por segundo. Ele não tem nenhum tipo de aceleração. E a velocidade do aluno 2 é zero. Ou seja, ele começa parado. Porém, ele tem aceleração. Então, imediatamente, ele já começa a ganhar velocidade. Ele tá bem devagarzinho. Daí, ele vai ficando mais rápido. Até que ele ganha velocidade suficiente para ultrapassar o aluno 1 bem no finalzinho, passar o aluno vermelho. Aqui, a gente fez algumas listas de alunos. Algumas listas para armazenar nos gráficos. Está aqui embaixo o código do gráfico. E aqui, a gente fez um loop. Pra finalizar quando o aluno 2, que é o mais rápido, chegasse ao fim do quadrado azul. Aqui, eu limitei o rate da simulação, porque, como eu disse, no Opera, laga bastante. E pra fazer funcionar certinho a velocidade dos dois. Uhum. E eu defini aqui a mudança de posição deles.

Aluno G: No caso do aluno 1, é mais simples. Porque, como eu disse, ele não tem aceleração. Mas, no aluno 2, ele já tem a aceleração. Então, a aceleração entra aqui e vai pra velocidade dele, que, conseqüentemente, aumenta a velocidade. Aqui estão os códigos para armazenar a posição deles, que vai ser usado para formar os gráficos aqui embaixo. E aqui tá o código do gráfico. Tem as curvas, que mostram certinho aqui a mudança de posição deles. Deixa eu terminar a simulação aqui pra aparecer o gráfico. Tá. Aqui embaixo. Como eu disse, o aluno vermelho, o aluno 1, ele tem velocidade constante. E o aluno 2 tem a aceleração constante. Então, a velocidade dele aumenta. É uma curvinha que vai subindo. E é basicamente isso, né? A nossa simulação. Alguma pergunta?

O trecho destacado fornece uma visão detalhada do próprio aluno no processo de criação e execução de uma simulação envolvendo física e programação feita pelo seu grupo. Sobre esses aspectos podemos identificar que o aluno aprendeu os conceitos de física, pois conseguiu descrever a criação de objetos físicos e suas variáveis intrínsecas, o que demonstra sua compreensão sobre a representação física desses elementos no contexto da simulação. Além disso, ele conseguiu discorrer sobre os conceitos de velocidade e aceleração, assim como a diferença dos movimentos MRU e MRUV, a partir da explicação das características de velocidade e aceleração dos alunos na simulação, o que indica a compreensão dos conceitos físicos e sua aplicação prática.

Sobre os conceitos de programação, o aluno soube explicar de forma lógica a construção do *script* da simulação feita pelo seu grupo. Além disso, houve a utilização de bibliotecas e criação de objetos, o que demonstrou o aprendizado sobre a importação de

bibliotecas, criação de objetos e definir suas propriedades, o que indica o aprendizado na linguagem de programação Python. Assim também, é possível perceber a utilização dos laços e condicionais para controlar o fluxo da simulação, o que evidencia a compreensão sobre estruturas de controle em programação.

Com isso, é possível perceber a aplicação prática dos conceitos Física, como velocidade e aceleração, com elementos de programação, como laços e condicionais, ao demonstrar a aplicação prática desses conhecimentos em ambas as áreas. Além disso, o aluno menciona a geração de gráficos para visualizar a mudança de posição das esferas que representam os alunos ao longo do tempo, o que indica a capacidade de analisar os dados gerados pela simulação.

Os aspectos destacados podem ser observados no código intitulado Corrida de Alunos criado pelo grupo de alunos.

```
Web VPython 3.2
from vpython import *

scene = canvas(width=800, height=400, title='Comparando Alunos')
corredor = box(pos=vector(0, 0, 0), size=vector(10, 0.2, 1),
color=color.blue)
aluno1 = sphere(pos=vector(-4.5, 0.4, 0), radius=0.2,
color=color.red)
aluno2 = sphere(pos=vector(-4.5, 0.8, 0), radius=0.2,
color=color.green)

velocidade_aluno1 = 1.0 # m/s (velocidade do primeiro aluno)
aceleracao_aluno1 = 0.0
velocidade_aluno2 = 0.0
aceleracao_aluno2 = 0.3 # m/s^2 (aceleração do segundo aluno)
tempo = 0.0
delta_tempo = 0.1

tempo_lista = []
posicao_aluno1_lista = []
posicao_aluno2_lista = []

while aluno2.pos.x < 4.5:
    rate(10)
    tempo += delta_tempo
    posicao_aluno1 = aluno1.pos.x + (velocidade_aluno1 *
```

```

delta_tempo)
    velocidade_aluno2 += aceleracao_aluno2 * delta_tempo
    posicao_aluno2 = aluno2.pos.x + (velocidade_aluno2 *
delta_tempo)

    aluno1.pos.x = posicao_aluno1
    aluno2.pos.x = posicao_aluno2

    tempo_lista.append(tempo)
    posicao_aluno1_lista.append(posicao_aluno1)
    posicao_aluno2_lista.append(posicao_aluno2)

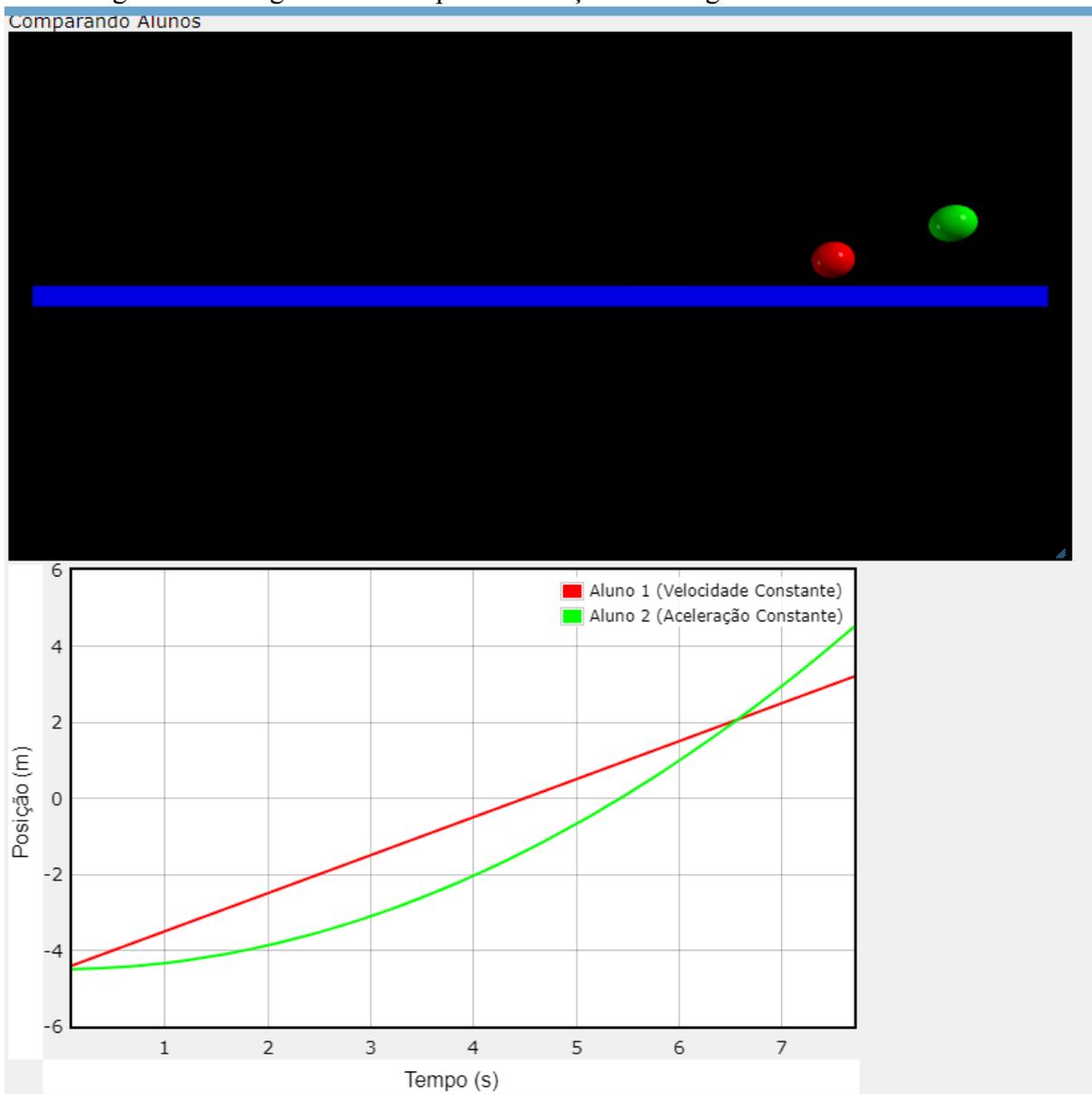
scene2 = canvas(width=800, height=400, title='Comparação de
Alunos')
posicao_tempo_grafico = graph(xtitle='Tempo (s)', ytitle='Posição
(m)')
curva1 = gcurve(graph=posicao_tempo_grafico, color=color.red,
label='Aluno 1 (Velocidade Constante)')
curva2 = gcurve(graph=posicao_tempo_grafico, color=color.green,
label='Aluno 2 (Aceleração Constante)')

for t, pos1, pos2 in zip(tempo_lista, posicao_aluno1_lista,
posicao_aluno2_lista):
    curva1.plot(t, pos1)
    curva2.plot(t, pos2)

```

Da mesma forma, podemos evidenciar a concretização do aprendizado ao observar o fenômeno físico sendo simulado de forma correta quando executamos o código na plataforma GlowScript e temos o seguinte resultado na figura 18.

Figura 19 - Imagem retirada após a execução do código Corrida de Alunos.



Fonte: Elaborado pelo autor

Não obstante, ainda temos relatos do segundo grupo de alunos que abordam a criação de suas simulações no segundo encontro.

Aluna B: A gente começou usando o from the Python Import, que é para puxar a biblioteca e assim conseguir fazer a montagem 3D. Depois disso, a gente fez o título do gráfico, né, para definir ele como uma posição horizontal pelo tempo. E configuramos os eixos X e Y, para dar o nome para eles. Depois disso vem a curva que pra criar essa curvinha no gráfico para demonstrar o lançamento oblíquo, o solo que é pra criar o objeto retangular que vai ser o arrastrinho da bola depois disso o teta pra definir o ângulo que a bolinha vai

fazer o g e que vai definir a aceleração da gravidade como 9,8 menos 9,8 porque aí nesse caso como ela tá caindo ela vai ser negativa né perfeito aí depois disso a gente vai definir a velocidade inicial da bola com o bola v com o cosseno sendo 20 e o seno sendo 0 se eu não me engano acho que é isso.

Aluna C: é aqui é porque tá em radianos define o ângulo inicial de 30 convertido para radianos pra ficar mais fácil e o tempo t é igual a 0, que inicializa as variáveis de tempo e o intervalo de tempo que é dt como 0.001 segundos e aí depois a gente vai pra legenda que é essa parte aqui ó.

Aluna C: aqui e ela cria uma posição de texto pra mostrar pra gente o quanto essa bolinha voou que foi 35.334 metros e aí a gente já vai pra parte mais falada que foi o loop que sempre foi falado durante as aulas e tal, o while que realiza a simulação do movimento da bola enquanto a posição vertical da bola, bola posição posição y for maior ou igual a 1 e aí depois a gente vai pro rate que é 500 e isso a Marina vai explicar.

Aluna B: esse rate é pra definir a atualização da animação para 500 quadros por segundo né pra dar a diferencinha.

Aluna C: e a bola v e essa eu não sou bom em fórmula essa fórmula aqui ela atualiza a velocidade da bola devido a aceleração da gravidade e a posição horizontal da bola com base na velocidade e aqui como você está vendo, que é o t igual a t mais dt atualiza o tempo da nossa bola, aí a curva plot, que é utilizada para fazer o gráfico registra os valores de tempo e posição horizontal da bola na curva do gráfico. Então o plot é utilizado na curva do gráfico, lembrando que o gráfico está aqui em cima, e o plot a gente usou aqui em baixo dentro do loop, e a legenda a gente tem que atualizar pra ela marcar em tempo real entre as aspas o que está acontecendo com o lançamento oblíquo, então ela atualiza a posição horizontal da bola.

O código da simulação em questão, é uma aplicação do lançamento oblíquo e pode ser observado pelo código abaixo e também no repositório⁹ do GlowScript criados pelos alunos.

```
Web VPython 3.2
```

```
# Importe a biblioteca necessária
```

⁹ Repositório online do código feito pelos grupo de alunos:

<https://www.glowscript.org/#/user/rgaspaparetoquierelli/folder/MyPrograms/program/TrabalhoCurso>

```

from vpython import *

# Crie o gráfico
grafico = graph(title="Posição Horizontal vs Tempo", xtitle="Tempo
(s)", ytitle="Posição Horizontal (m)")
curva = gcurve(color=color.red) # Curva para plotar os valores

# Crie os objetos 3D
bola = sphere(pos=vec(0, 1, 0), radius=1, color=color.blue,
make_trail=True)
solo = box(pos=vec(0, 0, 0), size=vec(100, 0.5, 5),
color=color.purple)

# Condições iniciais
theta = 30 * (pi / 180)
g = vec(0, -9.8, 0)
bola.v = vec(20 * cos(theta), 20 * sin(theta), 0)
t = 0
dt = 0.001

# Legenda
legenda = label(pos=vec(0, 20, 0), text="Legenda")

# Equações e simulação
while bola.pos.y >= 1:
    rate(500)
    bola.v = bola.v + g * dt
    bola.pos = bola.pos + bola.v * dt
    t = t + dt

# Registre os valores de tempo e posição x da bola no gráfico
curva.plot(t, bola.pos.x)

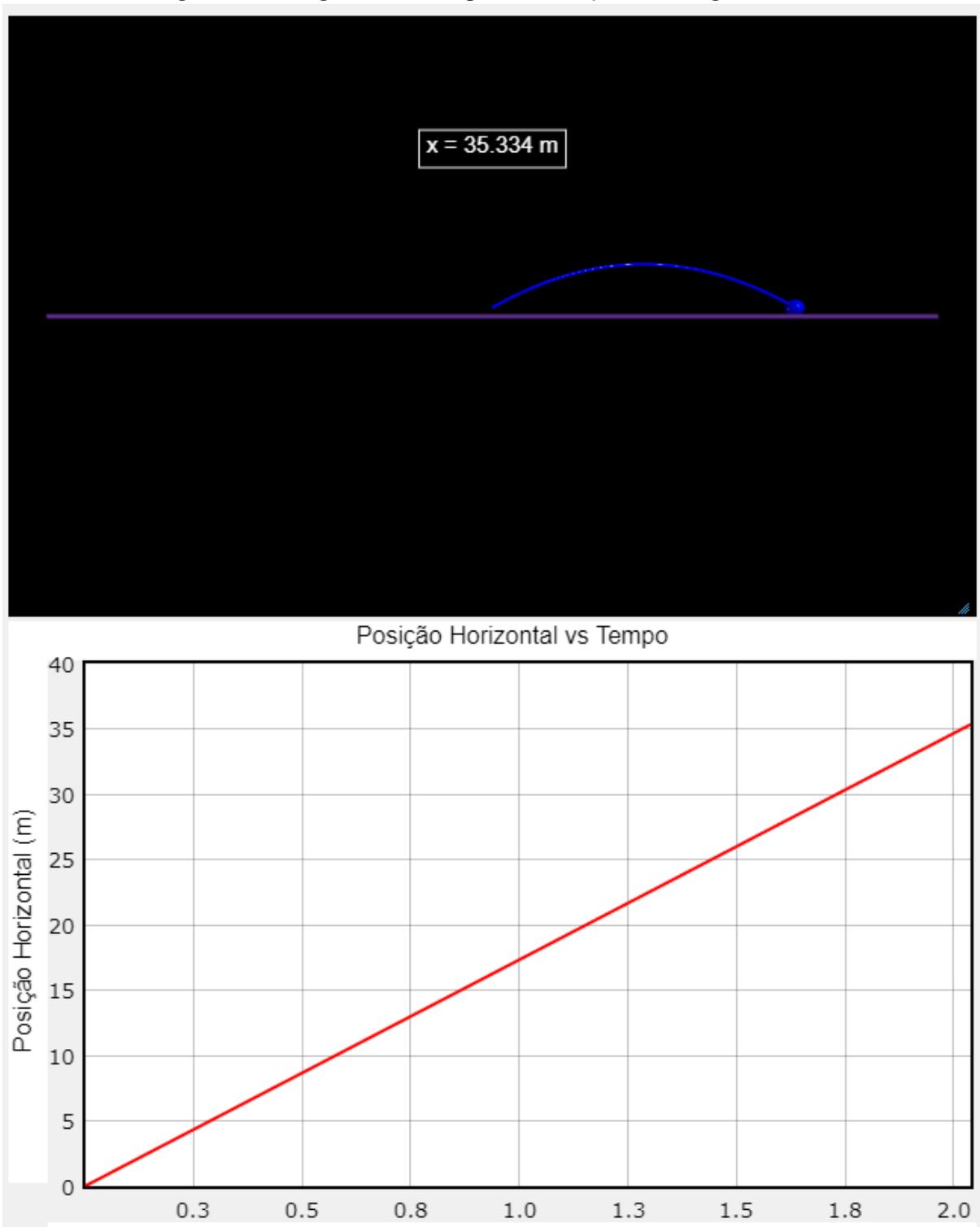
# Atualize a legenda
legenda.text = "x = {:.3f} m".format(bola.pos.x)

```

Além disso, percebe-se nesse grupo um maior cuidado em documentar o código, ou seja, expor comentários no próprio script sobre cada parte do código e sua aplicação na simulação. Além de demonstrar que os alunos entenderam a sequência lógica para a execução da simulação de um lançamento oblíquo baseado em um jogo de golf em que as condições e

variáveis iniciais são representadas de forma coerente no código, e ao executá-lo obtemos o seguinte resultado.

Figura 20 - Imagem retirada após a execução do código Trabalho Curso.



Fonte: Feito pelo autor.

Portanto, a partir dos relatos dos alunos, da análise dos seus códigos e dos resultados de suas simulações, é possível concluir que houve aprendizado tanto dos conhecimentos e conceitos de Física, como os de programação, uma vez que o conteúdo abordado em aula através das contextualizações feitas pelo pesquisador professor e debatida com os alunos, se revelaram posteriormente em experiências aplicáveis em simulações construídas a partir da linguagem de programação Python, o que demonstra uma profunda compreensão dos alunos, uma vez que as simulações demonstram uma representação real dos fenômenos físicos levando em conta os parâmetros foram reduzidos a eventos isolados onde não há perda de energia mecânica.

4.3.2 Problematização e/ou aplicação do assuntos abordados em aula

Segundo Angotti (2015), a terceira etapa dos 3MP é um momento crucial para aplicar os conhecimentos que foram incorporados pelos alunos durante as aulas, esse processo se faz por meio da aplicação e ou problematização desses conhecimentos, para que então possamos compreender se o processo de ensino e aprendizagem foi eficaz. Sendo assim, esse tipo de indicador demonstra indícios de aprendizagem, logo, destacamos alguns trechos que apresentam esses relatos.

Aluna C: E aí, a gente decidiu criar uma historinha, e o Gabriel [figurante da história], em uma de suas viagens, decidiu aprender um novo hobby, que era jogar golfe, e ele é um físico curioso, porque ele fica medindo as coisas, né? Assim, eu acho que é falta do que fazer, mas ele é muito curioso e fica medindo as coisas, tipo a torre de pisa, quem faz isso no meio de uma viagem, né, professor?

Professor: Não sei, nunca fui para a Itália.

Aluna C: Ah, claro. E aí, ele decidiu jogar golfe, e como ele é um curioso nato, ele decidiu calcular a trajetória da bola nos seus primeiros dias jogando. Então, a nossa animação vai ser sobre isso, vai ser sobre lançamento oblíquo, simulando o lançamento oblíquo, que envolve a criação, e quando a bola é lançada em uma velocidade inicial, ela vai para uma certa altura e cai devido à gravidade. E durante a simulação, a posição horizontal da bola é registrada e exibida em um gráfico em tempo real. O código também possui elementos visuais, como a esfera representando a bola

O trecho destacado é um dos exemplos de contextualização apresentados pelos alunos durante o curso, esse trecho é um relato feito pelo Aluna C sobre a aplicação dos conceitos de

física em uma situação prática envolvendo programação feita pelo seu grupo. Nele contém a contextualização do problema e aplicação dos assuntos abordados em aula.

Para contextualizar a situação hipotética, os alunos criam uma narrativa envolvendo um personagem, Gabriel, que decide aprender a jogar golfe e calcular a trajetória da bola nos seus primeiros dias de prática. Essa contextualização torna o problema mais concreto e envolvente para os alunos, facilitando a compreensão e a aplicação dos conceitos de física.

Além disso, a situação proposta envolve o lançamento oblíquo, um conceito de física que descreve o movimento de um objeto lançado com uma velocidade inicial em um ângulo em relação à horizontal. A partir disso os alunos utilizaram esse conceito para simular a trajetória da bola de golfe e entender como ela é afetada pela gravidade ao longo do tempo.

Essa aplicação prática dos conceitos de física foi criada pelo grupo do Aluna C, por meio da programação em Python, onde os alunos criaram um código que registra e exibe a posição horizontal da bola em um gráfico em tempo real. O que demonstra a capacidade dos alunos em utilizar a linguagem de programação para resolver problemas do mundo real relacionados à física.

Desta forma, é possível aferir que a proposta de integrar habilidades de programação com conceitos de física, demonstrou uma abordagem interdisciplinar que promove uma compreensão mais abrangente e aplicada dos assuntos abordados em aula. Os participantes são incentivados a pensar de forma criativa, aplicando seus conhecimentos teóricos em situações práticas e contextualizadas.

Além disso, durante os diálogos das aulas é possível observar diversos momentos que os alunos conseguem entender e interpretar as situações físicas que estão sendo contextualizadas, como exemplificado nestes relatos.

Professor: Na minha cesta, se eu quero atingir meu objetivo. Agora pensa bem, eu falei de algumas grandezas físicas aí, quais são grandezas físicas que estão impregnadas aí na minha fala?

CB - Aluno G: tudo isso tem a ver com grandeza física

CB - Aluno E: força

CB - Aluno H: Atmosfera?

CB - Aluna B: ângulo, velocidade, altura

CB - Aluno G: ângulo

CB - Aluna C: ângulo

CB - Aluna C: velocidade

No primeiro trecho o professor relata sobre o arremesso de basquete e quais as possíveis variáveis que influenciam o arremesso para acertar a cesta. Logo após é possível identificar que os alunos conseguem relacionar esse problemas com as grandezas física que a ele estão impostas

4.3.3 Indicativos de potencialidade ou limitação do ensino de Física com programação

Segundo Wing (2006), o pensamento computacional não é uma habilidade intrínseca das pessoas que trabalham com ciência da computação, mas é algo fundamental para todas as pessoas, pois desenvolve as habilidades de leitura, raciocínio lógico, capacidade analítica entre outras habilidades que permeiam a época em que vivemos.

Essas habilidades são importantes para a grande maioria das disciplinas ou conteúdos escolares, e também estão ligadas ao aprendizado de programação como discutido na fundamentação teórica. Com isso, essa categoria é importante pois visa destacar as potencialidades ou limitações que a linguagem de programação Python pode possuir como recurso de ensino e aprendizado em Física.

Desta forma, como apresentada na fundamentação teórica por Cruz *et al.* (2022), Rodrigues e Machado (2022) e Yamaguti e Carvalho Neto (2023), ainda existe uma lacuna que discuta a integração entre o aprendizado de Física e o uso da linguagem de programação Python como recurso pedagógico, que vise o desenvolvimento do aprendizado em Física por meio da construção de programas ou simuladores em Python e não somente a utilização de um aplicativo, software ou recurso finalizado.

Para tal, busca-se nessa categoria explorar as potencialidades sobre o uso da linguagem de programação Python como recurso pedagógico, e também se sua utilização emprega dificuldades e/ou limitações. Com isso, destacamos a seguir um diálogo entre alunos e pesquisador professor, que ocorreu durante o último encontro, o qual apresenta alguns desses indicativos.

Aluno G: Eu tentei até mexer com o PyCharm, só que no PyCharm eu não consegui baixar o VPython do meu PC. Então, eu voltei pro Glow Script e tive que ver um material de como fazer um gráfico. E daí, eu consegui isso aqui.

Aluno G: Os conceitos da cinemática, para bolar uma explicaçãozinha. Então, deixa eu pegar aqui o grupo no zap. Tinha um grupinho aqui. Os caras que estavam no grupo do zap, pode comprovar o quanto que a gente sofreu para fazer essa simulação aqui, professor. Não foi fácil.

Aluno G: O Lucas assistiu novamente a última ou a penúltima aula, não lembro, e começou a mandar scripts mostrando como você montou os programinhas. E você montando parece tão fácil, professor. Parece tão fácil, cara.

CB - *Aluna B: nossa sim*

Aluna C: Nossa, verdade. Ele fazendo parece que é a coisa mais fácil do mundo. Na hora que você vai fazer isso, você só pode quebrar a cabeça.

Aluno G: Parece que você está escrevendo uma redaçoãozinha. Imagina. É.

Professor: Não se preocupem pessoal, isso vem com a prática galera. Mas diz pra mim, por favor; durante esse processo, como foi o aprendizado de física? Porque eu vi que vocês usaram situações reais, ambos os grupos, né! Utilizaram situações reais, com dois tipos de movimento, uniforme e também uniformemente variado. Como foi a assimilação desse conhecimento com a linguagem de programação? Ela ajudou, ela dificultou? Se ela ajudou, como que ela ajudou? Como que foi a simulação desse conhecimento ou se ela ajudou, como que ela ajudou? Vocês conseguiriam dar esse overview?

Aluno G: Ajudou. A gente teve que usar a fórmula da aceleração, que eu até deixei marcadinho aqui, metros por segundo ao quadrado, aceleração do segundo aluno, para fazer o aluno ganhar velocidade. Então a gente colocou que o segundo aluno, a velocidade dele ia ser baseado numa multiplicação da aceleração, que é 0.3, pelo delta time e que ia vir a posição dele. Então a gente teve que ver as fórmulas de velocidade, de aceleração, de deslocamento, para entender como isso ia funcionar. E a gente pesquisou vários conceitos da cinemática, que a gente poderia usar, e a gente pegou o mais fácil que a gente conseguiu, porque como eu disse, estava bem difícil de fazer alguma coisa sair, mas saiu, deu tudo certo. E justamente da aceleração e da velocidade constante, foi o mais fácil e essa pesquisa foi muito boa, porque a gente conseguiu ver vários conceitos diferentes da cinemática, várias possibilidades do que a gente poderia fazer, mas como não tínhamos habilidade de código

suficiente para fazer algo mega elaborado e chique, a gente pegou um conceito interessante, simples, que dá para traduzir legal de uma forma visual e fez o programinha aqui.

Aluno G: Com essas simulações a gente consegue visualizar algo que normalmente quando você está estudando física e matemática é muito difícil, porque, por exemplo, você vai estudar um pêndulo, daí tem que fazer uma gambiarra com um cadarço, alguma coisa assim para poder visualizar. Isso quando dá para visualizar? Tem muitos conceitos que são basicamente impossíveis. Através dessas simulações em 3D a gente consegue botar de uma forma visual, uma forma que a gente consegue compreender o que está acontecendo. Ao invés de só ficar na teoria ou em fórmulas com números, a gente consegue realmente entender por que é desse jeito, como que é desse jeito, o que levou a ser assim, a gente entende muito mais o processo e o que justifica ele ser desse jeito.

Aluno E: Acho que assim, no começo você começa a ficar meio perdido, porque você não está muito acostumado a fazer, porque é algo que você não é muito comum no dia a dia. Aí com o tempo você vai acostumando, mas eu acho que potencializou, porque às vezes, principalmente em física, às vezes você é muito conceito teórico e às vezes você não consegue ver; pelo menos para mim, é mais fácil a gente ver uma coisa pela imagem do que ver na teoria, que é muito mais fácil. E você vendo por imagem, visualmente, acho que é muito melhor e mais fácil você compreender.

Aluno G: A gente tinha que entender por que esses conceitos eram desse jeito para eles funcionarem. Então, por exemplo, como eu já mencionei, na parte da aceleração, de fazer um aluno ter aceleração e o outro não, a gente teve que pesquisar as fórmulas de como que a aceleração funciona, qual é a diferença da velocidade constante para a velocidade variável. Então, na hora de construir o código, teve essa diferença. A gente teve que entender o que torna um objeto que se move no mesmo ritmo, diferente de um objeto que ganha velocidade. Para realmente botar isso na prática e conseguir construir que um aluno vai ter velocidade fixa de 1.0 e o outro aluno começa com 0 de velocidade, mas vai ganhando 0.3 a cada segundo, o que eventualmente torna ele mais rápido do que o aluno de 1.0.

O trecho destacado é um diálogo aberto no final do quinto encontro em que podemos observar relatos sobre os desafios enfrentados pelos alunos e as potencialidades que a integração da linguagem de programação Python como recurso para ensino e aprendizagem de Física pode trazer.

Inicialmente o aluno G menciona dificuldades técnicas em instalar a biblioteca VPython na IDE PyCharm. Isso demonstra um dos desafios enfrentados pelos alunos ao lidar com ferramentas de programação que não fazem parte de seu cotidiano, além disso, demonstra a importância de recursos variados e acessíveis para que o aluno possa escolher o que for mais adequado para ele, neste caso o GlowScript.

Outro fato importante, foi que os alunos sentiram-se motivados a pesquisar conceitos em programação e em Física, o que demonstra um engajamento no aprendizado ao utilizar a programação para simular conceitos de Física. Como também, é possível perceber no diálogo suas experiências em grupo de pesquisa e aprendizado de conceitos de cinemática para implementar as simulações, e a importância da compreensão desses conceitos abordados em aula para o bom funcionamento das simulações.

Além disso, as simulações feitas em Python também foram vistas como uma ferramenta poderosa para visualizar e compreender conceitos abstratos de Física. Os participantes enfatizam a capacidade das simulações em 3D de fornecer uma representação visual dos fenômenos físicos, facilitando a compreensão dos conceitos teóricos e promovendo uma aprendizagem mais significativa.

Portanto, mesmo que no início os alunos apresentem uma falta de familiaridade com a programação devido à pouca experiência prévia, eles observam que com o decorrer das aulas, eles se acostumam e percebem seus benefícios, especialmente em termos de facilitar a compreensão de conceitos teóricos complexos. Logo, diante desses relatos é possível compreender que o aprendizado da linguagem de programação Python aliada ao ensino de Física é percebida pelos alunos como uma oportunidade para potencializar o aprendizado, tornando os conceitos mais acessíveis, visuais, experimentais e significativos ao permitir uma compreensão mais profunda e intuitiva dos fenômenos estudados.

Além disso, essas mesmas conclusões podem ser observadas nos relatos das entrevistas abertas disponibilizadas aos alunos, os quais destacamos alguns trechos abaixo.

EA - Aluno G: Foi uma ótima experiência, o uso da programação ajudou a visualizar conceitos físicos e a didática do professor foi ótima, as aulas eram muito interativas e interessantes.

Apesar de no começo ter sido bem difícil, com o tempo a lógica ficou mais clara e auxiliou para o entendimento do porque algumas coisas são do jeito que são

Há muitos pontos positivos no ensino do curso, o principal sendo que o uso da programação ajuda a fazer mais sentido do porque os fenômenos físicos acontecem, o único ponto negativo seria a dificuldade de implementação de aulas de programação no ensino tradicional.

Construir códigos em Python mostra desde o começo como se é construída uma fórmula e como um fenômeno físico realmente acontece, por exemplo, como funciona velocidade, aceleração, gravidade, peso e etc.

EA - Aluna B: Tive um pouco de dificuldade no início, pois nunca havia tido contato com programação. No entanto, ao passar das aulas consegui entender e as fórmulas físicas ficaram mais fáceis de serem entendidas com a programação.

Somente positivos, auxiliaram na melhor visualização, sempre tive dificuldade com questões de lançamento oblíquo e com a programação ficou mais fácil de ser compreendido.

Com a linguagem consegui entender quais eram as variáveis utilizadas, conseqüentemente conseguir entender melhor as fórmulas e trazer isso para as resoluções das questões de vestibulares.

EA Aluno F: A minha experiência durante o curso foi muito boa, pois melhorou o meu aprendizado, ajudou no entendimento e no modo de se inserir a programação junto à física.

No começo do curso eu senti um pouco de dificuldade, no entendimento e aprendizado. Mas com isso eu vi que não era tão difícil, e logo peguei o jeito. Percebi que melhorou muito meu modo de pensar durante o curso e ajudou no meu aprendizado!!

Nos três trechos destacados das entrevistas abertas, é possível identificar que os alunos tiveram dificuldade ou até mesmo um espanto inicial ao se depararem com o uso da linguagem de programação Python, o que demonstra um ponto de limitação que precisa ser explorado para minimizar o impacto inicial do aluno e atenuar possíveis frustrações. Mesmo assim, foi possível observar que após o período de adaptação com a linguagem de programação os três alunos reconhecem que seu uso potencializa o aprendizado dos fenômenos físicos. Além disso, um trecho marcante a ser destacado é o relato do Aluna B que diz: “*sempre tive dificuldade com questões de lançamento oblíquo e com a programação ficou mais fácil de ser compreendido*”. Isso demonstra um forte indicativo do uso da linguagem de programação como recurso de intervenção pedagógica para dificuldades internalizadas pelos alunos.

Por fim, um ponto importante destacado pelo Aluna B é a possibilidade de resistência ou dificuldade de implementação do uso da linguagem de programação no ensino tradicional. Essa percepção do aluno pode ser comparada com os desafios destacados por Moreira (2017) em sua crítica ao ensino tradicional brasileiro e a falta de implementação das TDIC no ensino, dentre esses aspectos é relevante destacar que a formação ou capacitação docente é imprescindível para a aplicação de uma intervenção pedagógica que utilize como recurso algum tipo de linguagem de programação.

CONSIDERAÇÕES FINAIS

A presente pesquisa se destinou a explorar as potencialidade e limitações do uso da linguagem de programação Python como recurso pedagógico no ensino de Física, organizados em uma sequência didática baseada nos três momentos pedagógicos propostos por Delizoicov e Angotti (Delizoicov; Angotti, 1990a, Muenchen; Delizoicov, 2014). No decorrer da pesquisa e aplicação da sequência didática, investigamos a relação do uso da linguagem de programação Python nas atividades de ensino de Física, e constituímos dados para serem analisados os possíveis indícios de aprendizagem dos alunos e avaliar seus benefícios e desafios.

A partir dos resultados obtidos nas análises foi possível evidenciar que a linguagem de programação é um recurso que ajuda a promover a compreensão dos conceitos físicos, isso foi identificado tanto nos relatos dos alunos, quanto nos códigos criados por eles. Nos relatos, é apontado desde o entendimento das grandezas físicas que precisam ser instanciadas como variáveis nos códigos, até mesmo a relação dessas grandezas com as equações dos movimentos que precisam reproduzir o fenômeno que está sendo estudado em simulações computacionais. Além disso, foi possível identificar indícios que o uso da programação aliado aos 3MP engajou os alunos a serem mais ativos e explorarem formas diferentes de aplicar os conhecimentos aprendidos no curso.

Ao longo das análises, observamos no início do curso que os alunos enfrentaram alguns desafios para usar a linguagem de programação Python, como a instalação da IDE PyCharm, a instalação da biblioteca VPython e a assimilação de sintaxes próprias da linguagem de programação como o uso de palavras reservadas, dois pontos (“ : ”) ao criar um função ou laço e até mesmo o uso do ponto final (“ . ”) nos números decimais ao invés de vírgula.

No entanto, esses desafios foram amplamente superados por meio de iniciativas autodidatas, colaboração entre os alunos por meio de bate-papos extra curso, apoio do pesquisador professor e também pelo fornecimento de ferramentas auxiliares como o GlowScript para suavizar a carga de conhecimentos técnicos em programação. Acredita-se que a experiência de enfrentar esses desafios não apenas potencializou as habilidades dos participantes, mas também promoveu o desenvolvimento de competências essenciais, como resolução de problemas, pensamento crítico e trabalho em equipe.

Além disso, a integração da linguagem de programação Python nas atividades de ensino de Física proporcionou uma experiência de aprendizagem mais dinâmica e envolvente, que se traduziu em uma melhor compreensão dos conceitos físicos e uma maior motivação para o estudo da disciplina. Os alunos relataram uma maior facilidade em visualizar e aplicar os conceitos aprendidos por meio das simulações em 3D, destacando o potencial dessa abordagem para o ensino e aprendizagem de Física.

No entanto, como destacado pelos próprios alunos, é importante reconhecer que o uso da linguagem da programação Python no ensino de Física também apresenta desafios e limitações a serem considerados. Assim como os desafios apresentado por Moreira (2017) sobre as TDIC, a aplicação da linguagem de programação também enfrenta limitações como a falta de recursos adequados nas escolas, a curva de aprendizado inicial e a necessidade de adaptação curricular são questões que requerem atenção e investimento por parte das instituições educacionais e dos formuladores de políticas públicas.

Além disso, podemos refletir nessa pesquisa também sobre a prática docente que foi realizada pelo pesquisador professor que possui experiência na área de tecnologia da informação, fato que pode ser uma limitação para aplicação de uma intervenção pedagógica semelhantes nas escolas de ensino básico, uma vez que a grade curricular nas faculdades de licenciaturas não exigem uma formação obrigatória em linguagem de programação e/ou pensamento computacional. Por conta disso, mesmo que os resultados sejam satisfatórios, é preciso que o docente que pode vir aplicar a prática pedagógica realizada nesta pesquisa esteja capacitado. Sendo assim, podemos considerar que essa ponderação a respeito do professor pode nos levar a um tema de pesquisa voltado para os cursos de licenciatura e como as tecnologias digitais da informação e comunicação estão inseridas neles.

Outra possibilidade para continuação desta pesquisa é o desenvolvimento de recursos, estratégias, metodologias e práticas pedagógicas que minimizem o impacto e estranheza inicial causada pelos primeiros contatos com as linguagens de programação, a fim de não causar frustrações no processo de ensino e aprendizado.

Em suma, acreditamos que os resultados desta pesquisa mostraram que o uso da linguagem de programação Python no ensino de Física pode potencializar o processo de ensino e aprendizagem em Física, além de representar uma abordagem inovadora e promissora para novas pesquisas. E para maximizar os benefícios dessa abordagem, é necessário um compromisso contínuo com o desenvolvimento profissional dos educadores,

investimento em infraestrutura e recursos educacionais que permitam a aplicação de práticas experimentais, exploratórias tecnológicas e digitais.

REFERÊNCIAS BIBLIOGRÁFICAS

ANGOTTI, J. A. P. **Ensino de Física com TDIC**. 1. ed. rev - Florianópolis: UFSC/EAD/CED, 2015. 125 p.

AULER, D. Alfabetização Científico-Tecnológica: um novo. **Ensaio Pesquisa em Educação em Ciências**, v. 5, n. 1, p. 68-83, jun. 2003. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/1983-21172003050107>.

AULER, D.; DELIZOICOV, D. Alfabetização Científico-Tecnológica para quê? **Ensaio Pesquisa em Educação em Ciências**, Belo Horizonte, [S.L.], v. 3, n. 2, p. 122-134, dez. 2001. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/1983-21172001030203>.

BARDIN, L. **Análise de conteúdo**. São Paulo: Edições 70, 2002.

BECK, K. **TDD desenvolvimento guiado por testes**. 1. ed. Porto Alegre: Bookman, 2010.

BENEVIDES, L. S. **CodeRunner**: proposta de inserção de balanceamento de carga de servidores no contexto da ferramenta educacional. 2019. 71 f. Dissertação (Mestrado em Engenharia da Computação) – Centro de Ciências Computacionais, Universidade Federal do Rio Grande, Rio Grande, 2019.

BRASIL. Ministério da Educação. Secretaria de Educação Básica. **Base Nacional Comum Curricular (BNCC)**. Educação é a Base. Brasília, MEC/CONSED/UNDIME, 2017. p. 213-224.

CARDOSO, S. O. O.; DICKMAN, A. G.. Simulação computacional aliada à teoria da aprendizagem significativa: uma ferramenta para ensino e aprendizagem do efeito fotoelétrico. **Caderno Brasileiro de Ensino de Física**, [S.L.], v. 29, n. 2, p. 891-934, 15 out. 2012. Universidade Federal de Santa Catarina (UFSC). <http://dx.doi.org/10.5007/2175-7941.2012v29nesp2p891>.

CASSAL, M. L. **Programação De Computadores No Ensino De Física Na Educação Básica: Uma Alternativa Inovadora Com O Uso De Recursos Computacionais**. 2020. 341f. Tese(Programa de Pós-Graduação em Ensino de Ciências e Matemática) - Universidade Franciscana, Santa Maria - RS.

CODERUNNER. **Welcome to Moodle CodeRunner**. 2022. Disponível em: <https://coderunner.org.nz/>. Acesso em: 29 jun. 2023.

CRUZ, C. S.; GALVÃO, L. Q.; ROSA, S.; SANTANA, W. S. O uso do python na construção de simuladores computacionais: proposições e potencialidades para o ensino de Física. **Caderno Brasileiro de Ensino de Física**, [S.L.], v. 39, n. 1, p. 204-237, 7 abr. 2022. Universidade Federal de Santa Catarina (UFSC). <http://dx.doi.org/10.5007/2175-7941.2022.e82206>.

DELIZOICOV, D.; ANGOTTI, J. A. **Física**. São Paulo: Cortez, 1990a.

DELIZOICOV, D.; ANGOTTI, J. A. **Metodologia do ensino de ciências**. São Paulo: Cortez, 1990b.

FREIRE, P. **Pedagogia do Oprimido**. 40. ed. Rio de Janeiro: Paz e Terra. 2005.

FOUREZ, G. **Alfabetización Científica y Tecnológica**: acerca de las finalidades de la enseñanza de las ciencias. I. ed. 3. reimp. Buenos Aires: Ediciones Colihue, 2005, 256 p.

GASPAR, A. **Atividades experimentais no ensino de Física: Uma nova visão baseada na teoria de Vigotski**. São Paulo: Editora Livraria da Física. 2014.

- GITHUB. **PYPL PopularitY of Programming Language**. 2023. Disponível em: <https://pypl.github.io/PYPL.html>. Acesso em: 01 jul. 2023.
- GITHUB. **Sobre**. 2023. Disponível em: <https://github.com/about>. Acesso em: 01 jul. 2023.
- GLOWSCRIPT. **Using VPython to create 3D animations**. 2017. Disponível em: <https://www.glowscript.org/docs/VPythonDocs/index.html>. Acesso em: 15 out. 2023.
- GOLDENBERG, M. **A arte de pesquisar: como fazer pesquisa qualitativa em ciências sociais**. 8. ed. Rio de Janeiro: Record, 2004.
- HALLIDAY, D.; RESNICK, R; WALKER, J; **Fundamento de física**. 9.ed. Rio de Janeiro: LTC, v.1, 2012.
- KENSKI, V. M. **Educação e tecnologias: o novo ritmo da informação**. Campinas, SP: Papirus, 2009.
- LEMOS, A. **Cibercultura, tecnologia e vida social na cultura contemporânea**. 6º ed. Porta Alegre, RS: Sulina, 2013. 296 p.
- LÉVY, P. **As tecnologias da inteligência: o futuro do pensamento na era da informática**. Rio de Janeiro: Ed. 34, 1993.
- LOBB, R.; HUNT, T. CodeRunner. **CodeRunner**. 2022. Disponível em: https://github.com/trampgeek/moodle-qtype_coderunne. Acesso em: 14 fev. 2023.
- LUTZ, M. **Aprendendo Python**. 2. ed. Porto Alegre: Bookman, 2007. 568 p.
- MACHADO, G. F.; RODRIGUES, A. Alfabetização científica e tecnologias digitais na educação em ciências: um olhar para a literatura atual. **Brazilian Journal of Education, Technology and Society (BRAJETS)**, v.15, n.1, jan./mar., p.98-112, 2022. Disponível em: <https://brajets.com/v3/index.php/brajets/article/view/841> Acesso em: 30 mar. 2022.
- MALHEIROS, B. T. **Metodologia da Pesquisa em Educação**. Rio de Janeiro: Ltc Editora, 2011. 276 p.
- MANO, W. Moove. **Moodle plugins directory**. Disponível em: https://moodle.org/plugins/theme_moove. Acesso em: 15 out. 2023.
- MANETTA, M. A. **Lançamento oblíquo**. 2013. Disponível em: <http://www.dinamica.com.br/2013/08/lancamento-obliquo.html>. Acesso em: 18 mar. 2024.
- MENEZES, N. N. C. **Introdução à programação com Python: algoritmos e lógica de programação para iniciantes**. São Paulo : Novatec Editora, 2010.
- MILARÉ, T.; RICHETTI, G. P. História e compreensão da alfabetização científica e tecnológica. In: MILARÉ, T. *et al.* (orgs). **Alfabetização científica e tecnológica na Educação em Ciências - Fundamentos e Práticas**. São Paulo: Livraria da Física, 2021, 186 p.
- MILARÉ, T. **Ciências na 8ª Série: Da Química Disciplinar à Química do Cidadão**. 2008. 280 f. Dissertação (Mestrado) - Curso de Curso de Mestrado em Educação Científica e Tecnológica, Universidade Federal de Santa Catarina, Florianópolis, 2008.
- MOODLE. **Guia rápido da instalação**. 2023a. MoodleDocs. Disponível em: https://docs.moodle.org/all/pt_br/Guia_r%C3%A1pido_da_instala%C3%A7%C3%A3o. Acesso em: 15 out. 2023.
- MOODLE. **Nossa Missão**. 2023b. Disponível em: <https://moodle.com/pt-br/sobre/>. Acesso em: 29 jun. 2023.

MORAES, R. Análise de conteúdo. **Revista Educação**, Porto Alegre, v. 22, n. 37, p. 7-32, 1999.

MOREIRA, M. A. Grandes Desafios para o Ensino da Física na Educação Contemporânea. **Revista do Professor de Física**, [S.L.], v. 1, n. 1, p. 1-13, 7 ago. 2017. Biblioteca Central da UNB. <http://dx.doi.org/10.26512/rpf.v1i1.7074>.

MOZZATO, A. R.; GRZYBOVSKI, D. Análise de conteúdo como técnica de análise de dados qualitativos no campo da administração: potencial e desafios. **Revista de Administração Contemporânea**, [S.L.], v. 15, n. 4, p. 731-747, ago. 2011. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s1415-65552011000400010>.

MUENCHEN, C.; DELIZOICOV, D. Os três momentos pedagógicos e o contexto de produção do livro. **Ciência & Educação (Bauru)**, [S.L.], v. 20, n. 3, p. 617-638, set. 2014. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/1516-73132014000300007>.

ORTEGA Y GASSET, J. **Meditação da técnica**. Rio de Janeiro: Livro Ibero-Americano, 1963.

PANOZZO, N. S. P. Pesquisa qualitativa na educação. In: Nilda Stecanela. (Org.). **Diálogos com a educação: a escolha do método e a identidade do pesquisador**. 1ed. Caxias do Sul: Educ, 2013, v. 2, p. 99-111.

PIRES, A. S. T. **Evolução das ideias da Física**. 2. ed. São Paulo: Livraria da Física, 2011. 478 p.

PSF. **Python, a programming language changes the world**. 2023. Disponível em: <https://brochure.getpython.info/media/releases/prerelases/psf-python-brochure-vol-1-final-content-preview>. Acesso em: 19 set. 2023.

PYGAME (comp.). **Pygame Front Page**. 2023. Disponível em: <https://www.pygame.org/docs/>. Acesso em: 15 out. 2023.

PYTHON. **Turtle — Turtle graphics**. 2023. Disponível em: <https://docs.python.org/3/library/turtle.html>. Acesso em: 15 out. 2023.

REYNOL, F. **Guerra fria promoveu a corrida tecnológica**. In: Guerra e ciência. 2004. Disponível em: <https://www.comciencia.br/dossies-1-72/reportagens/guerra/guerra07.htm>. Acesso: 17 ago. 2023.

MACHADO, G.; RODRIGUES, A.. Alfabetização Científica e Tecnologias Digitais na Educação em Ciências: um olhar para a literatura atual. **Cadernos de Educação Tecnologia e Sociedade**, [S.L.], v. 5, n. 1, p. 98-112, 21 mar. 2022. Brazilian Journal of Education, Technology and Society (BRAJETS). <http://dx.doi.org/10.14571/brajets.v15.n1.98-112>.

RODRIGUES, A.; MACHADO, G. F. Alfabetização Científica e Tecnologias Digitais de Informação e Comunicação: reflexões teóricas para a educação em ciências. **Atos de Pesquisa em Educação**, [S.L.], v. 18, p. 9844, 6 jul. 2023. Fundação Universidade Regional de Blumenau. <http://dx.doi.org/10.7867/1809-03542022e9844>.

SAHB, W. F.; ALMEIDA, F. J. Tecnologia como direito: as TDIC e o desafio da educação escolar. **Comunicações**, v. 23, n. 2, p. 69-91, 2016. Disponível em: <https://www.metodista.br/revistas/revistas-unimep/index.php/comunicacoes/article/download/2841/1718>. Acesso em: 22 jan. 2022.

SASSERON, L. H.; CARVALHO, A. M. P. Alfabetização Científica: Uma Revisão Bibliográfica. **Investigações Em Ensino De Ciências**, v.16, n.1, p. 59–77, 2016. Recuperado de <https://ienci.if.ufrgs.br/index.php/ienci/article/view/246>

SIEMSEN, G. H. **O ensino de astronomia em uma abordagem interdisciplinar no ensino médio**: potencialidades para a promoção da alfabetização científica e tecnológica. 2019. 248 f. Dissertação (Mestrado) - Curso de Educação em Ciências e em Matemática, Universidade Federal do Paraná, Curitiba, 2019.

SILVA, V. R.; LORENZETTI, L. A alfabetização científica nos anos iniciais: os indicadores evidenciados por meio de uma sequência didática. **Educação e Pesquisa**, v. 46, p. e222995, 2020.

SOUZA, S. C.; DOURADO, L. Aprendizagem Baseada em Problemas (ABP): um método de aprendizagem inovador para o ensino educativo. **Holos**, [S.L.], v. 5, p. 182-200, 1 out. 2015. Instituto Federal de Educacao, Ciencia e Tecnologia do Rio Grande do Norte (IFRN). <http://dx.doi.org/10.15628/holos.2015.2880>.

VALENTE, J. A. A Comunicação e a Educação baseada no uso das Tecnologias Digitais de Informação e Comunicação. **Revista UNIFESO**, v. 1, n. 1, p. 141-166, 2014.

VALENTE, J. A. O uso inteligente do computador na educação. **Revista Pátio**, v. 1, n. 1, p. 19-21, 1997.

VALENTE, J.A. Análise dos diferentes tipos de softwares usados na educação. *In*: VALENTE, J. A. (Org.). **O computador na sociedade do conhecimento**. Campinas, SP: Gráfica da UNICAMP, 1999.

VALENZUELA-ZAMBRANO, B.; PEREZ-VILLALOBOS, M. V. Aprendizaje autorregulado a través de la plataforma virtual Moodle. **educ.educ.**, Chia , v. 16, n. 1, p. 66-79, Apr. 2013 . Disponível em: <http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0123-12942013000100004&lng=en&nrm=iso>. Acesso em: 29 jun. 2023.

VAN ROSSUM, G.; DRAKE JR, F. L. **Python tutorial**. Amsterdam, The Netherlands: Centrum voor Wiskunde en Informatica, 1995.

WING, J. M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33–35, 2006.

WORLD ECONOMIC FORUM (Suíça). World Economic Forum. **The Future of Jobs Report 2023**. Genebra: Wef, 2023. 296 p. Disponível em: <https://www.weforum.org/reports/the-future-of-jobs-report-2023/>. Acesso em: 25 maio 2023.

YAMAGUTI, M. X. **Testes-de-Bibliotecas-Para-Animacao-Físicas**. 2023. Disponível em: <https://github.com/mateusyamaguti/Testes-de-Bibliotecas-Para-Animacao-Físicas>. Acesso em: 15 out. 2023.

YAMAGUTI, M. X.; CARVALHO NETO, J. T. O Uso de Linguagem de Programação no Ensino de Física: Uma Revisão da Literatura. *In*: **Anais do Encontro Nacional de Pesquisa em Educação em Ciências**. Campina Grande: Realize Editora, 2023. Disponível em: <<https://editorarealize.com.br/artigo/visualizar/92997>>. Acesso em: 02 jan. 2023.

YAMAGUTI, M. X.; ARLE, R. J. . Napp Academy: Programa De Incentivo à Educação Tecnológica ? Um Estudo de Caso. *In*: **Congresso Internacional de Educação e Tecnologias**, 2022, São Carlos. Anais do CIET:CIESUD:2022, 2022. p. 1-5.

YAMAGUTI, M. X.; GOMES, A. E. R.; ARLE, R. J. Resultados obtidos pelo Estudo de Caso do Programa de Incentivo a Educação em Tecnologia - Napp Academy. *In*: **Anais do Congresso Brasileiro Interdisciplinar em Ciência e Tecnologia**, 2022, Online. Anais do(a) Anais do Congresso Brasileiro Interdisciplinar em Ciência e Tecnologia. Recife: Even3, 2022.

YAMAMOTO, I.; BARBETA, V. B. Simulações de experiências como ferramenta de demonstração virtual em aulas de teoria de Física. **Revista Brasileira de Ensino de Física**, v. 23, n. 2, p. 215–25, 2001. DOI.org (Crossref), <https://doi.org/10.1590/S1806-11172001000200013>.

YIN, R. K. **Estudo de caso: planejamento e métodos**. 5. ed. Porto Alegre: Bookman, 2015.

APÊNDICES

APÊNDICE A - DOCUMENTAÇÃO DA INSTALAÇÃO DA APLICAÇÃO MOODLE

Configurações iniciais

Acesse o terminal do servidor e execute os seguintes passos:

Passo 01

```
sudo apt update && sudo apt upgrade -y
```

Passo 02

```
sudo timedatectl set-timezone America/Sao_Paulo
```

Passo 03

```
timedatectl status
```

Passo 04

```
sudo reboot
```

Passo 05

```
sudo mkdir -p /opt/moodldata
```

Passo 06

```
sudo chown www-data /opt/moodldata -R
```

Instalação do PHP 8 e configuração de parâmetros

Acesse o terminal do servidor e execute os seguintes passos:

Passo 01

```
sudo apt install lsb-release ca-certificates apt-transport-https  
software-properties-common -y
```

Passo 02

```
sudo add-apt-repository ppa:ondrej/php
```

Passo 03

```
sudo apt-get install php8.0-fpm php8.0-cli php8.0-pgsql php8.0-mbstring  
php8.0-xmlrpc php8.0-zip php8.0-gd php8.0-xml php8.0-bcmath php8.0-ldap  
php8.0-pspell php8.0-curl php8.0-intl php8.0-soap
```

Passo 04

```
sudo systemctl status php*-fpm.service
```

Passo 05

Edite o arquivo: [/etc/php/8.0/fpm/pool.d/www.conf](#) e modifique conforme abaixo.

```
sudo vim /etc/php/8.0/fpm/pool.d/www.conf  
listen = 127.0.0.1:9000  
security.limit_extensions = .php
```

Passo 06

Editar o arquivo: [/etc/php/8.0/fpm/php.ini](#) e modifique conforme abaixo:

```
file_uploads = On  
upload_max_filesize = 64M  
max_execution_time = 300  
memory_limit = 512M  
post_max_size = 64M  
max_input_time = 300  
max_input_vars = 5000  
date.timezone = America/Sao_Paulo
```

Passo 07

```
sudo systemctl restart php*-fpm.service
```

Instalação do NGINX

Acesse o terminal do servidor e execute os seguintes passos:

Passo 01

```
sudo apt install nginx -y
```

Observação: Abaixo estão dois arquivos de configurações sem certificado SSL e com. Recomenda-se se for instalação inicial a começar sem o SSL. Para converter o moodle em SSL existe uma opção dentro dele em configurações para tal, feito isso, utilize o config com SSL. Este passo está citado no final do documento.

```
server {
    listen 80;
    listen [::]:80;
    root /var/www/html/moodle/;
    index index.php index.html index.htm;
    server_name codegeniuskids.com.br;

    client_max_body_size 100M;
    autoindex off;
    location / {
        try_files $uri $uri/ =404;

    }

    location ~ ^(\.+\.php)(.*)$ {
        root /var/www/html/moodle;
        fastcgi_split_path_info ^(\.+\.php)(.*)$;
        fastcgi_index index.php;
        fastcgi_pass 127.0.0.1:9000;
        include /etc/nginx/mime.types;
        include fastcgi_params;
        fastcgi_param PATH_INFO $fastcgi_path_info;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }
}
```

```
server {
    root /var/www/html/moodle/;
    index index.php index.html index.htm;
    server_name codegeniuskids.com.br;

    client_max_body_size 100M;
    autoindex off;
    location / {
        try_files $uri $uri/ =404;

    }
}
```

```

location ~ ^(\.+\.php)(.*)$ {
    root /var/www/html/moodle;
    fastcgi_split_path_info ^(\.+\.php)(.*)$;
    fastcgi_index          index.php;
    fastcgi_pass           127.0.0.1:9000;
    include /etc/nginx/mime.types;
    include                fastcgi_params;
    fastcgi_param          PATH_INFO      $fastcgi_path_info;
    fastcgi_param          SCRIPT_FILENAME $document_root$fastcgi_script_name;
}

location ~
(/vendor/|/node_modules/|composer\.json|/readme|/README|readme\.txt|/upgr
ade\.txt|db/install\.xml|/fixtures/|/behat/|phpunit\.xml|\.lock|environme
nt\.xml) {
    deny all;
    return 404;
}

listen [::]:443 ssl ipv6only=on;
listen 443 ssl; # certificado cloudflare
ssl_certificate /etc/ssl/certs/cert.pem;
ssl_certificate_key /etc/ssl/private/key.pem;
}
server {
    if ($host = codegeniuskids.com.br) {
        return 301 https://$host$request_uri;
    }

    listen 80;
    listen [::]:80;
    server_name codegeniuskids.com.br;
    return 404;
}

```

Passo 02

Crie o arquivo em: </etc/nginx/sites-available/moodle.conf> e faça o link simbólico

```

sudo ln -s /etc/nginx/sites-available/moodle.conf
/etc/nginx/sites-enabled/

```

Passo 04

```
nginx -t
```

Passo 05

```
rm /etc/nginx/sites-enabled/default
```

Passo 06

```
sudo systemctl restart nginx
```

Passo 07

```
sudo systemctl status nginx
```

Instalação PostgreSQL

Acesse o terminal do servidor e execute os seguintes passos:

Passo 01

```
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt  
$(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'
```

Passo 02

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc |  
sudo apt-key add -  
sudo apt-get update
```

Passo 03

```
sudo apt-get -y install postgresql
```

Passo 04

```
sudo vim /etc/postgresql/15/main/postgresql.conf
```

Passo 05

```
listen_addresses = 'localhost'  
port = 5432  
max_connections = 2000
```

Passo 06

```
sudo su postgres
psql

CREATE USER moodleuser WITH PASSWORD 'Senha ocultada para segurança dos
participantes';
CREATE DATABASE moodle WITH OWNER moodleuser;
```

Passo 07

```
sudo systemctl restart postgresql
```

Passo 08

```
sudo systemctl status postgresql
```

Instalação Moodle

Passo 01

Baixe a última release do site oficial <https://download.moodle.org/releases/latest/>

Passo 02

Descompacte em /var/www/html

Passo 03

Permissão as pastas abaixo:

```
sudo chown www-data:www-data /var/www/html/moodle/blocks -R
sudo chown www-data:www-data /var/www/html/moodle/config.php -R
sudo chown www-data:www-data /var/www/html/moodle/theme -R
```

Agendar no CRON¹⁰

Acesse o terminal do servidor e execute os seguintes passos:

Passo 01:

Agendada CRON

```
crontab -e
```

```
* * * * * /usr/bin/php /var/www/html/moodle/admin/cli/cron.php >/dev/null
```

Feito isso, basta apontar o DNS ou acessar via IP a página do Moodle (site ou url que foi criado). E para abrir a tela inicial de instalação, é somente seguir com o botão de próxima.

Conversão para HTTPS

Passo 01

Acesse a administração dos site:

Administração do Site > Segurança HTTP > Ferramenta de conversão HTTPS

Passo 02

Altere o arquivo /var/www/html/moodle/config.php para https

```
$CFG->wwwroot = 'https://codegeniuskids.com.br';
```

¹⁰ Maiores informações disponíveis em:

https://docs.moodle.org/all/pt_br/Cron#:~:text=%2D%20V%C3%A1%20em%20Iniciar%20%3E%3E%20Painel,Agendadas%20%3E%3E%20Adicionar%20Tarefa%20Agendada.&text=%2D%20Type%20%22Moodle%20Cron%22%20as,%22Daily%22%20as%20the%20schedule.

APÊNDICE B - PROGRAMAS MRU FEITO PARA TESTE DE BIBLIOTECAS

Programa feito para testar biblioteca Pygame

```
import pygame
from pygame import *
from sys import exit
'''Iniciar pygame'''
pygame.init()

'''Criar tela do jogo'''
largura = 640
altura = 480
tela = pygame.display.set_mode((largura, altura))

'''Criar variaveis de posição'''
x_rect = 0
y_rect = 240

'''Variável de velocidade'''
velocidade = 1

'''Colocar nome no display da janela'''
pygame.display.set_caption('Teste')

'''Criar objeto para controle de frames'''
relogio = pygame.time.Clock()

'''Laço de execução infinito para rodar o jogo'''
while True:
    '''Determinar os frames/segundo'''
    relogio.tick(100)
    '''limpar rastro do loop do retangulo para manter a tela preta'''
    tela.fill((0, 0, 0))
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
            exit()
    pygame.draw.rect(tela, (255, 0, 0), (x_rect, y_rect, 40, 50))

    x_rect += velocidade # Equação MRU
    pygame.display.update()
```

Programa feito para testar biblioteca Turtle

```
import turtle
from turtle import Turtle
from time import sleep

'''Criação de janela'''
Screen = turtle.screensize(640, 480)

'''Criação de objeto'''
Turtle = Turtle()
Turtle.shape("circle")
Turtle.penup()
Turtle.shapesize(5, 5, 5)

'''Criação de variável de velocidade'''
velocidade = 1

'''Laço de animação'''
while True:
    sleep(0.001)
    Turtle.forward(velocidade) # Equação MRU
    posicao = Turtle.position()
    if (str(posicao).split(",")[0]=="(480.00)":
        break
```

Programa feito para testar biblioteca VPython

```
from vpython import *

# Criação de objetos 3D
objeto = sphere(pos=vector(-80, 1, 0), radius=1, color=color.red)
solo = box(pos=vector(0, 0, 0), size=vector(200, 0.5, 2),
color=color.blue)

# Criação de gráficos
graph = gcurve(color = color.cyan)

# Variáveis iniciais
```

```
t = 0
dt = 0.01 # Incremento de variação temporal de 1 centésimo de
segundo
objeto.vel = vector(2, 0, 0)

# Animação do movimento e equação
while True:
    rate(300)
    t = t + dt
    objeto.pos = objeto.pos + objeto.vel*dt # equação do MRU com
incremento de variação temporal
    graph.plot(objeto.pos.x, t)
```

APÊNDICE C - SCRIPT PARA TRANSCRIÇÃO DE ÁUDIOS

Exemplo de script para transcrição de áudio da aula 02.

```
import whisper
from time import time

'''Tempo inicial'''
time1 = time()

'''Escolha de modelo para transcrição'''
model = whisper.load_model("medium")
result = model.transcribe("Aula02_parte_03.m4a")

'''Conversão da saída em texto'''
texto = str(result["text"])

'''Tempo final da execução da transcrição'''
time2 = time()

'''Tempo total pra transcrição do áudio'''
print(time2 - time1)

'''Salvar arquivo texto em arquivo TXT'''
arquivo = open("transcricao.txt", "a")
arquivo.write(texto)
```

APÊNDICE D - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

CENTRO DE CIÊNCIAS AGRÁRIAS (CCA) - CAMPUS ARARAS

DEPARTAMENTO DE CIÊNCIAS DA NATUREZA, MATEMÁTICA E EDUCAÇÃO (DCNME)

PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO EM CIÊNCIAS E MATEMÁTICA

(PPGEdCM)

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO**(Resolução CNS 510/2016)****O USO DA LINGUAGEM DE PROGRAMAÇÃO PYTHON NO ENSINO DE Física ALIADA AOS TRÊS MOMENTOS PEDAGÓGICOS COMO FERRAMENTA POTENCIALIZADORA DE ENSINO E APRENDIZAGEM**

Eu, Mateus Xavier Yamaguti, estudante do Programa de Pós-Graduação em Educação em Ciências e Matemática da Universidade Federal de São Carlos – UFSCar convido seu(sua) filho(a) a participar da pesquisa “O Uso da Linguagem de Programação Python no Ensino de Física aliada aos Três Momentos Pedagógicos como Ferramenta Potencializadora de Ensino e Aprendizagem”, orientado pelo Prof. João Teles de Carvalho Neto do Programa de Pós-Graduação em Educação em Ciências e Matemática, da Universidade Federal de São Carlos (PPGEdCM/UFSCar).

A pesquisa tem o objetivo de entender como a linguagem de programação Python pode auxiliar no ensino de Física clássica mediante uma sequência didática voltada para os alunos do ensino médio, baseada na metodologia dos Três Momentos Pedagógicos (3MP). Uma vez que houve uma crescente demanda pelo uso de Tecnologias Digitais da Informação e Comunicação (TDIC) no ensino, além da grande expansão do mercado de trabalho na área de tecnologia, entende-se que atualmente é importante pesquisar a respeito da linguagem de programação como ferramenta potencializadora de ensino e aprendizagem. Logo, a proposta desta pesquisa busca entender como a linguagem de programação Python pode ajudar os alunos a aprender os conhecimentos de Física clássica.

Nesse sentido, seu(sua) filho(a) foi convidado(a) a participar desta pesquisa por ser aluno(a) do ensino médio da Escola Técnica Estadual Dep. Salim Sedeh, local onde será divulgado o convite para os alunos. Na realização da pesquisa ele(a) fará parte de um estudo em que o professor pesquisador irá buscar entender as potencialidades da utilização da linguagem de programação Python com ferramenta potencializadora do ensino de Física, mais precisamente sobre os conhecimentos relativos ao conteúdo de cinemática.

A pesquisa será realizada em formato virtual pelo ambiente de aprendizagem Moodle da Universidade Federal de São Carlos onde serão disponibilizados todos os recursos, materiais e atividades. Serão realizadas 5 aulas remotas com duração de uma hora e meia (1h e 30min) cada, as aulas serão realizadas por meio do uso do aplicativo Google Meet, as quais serão gravadas e disponibilizadas para consulta posteriormente no ambiente de aprendizagem virtual, assim como para levantamento de dados que serão tratados posteriormente na análise de dados desta pesquisa.

A participação de seu(sua) filho(a) nessa pesquisa auxiliará na obtenção de dados que poderão ser utilizados para fins científicos, proporcionando maiores informações e discussões que poderão trazer benefícios para a área da Educação no ensino Ciência e Matemática, seja para a construção de novos conhecimentos, seja para a identificação de novas alternativas e possibilidades para o trabalho de professores que atuam no ensino médio.

A participação de seu(sua) filho(a) é voluntária e não haverá compensação em dinheiro pela participação. A qualquer momento seu(sua) filho(a) pode desistir de participar e retirar seu consentimento, até mesmo após o término da pesquisa. A recusa ou desistência de seu(sua) filho(a) não lhe trará nenhum prejuízo, seja em sua relação com o professor pesquisador ou na escola. Todas as informações obtidas através da pesquisa serão confidenciais, sendo assegurado o sigilo sobre a participação de seu(sua) filho(a) em todas as etapas do estudo. Caso haja menção a nomes, a eles serão atribuídas letras aleatórias, com garantia de anonimato nos resultados e publicações, impossibilitando sua identificação.

Entende-se que alguns desconfortos poderão surgir durante a pesquisa. São eles os seguintes: (I) Constrangimento ao participar de uma aula online com captação audiovisual, lembrando que não é obrigatório o estudante habilitar sua câmera ou microfone; (II) Constrangimento ou desconforto ao responder perguntas realizadas pelo professor pesquisador, a fim de problematizar o conhecimento a ser estudado; (III) Possível desgaste mental durante a elaboração de códigos e programas em linguagem de programação Python na aplicação de problemas de cinemática.

Os procedimentos para minimização dos riscos serão: a possibilidade de desistir da participação da pesquisa a qualquer momento. As perguntas jamais serão inquisitivas, mas terão o caráter de convite à resposta, sempre salientando que não há a obrigatoriedade de resposta.. Além disso, também há garantia de esclarecimento sobre os objetivos e benefícios da pesquisa a quaisquer interessados, como forma de evitar possíveis desconfianças sobre a finalidade e seriedade da pesquisa. Será garantido, também, total sigilo sobre as respostas individuais dadas pelo participante, como forma de garantir a segurança e confidencialidade dos dados.

Solicito, também, sua autorização para gravação audiovisual das aulas. As gravações realizadas serão gravadas pelo aplicativo Google Meet e anexadas a uma pasta privada no Google Drive em formato de acesso restrito para garantir que apenas os participantes da pesquisa possam assistir pelo ambiente de aprendizagem virtual onde as aulas serão incorporadas. Posteriormente, as gravações serão transcritas pelo pesquisador, podendo ser auxiliadas por softwares especializados em transcrição de áudio. Essas transcrições serão comparadas com os áudio originais para verificar a concordância, garantindo a fidelidade dos relatos coletados.

Caso seja do seu consentimento que seu(sua) filho(a) participe desta pesquisa nos termos deste documento, você receberá uma via deste termo que deverá ser rubricada em todas as páginas por você e pelo pesquisador, onde consta o telefone e o endereço do pesquisador principal. Você poderá tirar suas dúvidas sobre o projeto e sua participação agora ou a qualquer momento.

Este projeto de pesquisa foi aprovado por um Comitê de Ética em Pesquisa (CEP) que é um órgão que protege o bem-estar dos participantes de pesquisas. O CEP é responsável pela avaliação e acompanhamento dos aspectos éticos de todas as pesquisas envolvendo seres humanos, visando garantir a dignidade, os direitos, a segurança e o bem-estar dos participantes de pesquisas. Caso você tenha dúvidas e/ou perguntas sobre seus direitos como participante deste estudo, entre em contato com o **Comitê de Ética em Pesquisa em Seres Humanos (CEP)** da UFSCar que está

vinculado à Pró-Reitoria de Pesquisa da universidade, localizado no prédio da reitoria (área sul do campus São Carlos). Endereço: Rodovia Washington Luís km 235 - CEP: 13.565-905 - São Carlos-SP. Telefone: (16) 3351-9685. E-mail: cephumanos@ufscar.br. Horário de atendimento: das 08:30 às 11:30.

O CEP está vinculado à **Comissão Nacional de Ética em Pesquisa (CONEP)** do Conselho Nacional de Saúde (CNS), e o seu funcionamento e atuação são regidos pelas normativas do CNS/Conep. A CONEP tem a função de implementar as normas e diretrizes regulamentadoras de pesquisas envolvendo seres humanos, aprovadas pelo CNS, também atuando conjuntamente com uma rede de Comitês de Ética em Pesquisa (CEP) organizados nas instituições onde as pesquisas se realizam. Endereço: SRTV 701, Via W 5 Norte, lote D - Edifício PO 700, 3º andar - Asa Norte - CEP: 70719-040 - Brasília-DF. Telefone: (61) 3315-5877 E-mail: conep@saude.gov.br.

Mateus Xavier Yamaguti

Pesquisador principal

Universidade Federal de São Carlos

Departamento de Ciências da Natureza, Matemática e Educação (DCNME).

Rodovia Anhanguera, km 174 - SP-330

CEP 13600-970 - Araras - São Paulo – Brasil.

Telefone: (19) 3543-2600 ou 3543-2601

mateus.yamaguti@estudante.ufscar.br

Eu, _____, declaro que entendi os objetivos, riscos e benefícios desta pesquisa. Deste modo, concordo e autorizo meu(minha) filho(a) _____ a participar dela.

Ao assinar esse termo, entende-se que você leu e concordou com a participação na pesquisa.

Araras, ____ de _____ de 2023.

Nome do Pesquisador

Representante Legal do Participante

APÊNDICE E - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)**Termo de Assentimento do menor**

Você está sendo convidado para participar da pesquisa: “O uso da linguagem de programação Python no ensino de Física aliada aos três momentos pedagógicos como ferramenta potencializadora de ensino e aprendizagem”, que seus pais permitiram que você participe.

Essa pesquisa tem como objetivo entender como o uso da linguagem de programação Python pode ajudar no ensino de cinemática a partir de alguns encontros que buscarão desenvolver o pensamento computacional, o raciocínio lógico e o aprendizado dos conhecimentos em cinemática.

Os participantes serão alunos do Ensino Médio da Escola Técnica Estadual Dep. Salim Sedeh e sua participação não é obrigatória, além de poder deixar a pesquisa a qualquer momento durante sua realização, ou até mesmo após finalizada a pesquisa. Serão realizadas 5 aulas remotas com duração de uma hora e meia (1h e 30min) cada, as aulas serão realizadas por meio do uso do aplicativo Google Meet, as quais serão gravadas e disponibilizadas para consulta posteriormente no ambiente de aprendizagem virtual, assim como para levantamento de dados que serão tratados posteriormente na análise de dados desta pesquisa. Em alguns encontros iremos abordar os conhecimentos de cinemática, em outros os conhecimentos da linguagem de programação Python e por fim, realizar a aplicação dos conhecimentos da linguagem Python para resolver problemas de cinemática e poder debater sobre o que foi realizado e aprendido.

Caso você queira desistir desta pesquisa, não terá nenhum problema. Mas quem aprender a linguagem de programação Python aplicada aos conhecimentos de cinemática, poderá desenvolver várias habilidades como: pensamento computacional, raciocínio lógico e matemática, lógica de programação, a própria linguagem de programação, resolução de problemas, determinação, resiliência para superar desafios, entre outras habilidades.

Você pode ter alguns desconfortos durante o estudo: (I) Constrangimento ao participar de uma aula online com captação audiovisual, lembrando que não é obrigatório o estudante habilitar sua câmera ou microfone; (II) Constrangimento ao responder às perguntas feitas pelo pesquisador a fim de problematizar o conhecimento a ser estudado, implicando no risco de que alguns estudantes se sintam desconfortáveis em participar dos diálogos suscitados pelas perguntas; (III) Possível desgaste e cansaço mental durante a elaboração de códigos e programas em linguagem de programação python na aplicação de problemas de cinemática.

As conversas e nossas experiências sobre as atividades que nós vamos compartilhar durante o estudo serão utilizadas sempre sem citar os nomes. Isto quer dizer que ninguém ficará sabendo quem fez as contribuições, quando publicarmos os resultados. Ou seja, essas informações são sigilosas e não falaremos a outras pessoas, nem daremos a estranhos os seus dados.

Os resultados da pesquisa vão ser publicados, mas sem identificar os participantes da pesquisa. Quando terminarmos a pesquisa haverá a explicação dos benefícios que o estudo trouxe para você e para a escola.

Se você tiver alguma dúvida, você pode perguntar ao pesquisador professor Mateus Yamaguti que participa com você deste estudo.

Eu _____ aceito participar da pesquisa: “O uso da linguagem de programação Python no ensino de Física aliada aos três momentos pedagógicos como ferramenta potencializadora de ensino e aprendizagem”. Que tem o objetivo de utilizar a

linguagem de programação Python como ferramenta potencializadora do ensino e aprendizagem dos conhecimentos de cinemática. Observando se a sua utilização ajudará a melhorar a minha aprendizagem de conteúdos cinemáticos. Entendi as situações de desconforto e as situações boas que podem me acontecer. Entendi que posso dizer “sim” e participar, mas que, a qualquer momento, posso dizer não, desistir e ninguém ficará bravo comigo ou me tratará de forma diferente por causa disso. O pesquisador professor Mateus tirará minhas dúvidas e já conversou com os meus responsáveis. Recebi uma cópia deste termo de assentimento e li e concordo em participar da pesquisa.

Araras, ____ de _____ de 2023.

Assinatura do menor

Assinatura do pesquisador

APÊNDICE F - CARTA DE AUTORIZAÇÃO ESCOLAR

CARTA DE AUTORIZAÇÃO DA ESCOLA

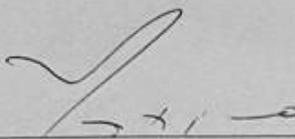
Ao Comitê de Ética em Pesquisa em Seres Humanos da Universidade Federal de São Carlos (UFSCar),

Prezado Comitê de Ética em Pesquisa da UFSCar, na função de representante legal da Escola Técnica Deputado Salim Sedeh, informo que o projeto de pesquisa intitulado “O Uso da Linguagem de Programação Python no Ensino de Física Aliada aos Três Momentos Pedagógicos como Ferramenta Potencializadora de Ensino e Aprendizagem”, apresentado pelo pesquisador, Prof. Mateus Xavier Yamaguti, que tem como objetivo principal entender as potencialidades da utilização da linguagem de programação Python como ferramenta didática para o ensino de cinemática a partir de uma sequência didática baseada nos Três Momentos Pedagógicos (3MP), poderá ser realizado nesta instituição após a apresentação do parecer favorável emitido pelo Comitê de Ética em Pesquisa em Seres Humanos da UFSCar. Solicito a apresentação do Parecer de Aprovação do CEP-UFSCar antes de iniciar a coleta de dados nesta Instituição.

“Declaro conhecer a Resolução CNS 466/12. Esta instituição está ciente de suas corresponsabilidades como instituição coparticipante do presente projeto de pesquisa e de seu compromisso no resguardo da segurança e bem-estar dos sujeitos de pesquisa nela recrutados, dispondo de infraestrutura necessária para a garantia de tal segurança e bem-estar.

Data: Leme 02 de maio de 2023.

Assinatura: _____



Paulo Henrique Maximo

(Nome completo, legível e carimbo institucional do representante legal)



Paulo Henrique Máximo
Diretor de Escola Técnica – ETEC
RG 18.618.019

ETEC Dep. Salim Sedeh. Rua Neida Zencker Leme, 500 - Cidade Jardim, Leme - SP, CEP: 13614-240. Telefone: (19) 3571-3705.
E-mail: e110dir@cps.sp.gov.br

APÊNDICE G - TERMO DE CONFIDENCIALIDADE**TERMO DE CONFIDENCIALIDADE**

Título do projeto: O uso da linguagem de programação Python no Ensino de Física aliada aos Três Momentos Pedagógicos como ferramenta potencializadora de ensino e aprendizagem.

Pesquisador responsável: Mateus Xavier Yamaguti.

Instituição/Departamento: Universidade Federal de São Carlos (UFSCar), Departamento de Ciências da Natureza, Matemática e Educação, Araras-SP.

Telefone para contato: (19) 99816-4736

Local da coleta de dados: Ambiente virtual de videochamada Google Meet

O pesquisador do presente projeto se compromete a preservar a privacidade dos participantes cujos dados serão coletados através da gravação dos encontros da sequência didática. Concorde, igualmente, que estas informações serão utilizadas única e exclusivamente para execução do presente projeto. As informações somente poderão ser divulgadas de forma anônima e serão mantidas em posse do pesquisador responsável por este projeto de pesquisa, por um período de 5 anos sob a sua responsabilidade. Após este período, os dados serão destruídos.

Araras, 06 de março de 2023.

.....
Mateus Xavier Yamaguti

APÊNDICE H - TERMO DE AUTORIZAÇÃO PARA USO DE IMAGEM E SOM**COMITÊ DE ÉTICA EM PESQUISA****Termo de cessão de uso de imagem e voz
Para fins de pesquisa**

Eu, _____ (nome do participante), portador do cpf _____, participante da pesquisa “O uso da linguagem de programação Python no Ensino de Física aliada aos Três Momentos Pedagógicos como ferramenta potencializadora de ensino e aprendizagem” de forma livre e esclarecida, cedo o direito de uso das fotografias, vídeos e voz adquiridos durante a participação desta pesquisa, e autorizo o pesquisador mestrando Mateus Xavier Yamaguti e seu orientador, Prof. Dr João Teles de Carvalho Neto da Universidade Federal de São Carlos, Campus Araras, responsáveis pelo trabalho a:

1- utilizar e veicular as fotografias, vídeos e/ou voz obtidas durante minha participação da pesquisa anterior para a Dissertação de Mestrado do Programa de Pós-Graduação em Educação em Ciências e Matemática, para fim de obtenção de grau acadêmico (e/ou divulgação científica), garantido a ocultação de identidade (mantendo-se a confidencialidade e a privacidade das informações), inclusive, mas não restrito a ocultação da face e/ou dos olhos, quando possível;

2- utilizar as fotografias, vídeos e/ou voz na produção de quaisquer materiais acadêmicos, inclusive aulas e apresentações em congressos e eventos científicos, por meio oral (conferências) ou impresso (pôsteres ou painéis); na publicação de artigos científicos em meio impresso e/ou eletrônico para fins de divulgação, sem limitação de número de inserções e reproduções;

3- no caso de imagens, executar livremente a montagem das fotografias, realizando cortes e correções de brilho e/ou contrastes necessários, sem alterar a sua veracidade, utilizando-as exclusivamente para os fins previstos neste termo e responsabilizando-se pela guarda e pela utilização da obra final produzida;

4- no caso da voz, executar livremente a edição e montagem do trecho, realizando cortes e correções necessárias, assim como de gravações, sem alterar a sua veracidade, utilizando-as exclusivamente para os fins previstos neste termo e responsabilizando-se pela guarda e pela utilização da obra final produzida.

O participante declara que está ciente que não haverá pagamento financeiro de qualquer natureza neste ou em qualquer momento pela cessão das fotografias, dos vídeos e/ou da voz, e que está ciente que pode retirar seu consentimento, em qualquer fase da pesquisa, sem penalização alguma, salvo os materiais científicos já publicados.

É vedado ao(s) pesquisador(es) utilizar as fotografias, os vídeos e/ou a voz para fins comerciais ou com objetivos diversos da pesquisa proposta, sob pena de responsabilização nos

termos da legislação brasileira. O(s) pesquisador(es) declaram que o presente estudo/pesquisa será norteado pelos normativos éticos vigentes no Brasil.

O participante receberá uma cópia assinada e datada deste termo.

Araras, __ de _____ de 2023.

MATEUS XAVIER YAMAGUTI
PESQUISADOR RESPONSÁVEL
CPF: 446.521.938-58

PROF. DR JOÃO TELES DE CARVALHO NETO
ORIENTADOR
CPF: 269.991.558-01

PARTICIPANTE DO ESTUDO
CPF:

APÊNDICE I - SLIDES UTILIZADOS EM AULA

SLIDES AULA 01

FIS_PY

Aula 01

O CURSO

Este curso tem por objetivo ensinar os conhecimentos de cinemática com o auxílio de linguagem de programação Python, utilizando temas da vida cotidiana

SEGURANÇA NO TRÂNSITO

SEGURANÇA NO TRÂNSITO

Quem aqui já chegou
atrasado na escola?

SEGURANÇA NO TRÂNSITO

E por que vocês
chegaram atrasados?

**SEGURANÇA
NO TRÂNSITO**

Como vocês fariam para calcular o tempo mínimo para chegar na escola sem ultrapassar a velocidade máxima permitida?

**SEGURANÇA
NO TRÂNSITO**

Por que vocês acham que no semáforo entre a luz vermelha e a verde, tem a laranja?

**SEGURANÇA
NO TRÂNSITO**

O que aconteceria se não houvesse a luz laranja no semáforo?

SEGURANÇA NO TRÂNSITO

Qual a importância de sabermos calcular o espaço mínimo para frear quando a luz laranja acende?

TECNOLOGIA EM SERVIÇO

TECNOLOGIA EM SERVIÇO

Quando algum condutor de veículo ultrapassa o farol vermelho, como é feita essa fiscalização de forma automática?

**TECNOLOGIA
EM SERVIÇO**

O radar de velocidade, como que ele faz o cálculo da velocidade do veículo?

**TECNOLOGIA
EM SERVIÇO**

Feito esse cálculo, como que o radar comunica a central que houve uma infração de trânsito?

**TECNOLOGIA
EM SERVIÇO**

E qual linguagem vocês acham que o radar ou os meios tecnológicos utilizam para se comunicar?

PARA PENSAR

PARA PENSAR

Para vocês, qual seria a importância de entender a física nas leis de trânsito e como a tecnologia está incorporada nela?

ORIENTAÇÕES

ACESSO AO PORTAL AVA

Implementar as orientações de acesso ao portal AVA

ACESSO ÀS ATIVIDADES E MATERIAIS

Implementar as orientações de acesso atividades e materiais

INSTALANDO PYTHON



1) Acesse <https://www.python.org/> e clique em **Download**



2) Clique na versão de mais atualizada e realize o download



3) Execute o instalador com as opções apresentada na figura acima e prossiga até o final da instalação.

* Instalação no sistema windows: <https://python.org.br/instalacao-windows/>

** Instalação no sistema linux: <https://python.org.br/instalacao-linux/>

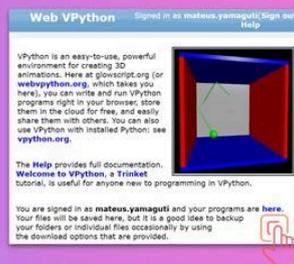
ACESSANDO GLOWSCRIPT



1) Acesse
<https://www.glowscript.org/>



2) Realize o login com uma
conta **gmail**



3) Clique em **Here** e acesse
o ambiente de
desenvolvimento

* Documentação: <https://www.glowscript.org/docs/VPythonDocs/index.html>

FIM

SLIDES AULA 02

FIS_PY

Aula 02

**VIAJANDO
COM A FÍSICA**

VIAJANDO COM A FÍSICA

Pra quem gosta de viajar, alguns detalhes como a distância da viagem ou o tempo gasto são coisas muito importante.

E pra você, quais detalhes são importantes para verificar antes de uma viagem?

O PODER DAS MEDIDAS

O PODER DAS MEDIDAS

Já se perguntou o motivo de haver padrões de medidas para diversos tipos de grandezas?

Física

As grandezas físicas são propriedades medíveis de um fenômeno, corpo e até mesmo de substâncias, as quais são padronizadas pelo Sistema Internacional de Unidades de medidas (SI). São alguns exemplos dessa unidades:

- Metro: Grandeza de espaço (m)
- Segundo: Grandeza tempo (s)
- Quilograma: Grandeza de massa (kg)

Python

Assim como na física o linguagem de programação Python também possui tipos de grandezas e valores, e algumas são chamados de variáveis primitivas. São elas:

- Int: Valores numéricos inteiros
- Float: Valores numéricos com casas decimais
- Strings: Caracteres ou letras
- Listas: Coleção de objetos
- Tuplas: Coleção de objetos imutáveis
- Dicionário: Coleção ordenada de objetos com padrão chave-valor

Física

A Corrida Internacional de São Silvestre é uma corrida de rua realizada anualmente na cidade de São Paulo, desde 1925 que possui atualmente 15 km de percurso. O ganhador da São Silvestre no ano de 2022 foi o atleta de Uganda "Andrew Rotich Kwemoi" de 22 anos que encerrou o percurso em 44min43s.

Entre as informações do texto podemos elencar algumas grandezas, como distância e tempo.

Python

Essas grandezas físicas podem ser atribuídas a variáveis na linguagem de programação Python. Como por exemplo:

```
>>> idade = 22
>>> type(idade)
<class 'int'>
>>> idade
22
```

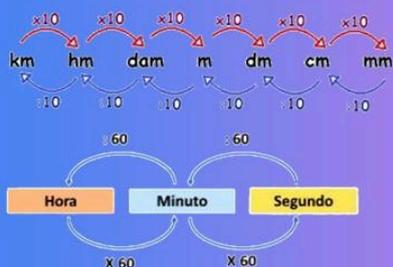
OS SUBMÚLTIPLOS DAS MEDIDAS

OS SUBMÚLTIPLOS DAS MEDIDAS

Já tentou medir uma viagem de carro em metros, ou contar o tempo gasto nessa viagem em segundos?

Física

Para facilitar o trabalho de medir grandezas foi criados seus múltiplos e submúltiplos, que no SI variam em uma determinada escala. Por exemplo:



Python

Esses múltiplos e submúltiplos podem ser atribuídos a variáveis, ou até mesmo criados a partir de operações matemáticas simples. Como por exemplo:

```
>>> metro = 1
>>> centimetro = metro / 100
>>> centimetro
0.01
>>> horas = 2
>>> segundos = horas * 60
>>> segundos
120
```

Física

Além de saber medir as grandezas, em muitos momentos precisamos saber manipulá-las, como por exemplo: Somar distâncias, subtrair uma quantidade de massa de um recipiente, calcular áreas e volumes e também dividir para conquistar.

E para conquistar essa habilidade nada melhor do que uma mãozinha da linguagem de programação Python

Python

Os operadores matemáticos básicos em Python são:

- + Adição
- Subtração
- * Multiplicação
- ** Potenciação
- / Divisão

```
>>> distancia_do_passeio_01 = 100
>>> distancia_do_passeio_02 = 250
>>> distancia_total = distancia_do_passeio_01
+ distancia_do_passeio_02
>>> distancia_total
350
```

TRABALHANDO COM GRANDEZAS

TRABALHANDO COM GRANDEZAS

Quem nunca teve curiosidade de conhecer as Pirâmides do Egito, a Torre Eiffel na França ou até mesmo a Torre de Pisa na Itália e ver de pertinho suas dimensões?

Física

Em uma viagem à torre de Pisa na Itália Gabriel teve curiosidade de descobrir por conta própria as dimensões da Torre de Pisa. Primeiramente ele mediu que o diâmetro interno da base da torre possui cerca de nove passos largos (0,8 m). Depois estipulou por comparação com sua altura que os oito andares da torre possuem uma em médio 7 metros de altura.

Como Gabriel tinha várias cálculos para serem feitos ele resolveu utilizar a linguagem de programação Python para ajudá-lo.

Python

Na linguagem de programação qualquer tipo de cálculo matemático pode ser traduzido para um computador, sempre respeitando a ordem das operações como foi feito por Gabriel:

```
>>> raio_da_torre = 0.8 * 4.5
>>> altura_torre = 8 * 7
>>> area_da_base = 3.14*raio_da_torre**2
>>> area_da_base
40.6944
>>> volume_da_torre = area_da_base * altura_torre
>>> volume_da_torre
2278.8864000000003
```

Física

Nesta viagem Gabriel visitou vários pontos turísticos e para não se atrapalhar com o tempo de deslocamento ele recorreu a alguns conhecimentos físicos de cinemática e também produziu alguns programas em Python para ajudá-lo a fazer os cálculos.

Primeiramente Gabriel usou o conceito de deslocamento, sendo este a diferença do ponto de partida com o ponto de chegada do percurso:

$$\Delta S = S_f - S_i$$

Python

Para poder reutilizar o código quantas vezes ele quisesse, Gabriel utilizou o conceito de função em Python e construiu o seguinte script:

```
def DeslocamentoDoGabriel(d_final, d_inicial):  
    d_total = d_final - d_inicial  
    return d_total  
print(DeslocamentoDoGabriel(253, 122))
```

Física

Além disso, Gabriel também começou a medir o tempo gasto durante os percursos, que corresponde à **diferença do tempo final e o tempo inicial** da partida.

$$\Delta t = t_f - t_i$$

Python

Para poder deixar mais eficiente as contas Gabriel também escreveu o seguinte código para o tempo do percurso:

```
def TempoDeDeslocamento(t_final, t_inicial):  
    t_total = t_final - t_inicial  
    return t_total  
print(TempoDeDeslocamento(653, 142))
```

Física

Por fim, Gabriel teve a surpreendente ideia de calcular a velocidade média de seus passeios para otimizar o tempo gastos em suas próximas viagens. Para isso, ele calculou a velocidade média de seus passeios que corresponde a **variação da distância percorrida pelo período de tempo** gasto.

$$v_m = \frac{\Delta S}{\Delta t}$$

Python

Como foram muitos passeios Gabriel criou um função em Python para retornar a velocidade média dos passeios e assim também agregou uma tolerância de duas casas decimais para ter mais precisão.

```
def VelocidadeMedia(deslocamento, tempo):  
    v_medio = deslocamento / tempo  
    return round(v_medio, 2)  
print(VelocidadeMedia(150, 33))
```

FIM

SLIDES AULA 03

FIS_PY

Aula 03

**SIMULADOR
DE VIAGENS**

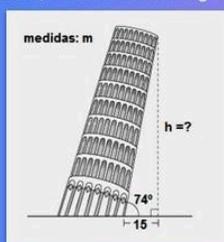
SIMULADOR DE VIAGENS

Quando se tem poucas chances para fazer algo dar certo, qual a melhor estratégia antes da execução?

O PODER DA MATEMÁTICA

Física

Não contente com sua estratégia de descobrir as intenções da torre de Pisa, Gabriel resolveu testar um novo método. Ele foi ao meio dia até a torre e mediu a sua sombra e o ângulo que a torre fazia com o solo, conforme a imagem abaixo



Python

Para ser mais preciso no cálculo Gabriel utilizou da fórmula matemático da tangente aliado a biblioteca **math** do Python

$$\tan \theta = \frac{CO}{CA}$$

```
import math

def Calculo_Do_Cateto_Oposto_Com_Tangente(l_adjacente, teta):
    angulo_graus = math.radians(teta)
    hipotenusa = l_adjacente * math.tan(angulo_graus)
    return hipotenusa

print(Calculo_Do_Cateto_Oposto_Com_Tangente(15, 74))
```

SÓ ACREDITO VENDO

Física

Gabriel quer planejar muitas outras viagens e para isso resolveu simular visualmente os seu deslocamento.

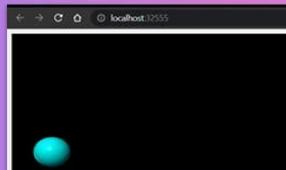
Para ficar mais fácil inicialmente ele quis representar a si próprio como uma partícula esférica.

Python

E para poder exemplificar isso visualmente ele escolheu utilizar a biblioteca VPython que permite a construção de objetos 3D, conforme a ilustração abaixo:

```
from vpython import *

esfera = sphere(pos=vector(-5, 0, 0),
                radius=0.5, color=color.cyan)
```



Física

Imaginando inicialmente que sua velocidade é retilínea e uniforme, Gabriel criou uma animação simples que a cada intervalo de tempo sua representação esférica sofre um determinado deslocamento

Python

Para realizar a animação Gabriel utilizou o conceito do laço While em Python, que permite que um comando seja executado até que a condição aferida nesse comando não seja mais satisfeita, conforme exemplo abaixo:

```
from vpython import *
from time import sleep

esfera = sphere(pos=vector(-5, 0, 0),
                radius=0.5, color=color.cyan)
parede = box(pos=vector(6, 0, 0),
             size=vector(0.2, 12, 12), color=color.green)

while esfera.pos.x <= 10:
    sleep(1)
    esfera.pos.x += 1
```

Física

Sabendo que em alguns momentos na viagem é necessário acelerar para chegar a tempo nos pontos turístico Gabriel percebe a necessidade de fazer o cálculo do Movimento Uniformemente Variado.

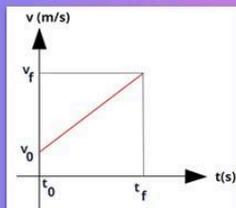
Ou seja, o movimento que existe **aceleração**, a qual **representa a variação da velocidade pelo tempo**.

$$a = \frac{\Delta V_m}{t}$$

E manipulando a equação da aceleração podemos chegar que a velocidade final de um movimento acelerado é:

$$v_f = v_0 + a \times t$$

Enquanto o gráfico da variação da velocidade pelo tempo, nos dá a informação do espaço percorrido



Levando em conta que t inicial é igual a zero e substituindo a velocidade final pela sua equação

$$\Delta s = s - s_0 = \frac{v_0 + v_0 + a \cdot t}{2} \cdot t$$

$$s - s_0 = \frac{2 \cdot v_0 \cdot t + a \cdot t^2}{2}$$

$$s - s_0 = \frac{2 \cdot v_0 \cdot t}{2} + \frac{a \cdot t^2}{2}$$

$$s - s_0 = v_0 \cdot t + \frac{a \cdot t^2}{2}$$

Python

Considerando que o tempo inicial e a velocidade inicial iguais a zero, e a aceleração igual a 10 m/s², Gabriel chegou na seguinte fórmula.

$$S = \frac{10t^2}{2}$$

O que resultou no seguinte programa em Python

```
def MovimentoUniformementeVariado():
    esfera = sphere(pos = vector(0,0,0))
    chao = box(pos=vector(60, 0, 0), color=color.cyan, size=vector(130, 0.5, 15))
    t = 0
    dt = 0.01
    aceleracao = 10

    while t <= 5:
        sleep(0.01)
        esfera.pos.x = (aceleracao*t**2)/2
        t += dt
        print(round(esfera.pos.x,1))

MovimentoUniformementeVariado()
```

FIM

SLIDES AULA 04

FIS_PY

Aula 04

**ACERTE A
CESTA**



ACERTE A CESTA

No ano de 2021 a média de arremessos de três pontos convertidos na temporada da NBA foi de 12,7

ACERTE A CESTA

O jogador Stephen Curry tem uma média de 8.3 arremesso de 3 convertido por jogo. Sendo que hoje ele é o maior arremessador da NBA, tendo em sua carreira um total de 3390 pontos nos arremessos de três pontos.

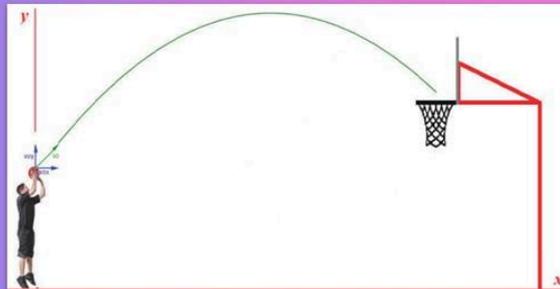
Será que ele tem algum segredo? Como a física poderia nos ajudar a descobrir?



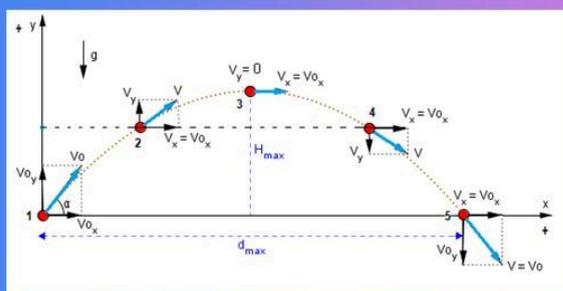
O ARREMESSO PERFEITO

O ARREMESSO PERFEITO

O arremesso como o realizado no basquete pode ser classificado como um lançamento oblíquo, o qual pode ser decomposto por um movimento na horizontal e outro na vertical.



O ARREMESSO PERFEITO



Para esse tipo de movimento consideramos a velocidade do objeto na horizontal como um MRU.

Enquanto na vertical, podemos considerar a velocidade do objeto como um MRUV em que a aceleração é o valor da aceleração gravitacional (9.8 m/s^2 aproximadamente)

DECOMPONDO O ARREMESSO

DECOMPONDO O ARREMESSO

MUV

$$s = s_0 + v_0 t + \frac{a}{2} t^2$$

$$v = v_0 + at$$

$$v^2 = v_0^2 + 2a \cdot \Delta s$$

Lançamento Vertical

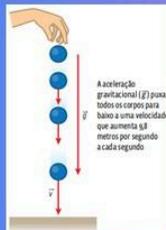
$$h = h_0 + v_0 t - \frac{gt^2}{2}$$

$$v = v_0 + gt$$

$$v^2 = v_0^2 - 2g \cdot \Delta h$$

Física

Se soltarmos uma esfera de uma altura h , e implicamos que não existem outras forças sobre ela, podemos deduzir que ela sofrerá alguma mudança em seu MRUV apenas quando colidir com o chão.



Python

Como a esfera pode ser lançada das mais diversas distâncias do solo, foi criada uma função em Python que aceita como parâmetro a altura de lançamento da esfera e reproduz uma animação do movimento.

Para poder exemplificar isso, dentro dessa função foi utilizado a estrutura de decisão em Python, para que se a esfera tocasse o chão ela pudesse parar.

Estrutura de decisão

```
if condição == True:
    # Se condição for verdadeira
    # Executa o comando
    '''Comando'''
else:
    # Se não for verdadeiro
    # Não executa outro comando
    '''Comando'''
```

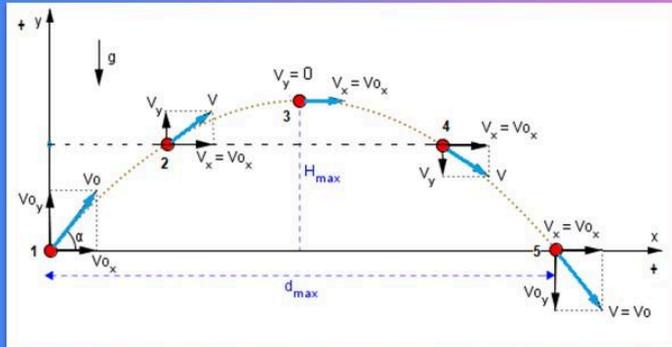
Exemplo

```
def MovimentoUniformementeVariado(altura):
    esfera = sphere(pos = vector(0,0,0))
    chao = box(pos=vector(0, -(altura), 0),
              color=color.cyan, size=vector(130, 0.5, 15))

    t = 0
    dt = 0.01
    aceleracao = 9.8

    while True:
        sleep(0.01)
        esfera.pos.y = -(aceleracao*t**2)/2
        t += dt
        if esfera.pos.y < -(altura):
            break
    print(round(esfera.pos.y, 1))
```

DECOMPONDO O ARREMESSO



$$v_x = v \cos \alpha$$

$$v_y = v \sin \alpha$$

Horizontal

$$v_x = cte$$

$$x = v \cos \alpha \cdot t$$

Vertical

$$v_y = v \sin \alpha \cdot t - gt$$

$$y = v \sin \alpha \cdot t - \frac{gt^2}{2}$$

Física

Utilizando as equações do lançamento oblíquo podemos produzir a seguinte animação em Python.

$$v_x = v \cos \alpha$$

$$v_y = v \sin \alpha$$

Horizontal

$$v_x = cte$$

$$x = v \cos \alpha \cdot t$$

Vertical

$$v_y = v \sin \alpha \cdot t - gt$$

$$y = v \sin \alpha \cdot t - \frac{gt^2}{2}$$

Python

```
from vpython import *
# Criação do objeto 3D
stone = sphere(pos=vector(0, 0, 0), color=color.green, make_trail=True)
floor = box(pos=vector(0, 0, 0), size=vector(150, 0.5, 10))
# Criação do gráfico
graphic = gcurve(color=color.cyan)
# Instanciando velocidade inicial e ângulo de lançamento
v = 42
ang = radians(40)
# Decompondo a velocidade no eixo x e y
vx = v*cos(ang)
vy = v*sin(ang)
# Atribuindo velocidade para o objeto stone
stone.velocity = vector(vx, vy, 0)
# Instanciando variáveis temporais
t = 0
dt = 0.01
# Laço de animação
while t <= 5.5:
    rate(100)
    # Movimento MRU na horizontal
    stone.pos.x = stone.pos.x + stone.velocity.x*dt
    # Movimento MRU na vertical
    stone.pos.y = stone.pos.y + stone.velocity.y*dt
    # y = y[inicial] + v*t
    stone.velocity.y = stone.velocity.y - 9.8*dt
    # v(y) = v[inicial] - 9.8*t
    # Plotando o gráfico
    graphic.plot(stone.pos.x, stone.pos.y)
    # Passo temporal
    t = t + dt
```

FIM

APÊNDICE J - ATIVIDADES

QUESTÕES AULA 02

Questão 1

Incorreto

Vale 1,00 ponto(s).

[Marcar questão](#)

[Editar questão](#)

Escreva uma função de nome `SubMultMetro`, que receba como parâmetro um número que represente a quantidade de metros e retorne seu submúltiplo em centímetros, conforme exemplo abaixo:

```
In [ ]:SubMultMetro(15)
Out[ ]:1500
```

Answer: (penalty regime: 0 %)

```
1
2
```

Questão 2

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)

[Editar questão](#)

Escreva uma função de nome `MostrarTipoDaVariavel`, que receba qualquer valor como parâmetro e retorne o tipo ou a classe que esse parâmetro representa em python, conforme exemplo abaixo.

```
In [ ]:MostrarTipoDaVariavel(10)
Out[ ]:<class 'int'>
```

Answer: (penalty regime: 0 %)

```
1
```

Questão 3

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Cria uma função de nome `CalcularHorasParaMinutos`, que receba como parâmetro um número que representa a quantidade de horas e retorne essa quantidade em minutos, conforme o exemplo abaixo

```
In [ ]:CalcularHorasParaMinutos(10)
Out[ ]:10 horas equivale a 600 minutos
```

Answer: (penalty regime: 0 %)

1

Questão 4

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Crie uma função de nome `CalcularMultMetro`, que receba como parâmetro um número que represente a quantidade de metros e retorne seu múltiplo em quilômetros, conforme exemplo abaixo

```
In [ ]:CalcularMultMetro(5000)
Out[ ]:5000 metros equivale a 5 km
```

Answer: (penalty regime: 0 %)

1

Questão 5

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Crie uma função de nome `CalcularDiametroDaEsfera`, que receba como parâmetro um número que representa o valor do raio da esfera em centímetros e retorne o valor do seu diâmetro em centímetros, conforme exemplo abaixo:

```
In [ ]:CalcularDiametroDaEsfera(5)
Out[ ]:10
```

Answer: (penalty regime: 0 %)

1

Questão 6

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Crie uma função de nome `CalcularAreaDaEsfera`, que receba como parâmetro um número que representa o valor do raio da esfera em centímetros e retorne o valor da sua área superficial em centímetro, conforme exemplo abaixo (obs: Sendo π igual a 3,14 com apenas duas casas de tolerância)

```
In [ ]:CalcularAreaDaEsfera(4)
Out[ ]:200.96
```

Answer: (penalty regime: 0 %)

1

Questão 7

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Crie uma função de nome `CalcularVolumeDaEsfera`, que receba um número que representa o valor do raio da esfera em centímetros e retorne o seu o valor do seu volume em centímetros cúbicos, conforme exemplo abaixo (obs: Sendo π igual a 3,14 com apenas duas casas de tolerância)

```
In [ ]: CalcularVolumeDaEsfera(4)
Out[ ]: 200.96
```

Answer: (penalty regime: 0 %)

1

Questão 8

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Sabendo que densidade de um objeto é a razão entre sua massa e o seu volume ($d = m / V$), crie uma função de nome `CalcularDensidade` que receba como parâmetro dois números que representem a massa e o volume, e retorne o valor da densidade do objeto em g/cm^3 , conforme exemplo abaixo.

```
In [ ]: CalcularDensidade(10, 100)
Out[ ]: 0.1
```

Answer: (penalty regime: 0 %)

1

Questão 9

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Sabendo que o deslocamento de um móvel é a diferença entre o seu ponto de partida e o de chegada, crie uma função de nome **CalcularDeslocamento** que receba como parâmetro dois números que representem o valor do ponto de partida e o de chegada, e retorne o valor do deslocamento do móvel entre os ponto, conforme exemplo abaixo

```
In [ ]: CalcularDeslocamento(10, 100)
Out[ ]: 90
```

Answer: (penalty regime: 0 %)

1

Questão 10

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Sabendo que a velocidade uniforme de um móvel em linha reta é a razão do espaço deslocado pelo intervalo de tempo ($v = S / t$). Crie uma função de nome **CalcularVelocidade** que receba como parâmetro dois números que representem o valor do deslocamento do móvel e o período de tempo, e retorne o valor da velocidade em metros por segundo, conforme exemplo abaixo.

```
In [ ]: CalcularVelocidade(100,10)
Out[ ]: 10.0
```

Answer: (penalty regime: 0 %)

1

QUESTÕES AULA 03

Questão 1

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)

[Editar questão](#)

Crie uma função de nome `CalcularDensidadeDaEsfera` que receba como parâmetro dois números que representam a massa e o raio da esfera, e retorne o valor da densidade da esfera em g/cm^3 com duas casas decimais, conforme exemplo abaixo.

```
In [ ]: CalcularDensidadeDaEsfera(500, 4)
Out[ ]: 1.87
```

Answer: (penalty regime: 0 %)

1

Questão 2

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)

[Editar questão](#)

Crie uma função de nome `CalcularVelocidadeEntrePontos` que receba como parâmetros três valores numéricos que representem o ponto de partida, ponto de chegada e período de tempo, e retorne o valor da velocidade do móvel em m/s com uma casa decimal, conforme exemplo abaixo.

```
In [ ]: CalcularVelocidadeEntrePontos(200, 400, 10)
Out[ ]: 20.0
```

Answer: (penalty regime: 0 %)

1

Questão 3

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Crie uma função de nome

CalcularDiferancaVelocidadeEntrePontos que receba como parâmetros dois valores numéricos que representem a velocidade do móvel no ponto inicial e ponto final, e retorne o valor da diferença de velocidade entre os pontos em m/s com uma casa decimal se houver necessidade, conforme exemplo abaixo.

```
In [ ]: CalcularDiferancaVelocidadeEntrePontos(90, 100)
```

```
Out[ ]: 20.0
```

Answer: (penalty regime: 0 %)

```
1
```

Questão 4

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Crie uma função de nome **CalcularAceleracao** que recebe como parâmetros dois valores numéricos que representem a diferença de velocidade do móvel e o período percorrido por ele, e retorne o valor da sua aceleração em m/s² com uma casa decimal, conforme exemplo abaixo.

```
In [ ]: CalcularAceleracao(100,10)
```

```
Out[ ]: 10.0
```

Answer: (penalty regime: 0 %)

```
1
```

Questão 5

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Crie uma função que represente a função horária do espaço escalar para um movimento uniformemente variado (MUV), em que o móvel comece a se deslocar sempre no instante inicial igual a zero, e a posição inicial igual a zero. Sendo assim, crie uma função de nome **CalcularEspaloMUV** que receba como parâmetro dois valores numéricos que representem a aceleração do móvel e o período de deslocamento, e retorne o valor do espaço em metros com uma casa decimal, conforme exemplo abaixo:

```
In [ ]: CalcularEspaloMUV(9.8, 10)
```

```
Out[ ]: 490.0
```

Answer: (penalty regime: 0 %)

1

Questão 6

Resposta salva

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Uma esfera com o atributo velocidade igual a `esfera.velocity = vector(25, 0, 0)`, se desloca durante 3 segundos, qual é a distância percorrida por essa esfera e em qual eixo?

- a. distância = 75, eixo x
- b. distância = 65, eixo y
- c. distância = 75, eixo y
- d. distância = 85, eixo z
- e. distância = 100, eixo x

[Limpar minha escolha](#)

Questão 7

Ainda não respondida

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Foi criada uma pista de corrida verde para uma esfera (`esfera.velocity = vector(25, 0, 0)`) na posição $x = 32.5$, $y = 0$ e $z = 0$, para se locomover durante 3 segundos. Sendo assim, preencha as lacunas com dados corretos

```
In [ ]: pista = box(pos=vector( [ ] , [ ] , [ ] ), size=vector( [ ] , 12, 0.2), color=color.[ ] )
```


Questão 8

Resposta salva

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Preencha as lacunas para que o código seja executado sem erro:

```
from vpython import *

esfera = sphere(pos=[ ] (-5, 0, 0), [ ]
=0.5, color=color.cyan)
parede = box(pos=vector(32.5, 0, 0),
size=vector(75, 12, 0.2), color=[ ] .green)
esfera.[ ] = vector(25, 0, 0)
deltat = 0.005
t = 0

[ ] t < 3:
    rate(100)
    esfera.pos = esfera.pos +
esfera.velocity*deltat
    t = t + deltat
```

Questão 9

Resposta salva

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Preencha as lacunas abaixo para que a função crie a esfera com sucesso e retorne a posição do eixo x.

```
from vpython import *

[ ] CriarEsfera(raio):
    esfera = sphere(pos=vector(0, 0, 0), [ ]
=raio, color=color.cyan)
    [ ] esfera.pos. [ ]
```

def

return

x

radius

Questão 10

Ainda não respondida

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Um estudante desejou encapsular o deslocamento de uma esfera em uma função, que resultou no código abaixo. O código finaliza retornando a posição final da esfera. Qual seria essa posição?

```
from vpython import *

def DeslocamentoDaEsfera():
    esfera = sphere(pos=vector(-5, 0, 0), radius=0.5, color=color.cyan)
    parede = box(pos=vector(32.5, 0, 0), size=vector(75,12,0.2),
color=color.green)
    esfera.velocity = vector(25, 0, 0)
    deltat = 0.005
    t = 0
    while t < 2.995:
        rate(100)
        t = t + deltat
        esfera.pos = esfera.pos + esfera.velocity*deltat
    return esfera.pos.x

DeslocamentoDaEsfera()
```

- a. 70
- b. 75
- c. 71
- d. 69
- e. 65

QUESTÕES AULA 04

Questão 1

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)

[Editar questão](#)

Durante um espirro, os olhos podem se fechar por até 0,50 s. Se você está dirigindo um carro a 90km/h e espirra, o quanto o carro pode se deslocar até você abrir novamente os olhos?

Para resolver essa questão crie uma função de `deslocation_by_blink_of_an_eye` que não passa nenhum parâmetro e retorna o valor do deslocamento do carro enquanto os olhos estão fechados.

```
In [ ]: deslocation_by_blink_of_an_eye()  
Out[ ]: 0.0125
```

Answer: (penalty regime: 0 %)

1

Questão 2

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Essa questão 2 tem 4 subquestões (a, b, c e d) que deveram ser respondidas individualmente

2) Calcule a velocidade média nos dois casos: a) Caso 1: você caminha 73,2 m a uma velocidade de 1,22 m/s e depois corre 73,2 m a 3,05 m/s, em uma pista reta; b) Caso 2: você caminha 1,00 min com uma velocidade de 1,22 m/s e depois corre por 1,00 min a 3,05 m/s em uma pista reta. c) Faça o gráfico de x em função de t nos dois casos e indique a velocidade e por fim, (d) faça a animação da velocidade das partículas sobre a pista de corrida.

a) Para resolver a questão “a” crie uma função de nome `average_speed_in_walk_and_run_of_73_meters_sub_question_1` que não passa nenhum parâmetro e retorna o valor da velocidade média do percurso inteiro do primeiro caso, com duas casas decimais.

```
In [ ]:
average_speed_in_walk_and_run_of_73_meters_sub_questi
Out[ ]: 1.74
```

b) Para resolver a questão “b” crie uma função de nome `average_speed_in_walk_and_run_of_73_meters_sub_question_2` que não passa nenhum parâmetro e retorna o valor da velocidade média do percurso inteiro do segundo caso, com duas casas decimais.

```
In [ ]:
average_speed_in_walk_and_run_of_73_meters_sub_questior
Out[ ]: 2.13
```

Answer: (penalty regime: 0 %)

1

Questão 4

Ainda não respondida

Vale 1,00 ponto(s)

Marcar questão

Editar questão

A questão 2 tem 4 subquestões (a, b, c e d) que deveram ser respondidas individualmente

2) Calcule a velocidade média nos dois casos: a) Caso 1: você caminha 73,2 m a uma velocidade de 1,22 m/s e depois corre 73,2 m a 3,05 m/s, em uma pista reta; b) Caso 2: você caminha 1,00 min com uma velocidade de 1,22 m/s e depois corre por 1,00 min a 3,05 m/s em uma pista reta. c) Faça o gráfico de x em função de t nos dois casos e indique a velocidade e por fim, (d) faça a animação da velocidade das partículas sobre a pista de corrida.

c) Para resolver a questão "c" foi criada uma função de nome `average_speed_in_walk_and_run_of_73_meters_sub_question_c` que não passa nenhum parâmetro e retorna o valor do percurso total percorrida e também apresente o gráfico do objeto conforme o exemplo abaixo. Sendo assim, preencha as lacunas da função para que ela execute o movimento correto.

```
def average_speed_in_walk_and_run_of_73_meters_sub_question_c():
    part1 = sphere(pos=vector(0, 0.5, 0), radius=2, color=color.red)
    part2 = sphere(pos=vector(0, 0.5, 0), radius=2, color=color.green)
    floor = box(pos=vector(80, 0, 0), size=vector(360, 0.5, 10))

    '''Criação de variável temporal e passo'''
    dt = 0.01
    t = 0

    '''Velocidade inicial'''
    part1.velocity = vector(1.22, 0, 0)
    part2.velocity = vector(1.22, 0, 0)

    part1_trajectory = gcurve(color=color.green)

    '''Fator de animação'''
    ft = 10

    '''Criação do laço onde ocorre a animação'''
    while t < 84:

        rate(100)

        if part1.pos.x < 146.4:
            part1_trajectory.plot(t * ft, part1.pos.x)
```

```
part1.pos.x <= 73.2:
    part1.pos.x += part1.velocity.x * dt * ft

elif part1.pos.x > 73.2 and part1.pos.x < 146.4:
    part1.velocity = vector(3.05, 0, 0)
    part1.pos.x += part1.velocity.x * dt * ft

t += dt * ft

t = 0

t < 120:

rate(100)
part2_trajectory.plot(t, part2.pos.x)

if t <= 60:
    part2.pos.x += part2.velocity.x * dt * ft

t > 60 and t < 120:
    part2.velocity = vector(3.05, 0, 0)
    part2.pos.x += part2.velocity.x * dt * ft

t += dt * ft
```

gcurve elif radius if vector while velocity

Questão 5

Ainda não respondida

Vale 1,00 ponto(s)

Marcar questão

Editar questão

A questão 2 tem 4 subquestões (a, b, c e d) que deveram ser respondidas individualmente

2) Calcule a velocidade média nos dois casos: a) Caso 1: você caminha 73,2 m a uma velocidade de 1,22 m/s e depois corre 73,2 m a 3,05 m/s, em uma pista reta; b) Caso 2: você caminha 1,00 min com uma velocidade de 1,22 m/s e depois corre por 1,00 min a 3,05 m/s em uma pista reta. c) Faça o gráfico de x em função de t nos dois casos e indique a velocidade e por fim, (d) faça a animação da velocidade das partículas sobre a pista de corrida.

d) Para resolver a questão "d" crie uma função de nome `average_speed_in_walk_and_run_of_73_meters_sub_question_d` que simule as duas partículas se deslocando conforme a imagem abaixo:

```
def average_speed_in_walk_and_run_of_73_meters_sub_question_c():
    part1 = sphere(pos=vector(0, 0.5, 0), radius=2, color=color.red)
    part2 = sphere(pos=vector(0, 0.5, 0), radius=2, color=color.green)
    floor = box(pos=vector(80, 0, 0), size=vector(360, 0.5, 10))

    '''Criação de variável temporal e passo'''
    dt = 0.01
    t = 0

    '''Velocidade inicial'''
    part1.velocity = vector(1.22, 0, 0)
    part2.velocity = vector(1.22, 0, 0)

    '''Gráfico'''
    part1_trajectory = gcurve(color=color.red)
    part2_trajectory = gcurve(color=color.green)

    '''Fator de animação'''
    ft = 10

    '''Criação do laço onde ocorre a animação'''
    while t < 84:

        rate(100)

        if part1.pos.x < 73.2:
            part1_trajectory.plot(t * ft, part1.pos.x + ft)
```

```
if part1.pos.x <= 73.2:
    part1_trajectory.plot(t * ft, part1.pos.x + ft)

elif part1.pos.x > 73.2 and part1.pos.x < 146.4:
    part1.velocity = vector(3.05, 0, 0)
    part1_trajectory.plot(t * ft, part1.pos.x + dt * ft)

t += dt * ft

t = 0

while t < 120:

    rate(100)
    part2_trajectory.plot(t, part2.pos.x)

    if t <= 60:
        part2.pos.x += part2.velocity.x * dt * ft

    elif t > 60 and t < 120:
        part2.velocity = vector(3.05, 0, 0)
        part2_trajectory.plot(t, part2.pos.x + dt * ft)

    t += dt * ft
```

gcurve part1.velocity.x dt 120 velocity 60

Questão 6

Resposta
incompletaVale 1,00
ponto(s).[Marcar
questão](#)[Editar
questão](#)Essa questão 3 tem subquestões (a, b, c, d, e, f e g) que
deveram ser respondidas

3) A posição de um objeto que se move ao longo do eixo x é dado por $x = 3t - 4t^2 + t^3$, onde x está em metros e t em segundo. Determine a posição do objeto para os seguintes valores de t : (a) 1s, (b) 2s, (c) 3s, (d) 4s. (e) Qual é o deslocamento do objeto entre $t=0$ e $t=4s$? (f) Qual é a velocidade média para o intervalo de tempo de $t=2s$ a $t=4s$? (g) Desenhe o gráfico de x em função de t para $0 \leq t \leq 4s$.

a, b, c e d) Para resolver essas quatro (4) questões crie uma única função de nome

`question_several_answers_about_position_in_relation_by_time_a_b_c_d` que passe como parâmetro o tempo em segundos e retorne a posição da partícula conforme a função dada na questão e no exemplo abaixo.

```
In [ ]:
]: several_answers_about_position_in_relation_by_time_a_b_c_d(10)
Out[ ]: 630
```

Answer: (penalty regime: 0 %)

1

Questão 7

Incompleto

Vale 1,00
ponto(s).[Marcar
questão](#)[Editar
questão](#)Essa questão 3 tem subquestões (a, b, c, d, e, f e g) que
deveram ser respondidas

3) A posição de um objeto que se move ao longo do eixo x é dado por $x = 3t - 4t^2 + t^3$, onde x está em metros e t em segundo. Determine a posição do objeto para os seguintes valores de t : (a) 1s, (b) 2s, (c) 3s, (d) 4s. (e) Qual é o deslocamento do objeto entre $t=0$ e $t=4s$? (f) Qual é a velocidade média para o intervalo de tempo de $t=2s$ a $t=4s$? (g) Desenhe o gráfico de x em função de t para $0 \leq t \leq 4s$.

e) Para resolver a questão (e) crie uma função de nome

`distance_between_position_e(time_1, time_2)` que passe como parâmetros dois tempos em segundos e retorne o valor do deslocamento da partícula nesse intervalo de tempo.

```
In [ ]: distance_between_position_e(10, 20)
Out[ ]: 5830
```

Answer: (penalty regime: 0 %)

1

Questão 8

Incompleto

Vale 1,00 ponto(s).

[Marcar questão](#)[Editar questão](#)

Essa questão 3 tem subquestões (a, b, c, d, e, f e g) que deveram ser respondidas

3) A posição de um objeto que se move ao longo do eixo x é dado por $x = 3t - 4t^2 + t^3$, onde x está em metros e t em segundo. Determine a posição do objeto para os seguinte valores de t : (a) 1s, (b) 2s, (c) 3s, (d) 4s. (e) Qual é o deslocamento do objeto entre $t=0$ e $t=4s$? (f) Qual é a velocidade média para o intervalo de tempo de $t=2s$ a $t=4s$? (g) Desenhe o gráfico de x em função de t para $0 \leq t \leq 4s$.

f) Para resolver a questão (f) crie uma função de nome `average_speed_between_two_positions_f(time_1, time_2)` que passe como parâmetros dois tempos em segundos e retorne a velocidade média para esse intervalo de tempo.

```
In [ ]: average_speed_between_two_positions_f(10, 20)
Out[ ]: 583.0
```

Answer: (penalty regime: 0 %)

1

Questão 9

Ainda não respondida

Vale 1,00 ponto(s).

[Marcar questão](#)

[Editar questão](#)

A questão 2 tem 4 subquestões (a, b, c e d) que deveram ser respondidas individualmente

2) Calcule a velocidade média nos dois casos: a) Caso 1: você caminha 73,2 m a uma velocidade de 1,22 m/s e depois corre 73,2 m a 3,05 m/s, em uma pista reta; b) Caso 2: você caminha 1,00 min com uma velocidade de 1,22 m/s e depois corre por 1,00 min a 3,05 m/s em uma pista reta. c) Faça o gráfico de x em função de t nos dois casos e indique a velocidade e por fim, (d) faça a animação da velocidade das partículas sobre a pista de corrida.

g) Para resolver a questão (g) crie uma função de nome `create_graph_of_position_versus_time()` que não passa nenhum parâmetro, mas que apresenta o gráfico da função $x = 3t - 4t^2 + t^3$ e retorne a posição da partícula quando $t = 4$ segundos com apenas uma casa decimal.

```
def create_graph_of_position_versus_time():

    obj =  (0, 0, 0)
    time = 0
    dt = 0.01
    graphic =  (color=color.cyan)

     time ≤ 4.0:

        x_position_obj = (time*3 - (4*( **2)) + time**3)
        obj.x = x_position_obj
        graphic.plot(time, obj.x)
        time += 

    return round((obj.x), 1)
```

while vector dt time gcurve