MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO

Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME

Professor: Bruno Camargo Braga

Componente Curricular: Física

Unidade temática: Investigação Científica e Processos Criativos.

Tema de estudo: Primeira interação com a linguagem de programação Python

Grupos 1 e 2: 1º ano do Ensino Médio	**Tempo:** 2 horas	**Data:** 22/04/2025

COMPETÊNCIAS E HABILIDADES

(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos;
(EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.

OBJETIVOS DE APRENDIZAGEM

Utilizando a linguagem de programação Python (versão 3.11), ser capaz de realizar:

1. Operações numéricas;
2. Manipulação de textos e uso de comentários;
3. Operações com listas (tanto de números quanto de textos).

RECURSOS/MATERIAIS DIDÁTICOS

Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.

PROCEDIMENTOS METODOLÓGICOS

1º momento (5 min): Saudações iniciais, chamada e anotação da ata de sala

2º momento (25 min): O convite dos alunos na turma do Google Classroom deveria ser feito préviamente, porém possivelmente muitos alunos não responderam ao convite. Tome esse começo da aula para terminar de colocar todos os alunos presentes na turma, e ajude-os a baixar os aplicativos necessários – Google Classroom e Pydroid 3 (para quem usar Android). Indique aos alunos com iphone como usar o *notebook* do Google Colab.

No começo da aula será discutido brevemente o formato adotado das aulas, como os alunos serão avaliados e o temas que serão abordados nas aulas. Após essa introdução é explicado como que os alunos terão acesso ao Python 3, seguindo o modelo sobre o uso direto e o uso local.

Uso direto sem instalação (aplicação pela nuvem):

Tanto no computador quanto no celular podemos utilizar o Python sem instalar qualquer aplicativo, conseguimos isso devido a sites, que, dentre os mais confiáveis, temos o **Google Colab**.

O problema de opções como o Google Colab é a dependência de conexão com a internet, além de possíveis falhas devido ao congestionamento da conexão ou super lotação do servidor. Para não lidar com esse possível problema, precisamos entender como instalar localmente o Python e todas as bibliotecas não padrões necessárias às aulas.

Uso local (instalando aplicativo e bibliotecas):

- **No celular, caso seja Android**, existem aplicativos personalizados disponíveis no Google Play, como por exemplo o **Pydroid 3**. O aplicativo já vem com a opção "Pip", que ajuda a instalar as bibliotecas de terceiros, como é o caso do *matplotlib* (uso para construir gráficos).
- **Em um computador Windows**, é possível executar o Python 3 direto no terminal PowerShell ou com auxílio de programas como o **Visual Studio Code** (recomendado 8GB+ de RAM). No Visual Studio Code vá em Extensões e instale Python e o Pip Manager.
- **Em um computador Linux**, é fácil de usar o Python 3, pois nas versões mais recentes do Ubuntu (e similares tipo o Linux Mint) o compilador de Python já vem pré-instalado no sistema operacional, e podemos checar sua versão com o comando:

```
$ python --version
```

Caso não tenha o Python 3 no seu Linux, é simples de instalar com os comandos seguintes:

```
$ sudo apt update
$ sudo apt install build-essential
$ sudo apt install python3.11
```

Por fim, para instalar o pip (instalador de bibliotecas):

```
$ sudo apt-get install python3-pip
```

3º momento (45 min): De inicio a avaliação pedagógica. Explique aos alunos como usar o Google Forms, a partir do link disponibilizado no Classroom.

4º momento (15 min): De inicio a uma breve introdução de como usar o Python, explique o que são comentários e operações básicas.

Deve-se discutir com os alunos alguns como o Python aceita os comandos, seu formato e algumas funções básicas. No Python e muitas outras linguagens similares, os comandos são lidos por linha, de cima para baixo. Podemos escrever

comentários no código sem atrapalhar os comandos, para isso utilizamos o símbolo #. Tudo que é escrito após o # é ignorado. Colocamos apenas para pessoas lerem, explicando o objetivo do comando ou deixando algum recado tipo (esse código não faz tal coisa, ex: conta uvas, mas não consegue contar melões);

Para imprimir qualquer resultado obtido pela execução do código, na tela do seu computador ou celular, usamos o comando **print()**. Nesta aula será disponibilizado o código do professor aos alunos (em anexo).

Deve-se destacar aos alunos que podemos de fato escrever comentários em português no código, mas tudo fora dos comentários deve seguir o molde da língua Inglesa. Assim, não podemos usar acentuações, o "ç" e quanto aos números devemos usar ponto ao invés da vírgula para as casas decimais (Errado: 8,5 | Correto: 8.5).

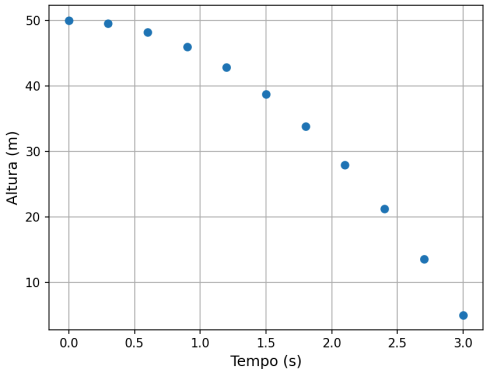
Será apresentado aos alunos como tratar três tipos de variáveis: números, textos e listas. Como em qualquer calculadora, o Python aceita algumas operações numéricas básicas (soma, subtração, multiplicação, divisão e potência). Após falar das operações numéricas é recomendado tratar textos e frases e como podemos manipula-los no Python. Por fim, será discutido o agrupamento de dados numéricos com textos por meio das listas. Com o tratamento correto desses dados, podemos resolver qualquer problema de Física e matemática. A lógica de programação será vista nas aulas seguintes.

5º momento (30 min): Com alguns comandos básicos explicados e descritos no quadro, deixe os alunos testarem em seus aparelhos. Circule a sala tirando dúvidas.

AVALIAÇÃO

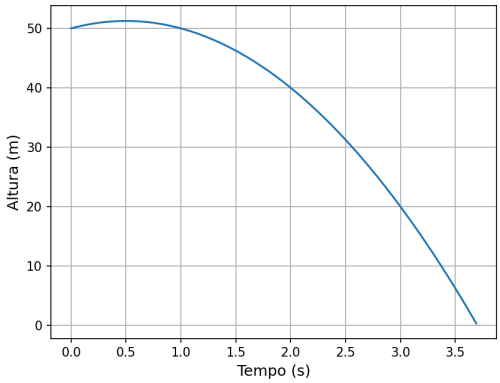
Avaliação diagnóstica. Recomenda-se fazer a avaliação pelo Google Forms, mas pode imprimir-la e entregar aos alunos. As questões são descritas a seguir:

1. Observando o gráfico, podemos dizer que:



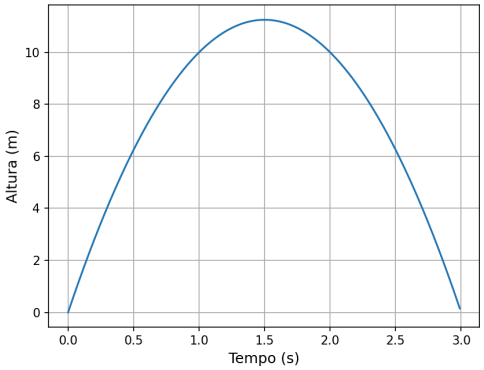
   a. A posição inicial do corpo é o chão
   b. O corpo possui aceleração diferente de zero **(correto)**
   c. Após 1,5 segundos o corpo deslocou 40 m da posição inicial

2. Observando o gráfico, podemos dizer que:



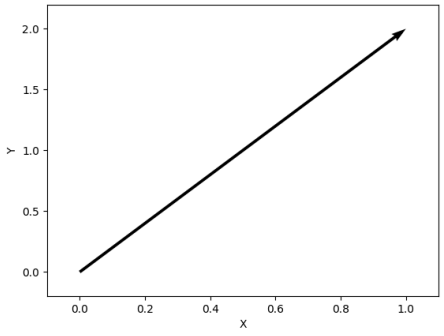
a. A velocidade inicial tem a mesma direção e sentido que a gravidade
b. A velocidade inicial é zero (partiu do repouso)
c. A velocidade inicial tinha sentido oposto à gravidade **(correto)**

3. Observando o gráfico do lançamento de um corpo, podemos dizer que:



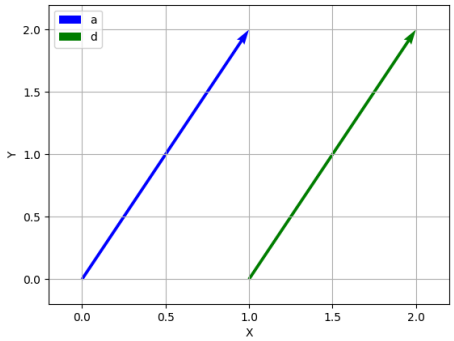
a. Após 3 segundos o corpo se deslocou 3 m para a direita
b. No tempo 1,5 s o corpo parou no ar **(correto)**
c. O deslocamento do corpo foi de 22 m

4. Qual tamanho o vetor:



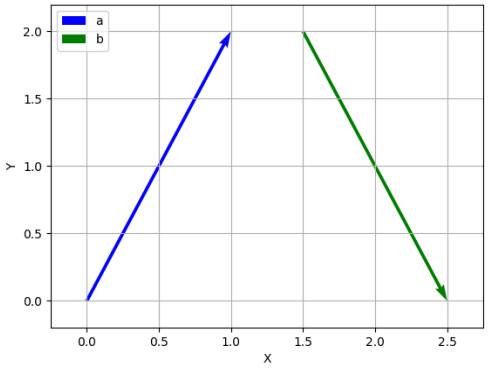
a. 1
b. raiz quadrada de 2
c. raiz quadrada de 5 **(correto)**

5. Os vetores a e d são:

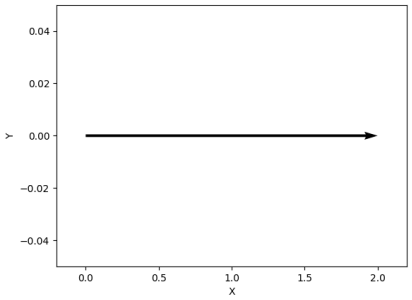


a. vetores irmãos
b. vetores diferentes
c. vetores iguais **(correto)**

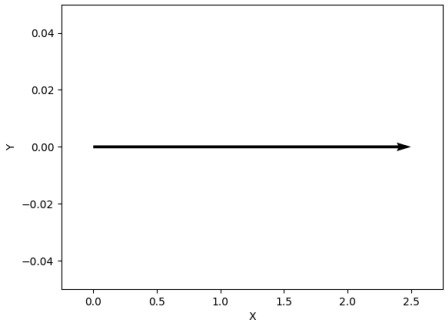
6. Qual será a soma dos vetores abaixo?



a. vetor nulo
b. o vetor de tamanho 2 abaixo **(correto)**



c. o vetor de tamanho 2,5 abaixo



7. Resolva a equação: x + 2y = 25
    a. x = 25 , y = 2
    b. se y=1, então x= 20
    c. se y=1, então x= 23 **(correto)**
8. Resolva a equação: x^2 -4 = 0
    a. x= 4, x=2
    b. x=-2, x=2 **(correto)**
    c. x=-8, x=8
9. Resolva o sistema de equações:
    x^2 - 2y = 0
    6x + y =-18
    a. x=-6, y=18 **(correto)**
    b. x=6, y=-18
    c. x=-18, y=6
10. Uma ave marinha costuma mergulhar de uma altura de 20 m para buscar alimento no mar. Suponha que um desses mergulhos tenha sido feito em sentido vertical, a partir do repouso. Desprezando-se as forças de atrito, a ave chegará à superfície do mar em qual velocidade em km/h? (Assuma g=9,8 m/s^2)
    a. 71,3 **(correto)**
    b. 76,4
    c. 64,4

REFERÊNCIAS

G. Van Rossum, F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace; 2009.

ANEXOS

Será disponibilizado ao alunos via Google Classroom o código a seguir:
(Para reduzir o tamanho deste arquivo, coloquei todos os códigos feitos no github e deixo disponível aqui o link de acesso)
https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/Aula_1.py

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO

Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME

Professor: Bruno Camargo Braga

Componente Curricular: Física

Unidade temática: Investigação Científica e Processos Criativos.

Tema de estudo: Fundamentos de programação Python

Grupos 1 e 2: 1º ano do Ensino Médio	**Tempo:** 2 horas	**Data:** 29/04/2025

COMPETÊNCIAS E HABILIDADES

(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos;
(EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.

OBJETIVOS DE APRENDIZAGEM

Utilizando a linguagem de programação Python (versão 3.11), ser capaz de:

1. Entender e utilizar estruturas de repetição (código em ciclos);
2. Interagir com o programa durante sua execução (uso do comando *input*);
3. Entender e utilizar a estrutura condicional *if-else*;

RECURSOS/MATERIAIS DIDÁTICOS

Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.

PROCEDIMENTOS METODOLÓGICOS

1º momento (10 min): Saudações iniciais, chamada e anotação da ata de sala

2º momento (20 min): Revise os comandos vistos no encontro anterior.

3º momento (10 min): Apresente o problema da repetição de comandos e a fácil solução, por meio dos blocos cíclicos.

Uma das vantagens das linguagens de alto nível, como o Python, está nas funcionalidades cujo objetivo é evitar repetição de comandos, o que deixa o código mais limpo e fácil de entender. A funcionalidade mais comum entre as linguagens

de programação é a estrutura de repetição (ou muitas vezes chamadas por ciclos/*loops)*. Os comandos que iniciam ciclos no Python são: *while* e *for*.

No começo da aula devemos mostrar aos alunos o modo mais trabalhoso de fazer, a exemplo preencher uma lista com dados. Para tal repetimos o comando de adicionar a lista toda vez que precisamos. Porém, com o uso correto de uma estrutura de repetição, obtemos a automação desse processo.

4º momento (20 min): Detalhe os comandos cíclicos "while" e "for".

A seguir é recomendado que seja explicado tal estrutura, seguindo os comandos:

- Comando **while expressão:** Do inglês "enquanto" a expressão for verdade, repita o código dentro da estrutura. Este comando para ser implementado corretamente precisa de um passo que modifica a expressão toda vez que completa-se um ciclo. Isso é feito para que eventualmente a expressão se torne falsa, e assim o ciclo pare. Ex: Com i=0, se a expressão for "while i<5", dentro da estrutura deve ser escrito algum passo que aumente 'i', de modo que ao 'i' se tornar 5 ou maior, a estrutura "while" termina.
- Comando **for x in lista**: Diferentemente do "while", o "for" deixa a quantidade de repetições bem definida na expressão escrita após o "for" – "x in lista". Do inglês significa: repita a estrutura "para" x assumindo os valores da lista um a um. Ex: "for x in [1,25,30]": repita a estrutura assumindo que o valor de x será 1 na primeira execução, 25 na segunda e 30 na última.

O comando "for" é mais utilizado quando a lista de valores de uma variável é bem conhecida, enquanto que o "while" é mais genérico e muitas vezes usado para ciclos com muitas repetições e condições muito raras de parada.

5º momento (30 min): Apresente a possibilidade de interação com o programa, durante sua execução. Termine o assunto falando da necessidade de verificar as respostas antes de usa-las, para isso apresente a estrutura condicional "if-else".

Para evitar que façamos tudo direto no código em si, existem funções de interação durante a execução do código. A mais comum é o pedido de uma resposta a perguntas pré-definidas. Fazemos isso no Python com o comando **input().** Ao executar o código ele irá pausar pedindo que seja repassado uma resposta. Ex: dentro de um ciclo pode ter uma linha "x=input('Forneça um número')"; significa que toda vez que o ciclo reiniciar o código pausa, imprime a mensagem ('Forneça um número') na tela e espera a resposta. Assim podemos modificar o significado de uma variável durante a execução, podendo por exemplo executar 2 ou 20 repetições de um ciclo usando o mesmo código.

Com comandos tipo o **input(),** teremos uma ou mais variáveis no código que não sabemos o valor e seu tipo (se é um número ou texto). Na mensagem se *input* podemos pedir um tipo de resposta específico, mas isso não garante que o código receberá o que se espera. Para garantir que estamos trabalhando corretamente, precisamos verificar as respostas dadas. Para tal usamos a estrutura condicional "if-else". Do inglês "se-senão", são blocos que permitem execuções diferentes baseado na verificação feita.
Ex: se x for par faça X1, senão (ele será ímpar) então faça X2; essa verificação é

escrita como:

"if x%2==0:

  X1

Else:

  X2"

6º momento (30 min): Apresente as atividades, deixe os alunos fazendo e circule pela sala tirando dúvidas.

Critérios de avaliação:

1. *Atingiu o objetivo da atividade: solucionou o problema usando o Python 3 como principal ferramenta – independente das estruturas e métodos usados, desde que o programa escrito seja funcional* ***(7,0pt)****;*
2. *Estrutura e métodos:*
    a. *O programa está organizado e fácil de entender* ***(1,0pt)****;*
    b. *Foi utilizado os métodos explanados em cada encontro* ***(1,0pt)****;*
    c. *O programa é bem comentado, explicando o raciocínio lógico* ***(1,0pt)****.*

AVALIAÇÃO

Atividades recomendadas:

1. Faça um código que peça uma sequência de valores e calcule a média deles.
2. Faça uma código que peça um número inteiro e nos diz se o número é par ou ímpar.

REFERÊNCIAS

G. Van Rossum, F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace; 2009.

ANEXOS

Será disponibilizado ao alunos via Google Classroom o código a seguir:

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/Aula_2.py

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO

Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME

Professor: Bruno Camargo Braga

Componente Curricular: Física

Unidade temática: Investigação Científica e Processos Criativos.

Tema de estudo: Fundamentos de programação Python – Parte 2

Grupos 1 e 2: 1º ano do Ensino Médio	**Tempo:** 2 horas	**Data:** 06/05/2025

COMPETÊNCIAS E HABILIDADES

(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos;
(EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.

OBJETIVOS DE APRENDIZAGEM

Utilizando a linguagem de programação Python (versão 3.11), ser capaz de:

1. Entender e utilizar estruturas de repetição indefinidas (ciclos infinitos);
2. Definir seus próprios métodos dentro da linguagem (comando *def*);
3. Trabalhar com funções matemáticas (comando *lambda*);

RECURSOS/MATERIAIS DIDÁTICOS

Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.

PROCEDIMENTOS METODOLÓGICOS

1º momento (10 min): Saudações iniciais, chamada e anotação da ata de sala

2º momento (20 min): Revise os comandos vistos no encontro anterior.

3º momento (20 min): Apresente os ciclos "infinitos" e suas vantagens

Muitas vezes é impossível de saber quando uma estrutura cíclica deve acabar. Nesses casos usamos o comando "while True", que significa: enquanto a verdade seja verdadeira repita a estrutura. Isso coloca o programa em um ciclo sem fim.

Para que a execução termine é necessário usar a estrutura condicional "if-else" dentro da estrutura do "while", quando obtermos o resultado esperado precisamos uma maneira de cessar a execução. No Python podemos usar o comando **break** dentro do "if-else", que literalmente quebra a execução do ciclo.

Comece a aula com um exemplo de ciclo indefinido. Um caso simples seria realizar alguma operação qualquer com números que ao terminar imprime o resultado e pede um novo número para reiniciar o processo. Esse ciclo usaria um tipo numérico específico para cessar o ciclo, por exemplo um número primo.

4º momento (20 min): Apresente a definição de métodos e sua vantagem em evitar repetições.

É extremamente comum que, durante a implementação da solução de um problema, exista uma sequência de comandos que é repetida diversas vezes no código. Para evitar o trabalho extra de sempre reescrever a mesma sequência de linhas, podemos definir métodos no Python usando o comando "def" e simplesmente chama-los toda vez que for necessário. Isso não apenas diminui o trabalho de escrita, mas também deixa a leitura do código mais fácil. Isso é feito seguindo o modelo:

"def metodo(variavel):

...sequência de comandos”

Quando necessário é então chamado no código seguindo o exemplo:

"...comandos..

metodo(x)

...comandos..

metodo(y)

...comandos.."

Um exemplo de como apresentar a definição de métodos seria pedir para os alunos calcular o fatorial de vários números, ou várias posições da sequência de Fibonacci. Assim fica evidente a vantagem do comando “def”.

5º momento (20 min): **No fim da aula apresente a ideia de utilizar funções matemáticas. Isso ajuda muito na construção de gráficos e resoluções de problemas comumente presentes em vestibulares.**

Uma das formas de calcular funções matemáticas no Python é usar o comando "lambda". Diferentemente dos métodos vistos até o momento, "lambda" não retorna um valor específico, mas sim uma função/regra.

Ex: o comando "f = lambda x: x**2", faz f receber a função quadrática. Ao chamar os valores f(2), f(3) e f(4) será recebido os valores 4, 9 e 16.

Com o a união dos comandos “def” e “lambda” conseguimos trabalhar com funções mais complexas, como a equação horária da posição no movimento uniformemente variado.

6º momento (30 min): Apresente as atividades, deixe os alunos fazendo e circule pela sala tirando dúvidas.

Critérios de avaliação:

1. *Atingiu o objetivo da atividade: solucionou o problema usando o Python 3 como principal ferramenta – independente das estruturas e métodos usados, desde que o programa escrito seja funcional **(7,0pt)**;*
2. *Estrutura e métodos:*
    a. *O programa está organizado e fácil de entender **(1,0pt)**;*
    b. *Foi utilizado os métodos explanados em cada encontro **(1,0pt)**;*
    c. *O programa é bem comentado, explicando o raciocínio lógico **(1,0pt)**.*

AVALIAÇÃO

Atividades recomendadas:

1. Uma ave marinha costuma mergulhar de uma altura de 20 m para buscar alimento no mar. Suponha que um desses mergulhos tenha sido feito em sentido vertical, a partir do repouso. Desprezando-se as forças de atrito, a ave chegará à superfície do mar em qual velocidade em km/h?

   Entregue um código que faça a solução do problema e imprime a resposta.
   Dica: use a definição de métodos "def".

2. Um trem de 150 m de comprimento se desloca com velocidade escalar constante de 16 m/s. Esse trem atravessa 3 túneis em dias diferentes e leva desde a entrada até a saída completa de cada túnel 50 s, 20 s e 90 s. Quanto é o comprimento de cada túnel?

   Entregue um código que faça a solução do problema e imprime a resposta.
   Dica: use a função "lambda".

REFERÊNCIAS

G. Van Rossum, F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace; 2009.

ANEXOS

Será disponibilizado ao alunos via Google Classroom o código a seguir:

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/Aula_3_base.py

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/1Ano/Aula_3.py

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO

Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME

Professor: Bruno Camargo Braga

Componente Curricular: Física

Unidade temática: Investigação Científica e Processos Criativos.

Tema de estudo: Construindo gráficos no Python

Grupos 1 e 2: 1º ano do Ensino Médio	**Tempo:** 2 horas	**Data:** 12/05/2025

COMPETÊNCIAS E HABILIDADES

(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos;
(EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.

OBJETIVOS DE APRENDIZAGEM

Utilizando a linguagem de programação Python (versão 3.11), ser capaz de:

1. Utilizar bibliotecas não padrões do Python;
2. Construir gráficos a partir de uma lista de pontos;

RECURSOS/MATERIAIS DIDÁTICOS

Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.

PROCEDIMENTOS METODOLÓGICOS

1º momento (10 min): Saudações iniciais, chamada e anotação da ata de sala

2º momento (20 min): Revise os comandos vistos no encontro anterior.

3º momento (20 min): Apresente e ajude os alunos a usar a biblioteca necessária para geração de gráficos.

Para gerar gráficos precisamos de uma biblioteca específica. Caso o aluno esteja usando aplicativos como o *Google Colab Notebook*, todas as bibliotecas já estão instaladas e ele não precisa de nenhum trabalho extra. Se o aluno estiver programando localmente (sem internet), ele precisará instalar o **matplotlib** e o **numpy**. O *numpy* é necessário para algumas funções do *matplotlib.*

Recomenda-se instalar o **Pip**, pois esse programa faz o trabalho de instalar corretamente novas bibliotecas no seu interpretador local do Python.
O guia de instalação do Pip está no material "Como utilizar o Python 3" da Aula 1.
Com o Pip instale as bibliotecas: **matplotlib**; **numpy**.

4º momento (20 min): **Comece a aula guiando os alunos a como usar o "matplotlib" para construir gráficos.**

A forma mais simples de construção de gráfico pelo matplotlib é utilizando uma lista de pontos a serem imprimidos no gráfico. Primeiramente precisamos que a biblioteca monte a interface do gráfico. Então chamamos a biblioteca com:

"import numpy as np

import matplotlib.pyplot as plt"

Assim deixamos mais fácil chama-la com a abreviação plt, e usamos então o comando:

"fig = plt.figure(dpi=150)", que cria a interface com resolução de 150 linhas.

Com a interface pronta, podemos definir a lista de pontos (com os métodos vistos nas aulas anteriores) e nomear os eixos do gráfico usando:

"plt.ylabel("nome do eixo y", fontsize=12)

plt.xlabel("nome do eixo x", fontsize=12)"

Se necessário podemos colocar uma grade no gráfico, para facilitar a visualização dos valores de cada ponto. Para isso usamos o comando "plt.grid(True)".

Para adicionar os pontos no gráfico, precisamos colocar separadamente os valores de x e de y de cada ponto. Ex: Para adicionar os pontos (1,2), (-5,3) e (2,0); use o comando:

"plt.plot([1,-5,2], [2,3,0], 'o')"

O símbolo 'o' no final indica que é para o gráfico apresentar os pontos, se não utilizarmos ele o gráfico é contruído usando uma linha. Essa linha simplesmente liga os pontos e os omite, então com poucos pontos fica uma representação incorreta. O gráfico apresentaria corretamente uma curva se tivesse muitos pontos (mais de 100).

Para o interpretador do Python entender que deve-se mostrar o gráfico na tela, usamos o comando: "plt.show()".

5º momento (20 min): **De um exemplo de gráfico na aula. Use um caso representativo do MCU (conteúdo do 2º Bim de Física).**

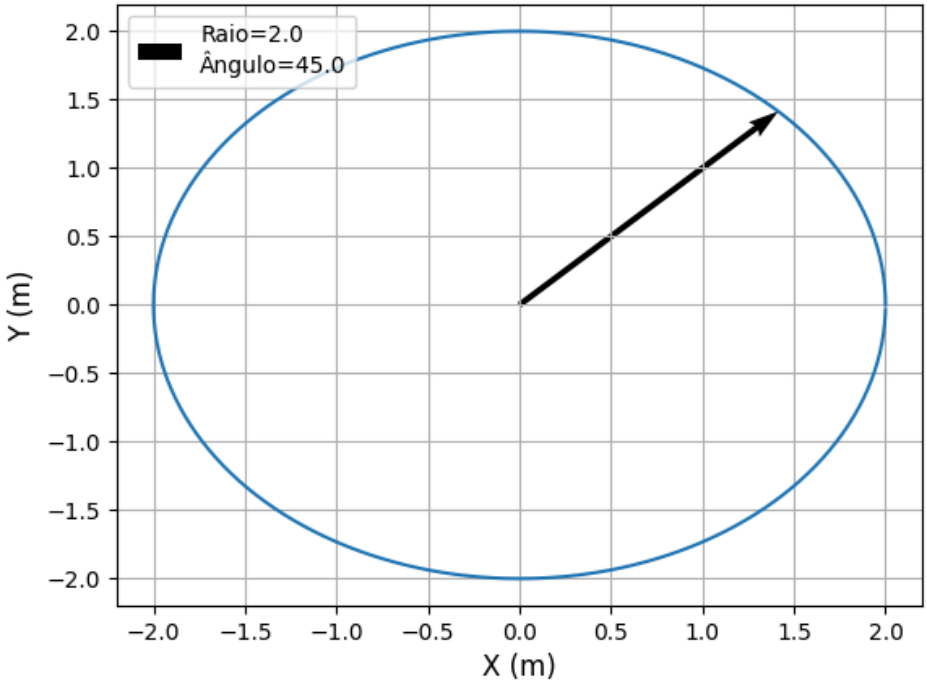
Lembre-se que para desenhar um círculo, precisamos das relações de x e y com o raio. Para construir as listas de pontos usamos:

"x= lambda theta: R* cos(theta)

y= lambda theta: R* sen(theta)"

e a conversão de graus para radiano ($rad = graus \cdot \frac{\pi}{180}$). Para completar uma

volta fazemos o *theta* variar de 0° a 360°, desenhamos também um vetor raio a 45°, e assim, obtemos o gráfico abaixo.



6º momento (30 min): Apresente as atividades, deixe os alunos fazendo e circule pela sala tirando dúvidas.

Critérios de avaliação:

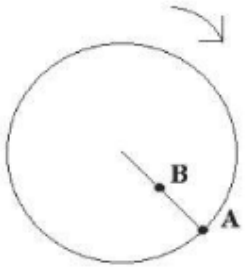
1. *Atingiu o objetivo da atividade: solucionou o problema usando o Python 3 como principal ferramenta – independente das estruturas e métodos usados, desde que o programa escrito seja funcional* ***(7,0pt)****;*
2. *Estrutura e métodos:*
    a. *O programa está organizado e fácil de entender* ***(1,0pt)****;*
    b. *Foi utilizado os métodos explanados em cada encontro* ***(1,0pt)****;*
    c. *O programa é bem comentado, explicando o raciocínio lógico* ***(1,0pt)****.*

AVALIAÇÃO
Atividades recomendadas: • *Um trabalhador mora a 2,4 km de distância do seu emprego. Ele tem que decidir entre duas opções de transporte para chegar ao trabalho: de ônibus, cuja velocidade média em sua região é de 18 km/h, ou de bicicleta, com a qual ele é capaz de desenvolver uma velocidade média de 8 m/s. Considerando que existe um ponto de ônibus bem em frente à sua casa e outro ponto em frente ao seu trabalho, e desconsiderando eventuais perdas de tempo na espera do ônibus, qual das opções de meio de transporte é*

mais rápido? Indique com gráfico.

<u>Entregue um código que faça a solução do problema e gere o gráfico. Adicione o seu código e a imagem do gráfico.</u>
Dica: use a função "lambda", e lembre do comando "import matplotlib.pyplot as plt" no começo do código.

- Considere uma polia girando em torno de seu eixo central, conforme figura abaixo. A velocidade dos pontos A e B são, respectivamente, 60 cm/s e 0,3 m/s. A distância AB vale 10 cm. O diâmetro e a velocidade angular da polia, respectivamente, valem:



REFERÊNCIAS

1. G. Van Rossum, F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace; 2009.
2. J. D. Hunter. *Matplotlib: A 2D Graphics Environment.*Computing in Science & Engineering. v. 9, n. 3, p. 90-95, 2007.

ANEXOS

Será disponibilizado ao alunos via Google Classroom o código a seguir:

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/1Ano/Aula_4.py

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO

Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME

Professor: Bruno Camargo Braga

Componente Curricular: Física

Unidade temática: Investigação Científica e Processos Criativos.

Tema de estudo: Trabalhando com vetores no Python

Grupos 1 e 2: 1º ano do Ensino Médio	**Tempo:** 2 horas	**Data:** 13/05/2025

COMPETÊNCIAS E HABILIDADES

(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos;
(EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.

OBJETIVOS DE APRENDIZAGEM

Utilizando a linguagem de programação Python (versão 3.11), ser capaz de:

1. Utilizar vetores no plano XY;
2. Entender a ideia base de classes;
3. Construir a representação de seta para vetores em gráficos.

RECURSOS/MATERIAIS DIDÁTICOS

Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.

PROCEDIMENTOS METODOLÓGICOS

1º momento (10 min): Saudações iniciais, chamada e anotação da ata de sala

2º momento (10 min): Revise os comandos vistos no encontro anterior. Lembre os alunos de chamar a biblioteca no começo do programa.

Como feito na aula 4, precisamos das bibliotecas de gráfico, então devemos lembrar de escrever:
"import numpy as np
import matplotlib.pyplot as plt"

3º momento (10 min): Lembre os alunos da definição de vetores.

4º momento (20 min): **Para facilitar a visualização focamos em vetores no plano XY (caso perguntem, existe um método de apresentar vetores em 3D). Apresente a ideia de classes no Python (não precisa de classes para tratar de vetores, porém fica mais organizado e fácil de relacionar pois o vetor é uma ferramenta especial da matemática).**

Como o vetor é uma ferramenta matemática única, definimos um método específico de construção dos vetores, em que colocamos suas propriedades junto a sua construção. No Python fazemos isso usando o método de "classes". De maneira resumida, uma classe é um conjunto de métodos que possuem variáveis comuns. As variáveis comuns do nosso vetor serão as coordenadas da origem e da ponta do vetor, além de seu nome.

Comece a aula falando brevemente de classes. É possível trabalhar com vetores sem usar classes, porém os comandos ficam mais legíveis com o método de classe. Lembre os alunos que por ser um método mais complexo, eles poderão sempre usar o mesmo código da aula como base para resolver as atividades.

Uma das vantagens de definir o vetor como uma classe é a facilidade em escrever algumas operações como a soma vetorial. Ex: se escrevemos a classe com nome Vetor, é possível fazer:

"a= Vetor(xa,ya)

b= Vetor(xb,yb)

c = a+b"

Assim ‘c’ já será montado como pertencente a classe Vetor. Fale um pouco sobre classes. Algumas de suas vantagens:

1. Podemos definir métodos dentro da classe que só existem para ela. O método de somar vetores, por exemplo, não será usado para somar grandezas escalares.
2. Podemos definir uma estrutura com vários parâmetros – vetores possuem tamanho, direção e sentido.
3. Podemos utilizar do método de Herança para aproveitar código – por exemplo, uma classe de vetor que o aluno faz para resolver uma atividade pode herdar todas a definições feitas pela classe de vetor feita em aula (sem precisar copiar as linhas de código, basta possuir o arquivo do vetor e usar herança).

5º momento (20 min): Detalhe como definimos uma classe e adicionamos métodos a ela.

Com o uso de classes, precisamos definir o método construtor "**__init__**", que nesse caso simplesmente receberá as coordenadas dos pontos que definem o vetor e um nome. O método classe basicamente é um método com vários submétodos, então precisamos definir um método padrão que sempre será usado ao chamar a classe. Esse método padrão é o "**__init__**". Todos os outros métodos da classe representarão as propriedades dos vetores ou operações opcionais. Lembrando que ao usar o método de classes, precisamos seguir o seguinte molde:

```
"class Vetor:
   def __init__(self, coordenadas, nome):
      self.origem =...
      self.ponta= …
      self.nome= ...
   def __add__(self,vetor):
      #permite fazer a soma
       .....
   def tamanho(self):
     ....."
```

6º momento (20 min): Termine apresentando um método para mostrar o vetor graficamente. Será muito semelhante aos métodos da aula 4, com uma pequena alteração.

Com o método do vetor bem escrito, temos que definir o método para apresenta-lo em gráfico. O método recomendado é semelhante ao que foi feito na aula 4 (sobre construção de gráficos). Porém, ao invés de colocar uma lista de pontos com o comando "plt.plot()", usamos os comandos:

```
"origem = [xo], [yo]
ponta = [xp], [yp]
plt.quiver(*origem, *ponta, angles='xy', scale_units='xy', scale=1)"
```

Os dados xo, yo, xp e yp são retirados do vetor que queremos visualizar.

7º momento (30 min): Apresente as atividades, deixe os alunos fazendo e circule pela sala tirando dúvidas.

Critérios de avaliação:

1. *Atingiu o objetivo da atividade: solucionou o problema usando o Python 3 como principal ferramenta – independente das estruturas e métodos usados, desde que o programa escrito seja funcional* ***(7,0pt)****;*
2. *Estrutura e métodos:*
    a. *O programa está organizado e fácil de entender* ***(1,0pt)****;*
    b. *Foi utilizado os métodos explanados em cada encontro* ***(1,0pt)****;*
    c. *O programa é bem comentado, explicando o raciocínio lógico* ***(1,0pt)****.*

AVALIAÇÃO

Atividades recomendadas:

1. Reutilizando o código visto na aula, implemente o produto de um vetor por um escalar. Ex: vetor no inicio: -> ; após produto por 3: --->

   Mostre a imagem do gráfico construído.

2. Em uma brincadeira de caça ao tesouro, o mapa diz que para chegar ao local onde a arca de ouro está enterrada, deve-se, primeiramente, dar 10 passos na direção norte, depois 12 passos para o leste, em seguida, 7

passos para o sul, e finalmente 8 passos para oeste. A partir dessas informações:

a. Construa um gráfico no Python com a trajetória descrita no mapa usando vetores
b. Se um caçador caminhasse em linha reta, desde o ponto de partida até o ponto de chegada, quantos passos ele daria? Mostre o vetor soma no gráfico.

REFERÊNCIAS

1. G. Van Rossum, F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace; 2009.
2. J. D. Hunter. *Matplotlib: A 2D Graphics Environment.*Computing in Science & Engineering. v. 9, n. 3, p. 90-95, 2007.

ANEXOS

Será disponibilizado ao alunos via Google Classroom o código a seguir:

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/1Ano/Aula_5.py

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO

Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME

Professor: Bruno Camargo Braga

Componente Curricular: Física

Unidade temática: Investigação Científica e Processos Criativos.

Tema de estudo: Resolvendo equações e/ou sistemas de equações no Python

Grupos 1 e 2: 1º ano do Ensino Médio	**Tempo:** 2 horas	**Data:** 20/05/2025

COMPETÊNCIAS E HABILIDADES

(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos;
(EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.
(EMIFCNT08) Selecionar e mobilizar intencionalmente conhecimentos e recursos das Ciências da Natureza para propor ações individuais e/ou coletivas de mediação e intervenção sobre problemas socioculturais e problemas ambientais.

OBJETIVOS DE APRENDIZAGEM

Utilizando a linguagem de programação Python (versão 3.11), ser capaz de:

1. Definir variáveis simbólicas e usa-las para montar equações;
2. Expandir e fatorar equações;
3. Resolver equações e sistemas de equações.

RECURSOS/MATERIAIS DIDÁTICOS

Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.

PROCEDIMENTOS METODOLÓGICOS

1º momento (10 min): Saudações iniciais, chamada e anotação da ata de sala

2º momento (20 min): Revise os comandos vistos no encontro anterior.

3º momento (10 min): Apresente uma breve introdução da biblioteca sympy.

Semelhante a aula 4, quando precisamos da biblioteca matplotlib de gráficos, aqui precisamos de uma nova biblioteca. Se o aluno estiver programando localmente

(sem internet), ele precisará instalar a biblioteca **sympy.** Os alunos devem lembrar de adicionar no começo do código o comando:
"import sympy as sp"

Essa biblioteca "sympy" foi criada para resolver cálculos matemáticos exatos, ou seja, sem aproximações. Exemplo: a raiz quadrada de 8

- Sem o "sympy" podemos fazer 8**0.5; isso gera o resultado aproximado **2.8284271247461903**.
- Com o "sympy" fazemos sp.sqrt(8) e isso gera **2*sqrt(2)**, que é a fatoração da raiz de 8.

4º momento (10 min): Descreva o método simbólico do sympy.

Até a aula passada, para descrever uma função como "f = x+2*y" precisaríamos ou associar valores a x e y diretamente ou definir um método f que faz "x+2*y". Com o 'sympy' podemos definir x e y como símbolos e criar f, usando:

"x, y = sp.symbols('x y')
f = x + 2*y"

Assim podemos realizar operações como f-x e obter resultados exatos (f - x = 2*y). Com o "sympy" podemos pedir que ele simplifique expressões com o comando **expand** por exemplo: "sp.expand((x-5)**2)" gera "x**2 -10*x + 25".

Podemos também fazer o contrário e pedir para o “sympy” fatorar as expressões. Por exemplo: se quisermos as raízes da equação "x**2 +2*x -15 =0", podemos fazer "factor(x**2 +2*x -15)" que gera "(x-3)*(x + 5)". É importante notar que nem sempre o *factor* gera algum resultado, pois o programa pode entender que a forma atual da expressão já está num formato simplês o suficiente. Devido a isso o *factor* resolve apenas os casos mais simples. Para soluções de qualquer equação usamos outro comando, o **solve**.

5º momento (10 min): Apresente o solve, linsolve e nonlinsolve. Comandos para resolver esquações e sistemas de equações.

Além de equações, o comando “solve” pode receber uma lista de equações vinculadas em um único sistema e procurar possíveis soluções.

Exemplos de uso do **solve**:

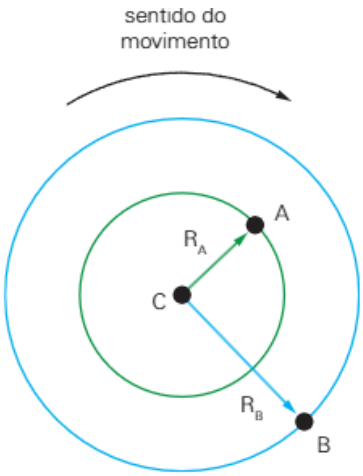
- Para $x^2 - 2 = 0$, escrevemos "sp.solve(x**2-2, x)" que gera "[-sqrt(2), sqrt(2)]";
- Para $x^2 - 16 = 9$, escrevemos "sp.solve(sp.Eq(x**2-16,9), x)" que gera "[-5, 5];
- Para o sistema $\{x - 2y = 0,\ x + y = 18\}$, escrevemos “sp.linsolve([x-2*y, sp.Eq(x+y,18)], (x,y))”.
- Para o sistema $\{x^2 - 2y = 0,\ 6x + y = -18\}$, escrevemos “sp.nonlinsolve([x**2-2*y, sp.Eq(6*x+y,-18)], (x,y))”.

Obs: se as variáveis do sistema de equações não tiverem expoentes (elevado a 1) usamos o comando **linsolve()**, para sistemas onde qualquer variável tenha expoente (2 ou maior) usamos o comando **nonlinsolve()**. Então, na cinemática no regime de Movimento Uniformemente Variado (MUV), para as equações de Torricelli

ou horária da posicção se forem usadas em conjunto com alguma outra para solucionar problemas, devemos usar o **nonlinsolve()**.

6º momento (20 min): Revise com os alunos os conceitos de cinemática dos movimentos curvilíneos.

Movimento circular com acoplamento no mesmo eixo



Nesse caso, os períodos T (tempo para uma volta completa) de A e B são iguais pois o movimento angular será igual,

$$T_A = T_B$$

A frequência $f = \frac{1}{T}$ *e a velocidade angular* $\omega = 2\pi f$*, também são iguais*

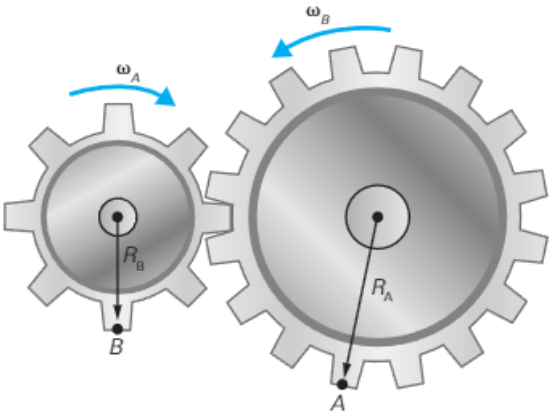
$$f_A = f_B;\ \ \omega_A = \omega_B$$

A relação da velocidade linear v *e do raio* R*, é dada por:*

$$\omega = \frac{v}{R}\text{, logo}$$

$$\frac{v_A}{R_A} = \frac{v_B}{R_B}$$

Movimento circular com acoplamento e em eixos separados



Nesse caso as grandezas angulares serão distintas, porém a velocidade linear será a mesma

$$v_A = v_B\text{, portanto}$$

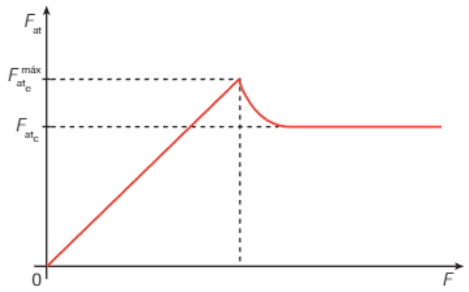
$$\omega_A R_A = \omega_B R_B$$

$$f_A R_A = f_B R_B$$

$$\frac{R_A}{T_A} = \frac{R_B}{T_B}$$

7º momento (10 min): Revise com os alunos sobre forças de atrito devido a interação de contato com uma superfície.

O módulo da força de atrito F_{at} *é proporcional a normal à superfície de contato N* $F_{at} = \mu N$*, onde o coeficiente de atrito* μ *depende se a situação permanece estática ou existe movimento. Enquanto permanece estático, a força de atrito se igual a força aplicada (e assim ela pode aumentar ou diminuir). Porém a força de atrito tem um limite superior, que quando quebrado fará a força de atrito diminuir levemente e permanecer constante durante o movimento do corpo – como esquematizado no gráfico.*



8º momento (30 min): Apresente as atividades, deixe os alunos fazendo e circule pela sala tirando dúvidas.

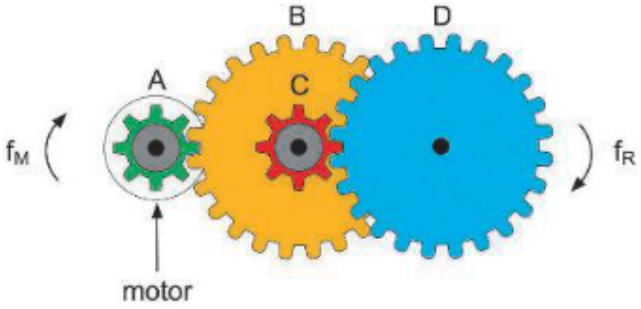
Critérios de avaliação:

1. *Atingiu o objetivo da atividade: solucionou o problema usando o Python 3 como principal ferramenta – independente das estruturas e métodos usados, desde que o programa escrito seja funcional* ***(7,0pt)****;*
2. *Estrutura e métodos:*
    a. *O programa está organizado e fácil de entender* ***(1,0pt)****;*
    b. *Foi utilizado os métodos explanados em cada encontro* ***(1,0pt)****;*
    c. *O programa é bem comentado, explicando o raciocínio lógico* ***(1,0pt)****.*

AVALIAÇÃO

Atividades recomendadas:

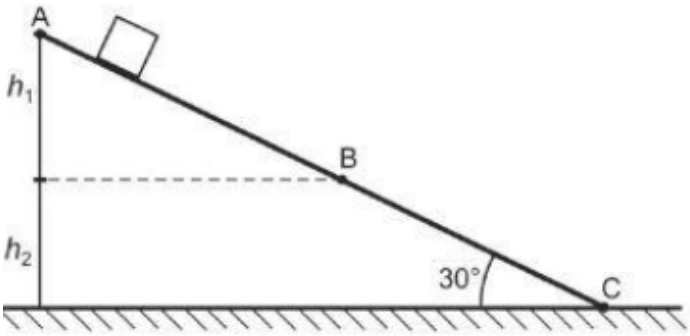
- Um pequeno motor a pilha é utilizado para movimentar um carrinho de brinquedo. Um sistema de engrenagens transforma a velocidade de rotação desse motor na velocidade de rotação adequada às rodas do carrinho. Esse sistema é formado por quatro engrenagens, A, B, C e D, sendo que A está presa ao eixo do motor, B e C estão presas a um segundo eixo e D a um terceiro eixo, no qual também estão presas duas das quatro rodas do carrinho.



  Nessas condições, quando o motor girar com frequência fM, as duas rodas do carrinho girarão com frequência fR. Sabendo que as engrenagens A e C possuem 8 dentes, que as engrenagens B e D possuem 24 dentes, que não há escorregamento entre elas e que fM=13.5Hz é correto afirmar que fR, em Hz, é igual a? Resolva o problema utilizando os comandos do **sympy** vistos na aula 6, e entregue o código usado.

- Um bloco escorrega, livre de resistência do ar, sobre um plano inclinado de

30° conforme a figura (sem escala) a seguir.



No trecho AB não existe atrito e no trecho BC o coeficiente de atrito vale mu = sqrt(3)/2. O bloco é abandonado, do repouso em relação ao plano inclinado, no ponto A e chega ao ponto C com velocidade nula. A altura do ponto A, em relação ao ponto B, é h1, e a altura do ponto B, em relação ao ponto C, é h2. Quanto vale a razão h1/h2?

REFERÊNCIAS

1. G. Van Rossum, F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace; 2009.
2. A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, A. Scopatz. *SymPy: symbolic computing in Python.* PeerJ Computer Science. v. 3, p. e103, 2017. Disponível em: https://doi.org/10.7717/peerj-cs.103.

ANEXOS

Será disponibilizado ao alunos via Google Classroom o código a seguir:

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/1Ano/Aula_6.py

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO

Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME

Professor: Bruno Camargo Braga

Componente Curricular: Física

Unidade temática: Investigação Científica e Processos Criativos.

Tema de estudo: Solução e Explicação das Atividades

Grupos 1 e 2: 1º ano do Ensino Médio	**Tempo:** 2 horas	**Data:** 26/05/2025

COMPETÊNCIAS E HABILIDADES

(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos;
(EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.

OBJETIVOS DE APRENDIZAGEM

Utilizando a linguagem de programação Python (versão 3.11), ser capaz de:

1. Compreender os possíveis erros quanto a solução de problemas de Física;
2. Encontrar soluções semelhantes a apresentada pelo professor;

RECURSOS/MATERIAIS DIDÁTICOS

Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.

PROCEDIMENTOS METODOLÓGICOS

1º momento (10 min): Saudações iniciais, chamada e anotação da ata de sala

2º momento (5 min): Nesta aula será apresentado as correções das atividades 1 a 5 feitas pelos alunos durante o projeto. Mencione aos estudantes que será apresentado sua solução como professor e que a aula será para tirar dúvidas.

OBS: os tempos dos momentos a seguir estão exagerados, pois é para usa-los não apenas para apresentar a solução do professor, mas para tirar dúvidas dos alunos e falar de problemas semelhantes caso nenhum aluno queira se pronunciar.

Reapresente as atividades propostas aos alunos uma a uma e discuta com eles

uma solução (e sua implementação em Python). As atividades foram:

1. ***3º momento (21 min): Faça um código que peça uma sequência de valores e calcule a média deles.***

   Explicação: *Esta atividade cobra o uso de comandos comuns de interação usuário-código e manipulação de textos. Uma solução é pedir que seja informado a sequência de valores em um formato específico, pois assim o programa conseguirá facilmente separar os valores para realizar a operação. O aluno deve lembrar que a resposta vem no formato texto e deve ser transformada para um tipo numérico. Utilizando um ciclo conseguimos somar todos os valores e apresentá-los usando o comando "print".*

2. ***4º momento (21 min): Faça uma código que peça um número inteiro e nos diz se o número é par ou ímpar.***

   Explicação: *Esta atividade cobra o uso de comandos comuns de interação usuário-código e manipulação de textos. Como o número informado deve ser par ou ímpar – ele será um número natural. Com isso basta converter para inteiro a reposta e utilizar o comando para resto de divisão "%". Se ao dividir por 2, der resto zero o número informado é par, caso contrário será ímpar.*

3. ***5º momento (21 min): Uma ave marinha costuma mergulhar de uma altura de 20 m para buscar alimento no mar. Suponha que um desses mergulhos tenha sido feito em sentido vertical, a partir do repouso. Desprezando-se as forças de atrito, a ave chegará à superfície do mar em qual velocidade em km/h?***

   Entregue um código que faça a solução do problema e imprime a resposta.
   Dica: use a definição de métodos "def".

   Explicação: *Nesta atividade basta utilizar a equação de Torricelli, na implementação usamos "def" para definir a equação e imprimimos o resultado utilizando as informações do problema.*

4. ***6º momento (21 min): Um trem de 150 m de comprimento se desloca com velocidade escalar constante de 16 m/s. Esse trem atravessa 3 túneis em dias diferentes e leva desde a entrada até a saída completa de cada túnel 50 s, 20 s e 90 s. Quanto é o comprimento de cada túnel?***
   Entregue um código que faça a solução do problema e imprime a resposta.
   Dica: use a função "lambda".

   Explicação: *Para solucionar o problema, primeiro consideramos o momento que a frente do trem alcança a abertura de cada túnel como t=0s. Para que o trem atravesse completamente o túnel, a parte traseira do trem deve percorrer o comprimento do trem (150 m), para alcançar o túnel e mais todo o comprimento do túnel (d metros) para atravessa-lo totalmente, logo:*

   $\Delta s = d + 150$ *(1)*

   $\Delta s = v * \Delta t,\ v = 16 m/s$ *(2)*

Substituindo (1) em (2)

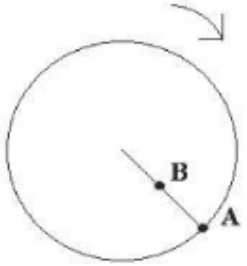
$$d + 150 = 16 * \Delta t,$$

Isolando o comprimento do túnel

$$d = 16 * \Delta t - 150 \text{ (3)}$$

Podemos então definir um método para a equação (3) usando a função lambda. Com isso basta substituir os intervalos de tempo de cada túnel de imprimir a respota.

5. ***7º momento (21 min): Considere uma polia girando em torno de seu eixo central, conforme figura abaixo. A velocidade dos pontos A e B são, respectivamente, 60 cm/s e 0,3 m/s. A distância AB vale 10 cm. O diâmetro e a velocidade angular da polia, respectivamente, valem:***



Explicação: *Nesse caso a velocidade angular de A e B são iguais.* $\omega_a = \omega_b$, $onde\, r * \omega = v.$ *Logo,* $\frac{v_a}{r} = \frac{v_b}{r-10}$, *invertendo as razões* $\frac{r}{v_a} = \frac{r-10}{v_b}$, *podemos agora separar a fração da direita e isolar o raio*

$\frac{r}{v_a} - \frac{r}{v_b} = -\frac{10}{v_b}$, *ou seja,* $r\left(\frac{1}{v_b} - \frac{1}{v_a}\right) = \frac{10}{v_b}$, *dividindo pelo termo das velocidades*

$$r = \frac{10}{v_b\left(\frac{1}{v_b} - \frac{1}{v_a}\right)} \quad (4)$$

Basta usar "def" para definir um método que calcula a equação (4) e devolve o dobro. Com o diâmetro e a velocidade linear de A, obtemos a velocidade angular.

AVALIAÇÃO

- Será feito uma avaliação pelo observado em sala.

REFERÊNCIAS

1. G. Van Rossum, F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace; 2009.
2. J. D. Hunter. *Matplotlib: A 2D Graphics Environment.*Computing in Science & Engineering. v. 9, n. 3, p. 90-95, 2007.
3. A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S.

Kulal, R. Cimrman, A. Scopatz. *SymPy: symbolic computing in Python.* PeerJ Computer Science. v. 3, p. e103, 2017. Disponível em: https://doi.org/10.7717/peerj-cs.103.

ANEXOS

Será disponibilizado ao alunos via Google Classroom os códigos discutidos em aula.

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/Atividade_1.py

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/Atividade_2.py

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/1Ano/Atividade_3.py

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/1Ano/Atividade_4.py

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/1Ano/Atividade_5_MCU.py

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO

Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME

Professor: Bruno Camargo Braga

Componente Curricular: Física

Unidade temática: Investigação Científica e Processos Criativos.

Tema de estudo: Solução e Explicação das Atividades – Parte 2

Grupos 1 e 2: 1º ano do Ensino Médio	**Tempo:** 2 horas	**Data:** 27/05/2025

COMPETÊNCIAS E HABILIDADES

(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos;
(EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.

OBJETIVOS DE APRENDIZAGEM

Utilizando a linguagem de programação Python (versão 3.11), ser capaz de:

1. Compreender os possíveis erros quanto a solução de problemas de Física;
2. Encontrar soluções semelhantes a apresentada pelo professor;

RECURSOS/MATERIAIS DIDÁTICOS

Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.

PROCEDIMENTOS METODOLÓGICOS

1º momento (10 min): Saudações iniciais, chamada e anotação da ata de sala. Nesta aula será apresentado as correções das atividades 6 a 9 feitas pelos alunos durante o projeto.

Reapresente as atividades propostas aos alunos uma a uma e discuta com eles uma solução (e sua implementação em Python). As atividades foram:

1. ***2º momento (27 min): Reutilizando o código visto na aula 5, implemente o produto de um vetor por um escalar. Ex: inicio: -> ; após produto por 3: --->***

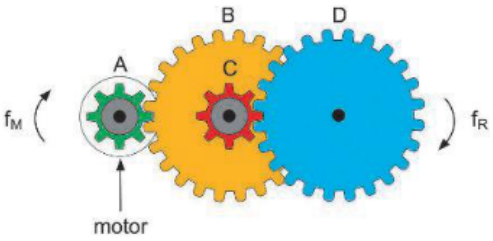
   Mostre a imagem do gráfico construído.

Explicação: *Nesta atividade o aluno foi permitido reutilizar o código visto em sala, logo, a forma mais fácil de resolver o exercício era copiar todo o código da classe de vetor e adicionar um método extra na classe – de produto escalar. A solução apresentada pelo professor usa o método de herança de classes, apenas por estética. Como feito em sala para a adição vetorial usando "__add__", a forma mais simples de realizar o produto é definindo o método "__mul__" e retornando nele um novo vetor com as coordenadas multiplicadas.*

2. ***3º momento (29 min): Em uma brincadeira de caça ao tesouro, o mapa diz que para chegar ao local onde a arca de ouro está enterrada, deve-se, primeiramente, dar 10 passos na direção norte, depois 12 passos para o leste, em seguida, 7 passos para o sul, e finalmente 8 passos para oeste. A partir dessas informações:***
    a. ***Construa um gráfico no Python com a trajetória descrita no mapa usando vetores***
    b. ***Se um caçador caminhasse em linha reta, desde o ponto de partida até o ponto de chegada, quantos passos ele daria? Mostre o vetor soma no gráfico.***

Explicação: *Nesta atividade basta utilizar as informações passadas no enunciado para construir os vetores (ex: 12 passos a leste pode ser representado como um vetor de tamanho 12, na horizontal para a direita). Reutilizando o código da aula 5 montamos a sequência de passos e ligamos os vetores (ponta do primeiro com o pé do segundo). Para responder tanto a letra (a) quanto a (b), basta realizar a soma vetorial e pedir para mostrar todos os vetores de uma vez.*

3. ***4º momento (27 min): Um pequeno motor a pilha é utilizado para movimentar um carrinho de brinquedo. Um sistema de engrenagens transforma a velocidade de rotação desse motor na velocidade de rotação adequada às rodas do carrinho. Esse sistema é formado por quatro engrenagens, A, B, C e D, sendo que A está presa ao eixo do motor, B e C estão presas a um segundo eixo e D a um terceiro eixo, no qual também estão presas duas das quatro rodas do carrinho.***



Nessas condições, quando o motor girar com frequência fM, as duas rodas do carrinho girarão com frequência fR. Sabendo que as engrenagens A e C possuem 8 dentes, que as engrenagens B e D possuem 24 dentes, que não há escorregamento entre elas e que fM=13.5Hz é correto afirmar que fR, em Hz, é igual a?Resolva o problema utilizando os comandos do sympy vistos na aula 6, e entregue o código usado.

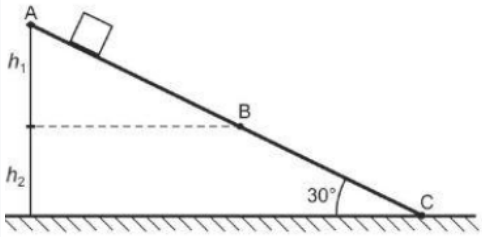
Explicação: *Recordando as equações:* $r * \omega = v;\ \omega = 2\pi * f$

No caso de engrenagens ligando os dentes $v_a = v_b$*, assim*

$r_a * \omega_a = r_b * \omega_b;\ \ r_a * f_a = r_b * f_b$*, onde* $f_a = f_m = 13.5Hz$

Precisamos determinar f_b*, para isso precisamos das relações de* r_a *e* r_b *com os dentes das engrenagens. Podemos supor que o comprimento de cada engrenagem é igual ao número de dentes, e relacionando com o comprimento da circunferência,* $2\pi * r = número\ de\ dentes.$ *Utilizando algum método de solução de equação do sympy visto em sala obtemos a resposta.*

4. ***5º momento (27 min): Um bloco escorrega, livre de resistência do ar, sobre um plano inclinado de 30° conforme a figura (sem escala) a seguir.***



No trecho AB não existe atrito e no trecho BC o coeficiente de atrito vale $\mu = \frac{\sqrt{3}}{2}$***. O bloco é abandonado, do repouso em relação ao plano inclinado, no ponto A e chega ao ponto C com velocidade nula. A altura do ponto A, em relação ao ponto B, é h1, e a altura do ponto B, em relação ao ponto C, é h2. Quanto vale a razão h1/h2?***

Explicação: *Para resolver o seno usando o "sympy", trocamos a unidade do ângulo para radiano,* $30° = \frac{\pi}{6}\ rad.$ *Podemos relacionar as alturas com os triângulos interno e externo,* $h_1 = AB * sen\left(\frac{\pi}{6}\right);\ \ h1 + h2 = AC * sen\left(\frac{\pi}{6}\right)$*, isolando o* $sen\left(\frac{\pi}{6}\right)$ *podemos unir as equações, logo* $\frac{h_1}{AB} = \frac{(h1+h2)}{AC}$*. Ao separar a fração da direita podemos isolar o termo com* h_1*,*

$h_1 * \left(\frac{1}{AB} - \frac{1}{AC}\right) = \frac{h_2}{AC}$*, agora fica claro como determinaremos a razão das alturas.*

$\frac{h_1}{h_2} = \frac{1}{\left(AC*\left(\frac{1}{AB} - \frac{1}{AC}\right)\right)}$, $onde\ AC = AB + BC$

$\frac{h_1}{h_2} = \frac{1}{(AB+BC)*\left(\frac{1}{AB} - \frac{1}{(AB+BC)}\right)}\ (Eq1)$

Para resolver (Eq1) precisamos dos valores de AB e BC, mas a única informação que temos é que o bloco partiu do repouso e parou ao final da

descida. Com isso usaremos a equação de Torricelli, assim

$v_B^{\,2} = v_A^{\,2} + 2 * a * AB$, *onde* $v_A = 0$ *e* $a = g * sen\left(\frac{\pi}{6}\right)$, *pois não tem atrito*

$v_B^{\,2} = 2 * g * sen\left(\frac{\pi}{6}\right) * AB;$

Em BC devemos considerar a força de atrito para calcular a aceleração a_{BC}, *então,*

$v_C^{\,2} = v_B^{\,2} + 2 * a_{BC} * BC$, $v_C = 0$ *pois o bloco para. Substituindo o valor encontrado de* $v_B^{\,2}$,

$0 = 2 * g * sen\left(\frac{\pi}{6}\right) * AB + 2 * a_{BC} * BC \quad (Eq2)$

Conseguimos a aceleração utilizando a 2ª lei de Newton,

$\left|\vec{F_R}\right| = \left|\vec{P} * sen\left(\frac{\pi}{6}\right)\right| - \left|\vec{F_{at}}\right|$, *onde* $\left|\vec{F_{at}}\right| = \left|\vec{N}\right| * \mu = \left|\vec{P}\right| * cos\left(\frac{\pi}{6}\right) * \mu$

Substituindo $\left|\vec{F_R}\right| = m * a_{BC}$, $\left|\vec{F_{at}}\right| = \left|\vec{P}\right| * cos\left(\frac{\pi}{6}\right) * \mu$, *e* $\left|\vec{P}\right| = m * g$

$m * a_{BC} = m * g * sen\left(\frac{\pi}{6}\right) - m * g * cos\left(\frac{\pi}{6}\right) * \frac{\sqrt{3}}{2}$, *portanto,*

$a_{BC} = g * \left(sen\left(\frac{\pi}{6}\right) - \frac{\sqrt{3}}{2} cos\left(\frac{\pi}{6}\right)\right) \quad (Eq3)$

Agora precisamo apenas definir as equações Eq1, Eq2 e Eq3 no Python para solucionar o problema. Outra solução seria implementar parte da solução feita a mão para o Python e resolver um sistema maior de equações.

AVALIAÇÃO

- Será feito uma avaliação pelo observado em sala.

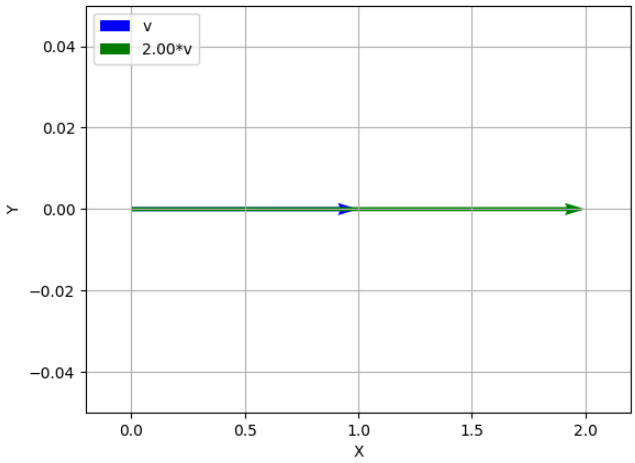
REFERÊNCIAS

1. G. Van Rossum, F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace; 2009.
2. J. D. Hunter. *Matplotlib: A 2D Graphics Environment.*Computing in Science & Engineering. v. 9, n. 3, p. 90-95, 2007.
3. A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, A. Scopatz. *SymPy: symbolic computing in Python.* PeerJ Computer Science. v. 3, p. e103, 2017. Disponível em: https://doi.org/10.7717/peerj-cs.103.

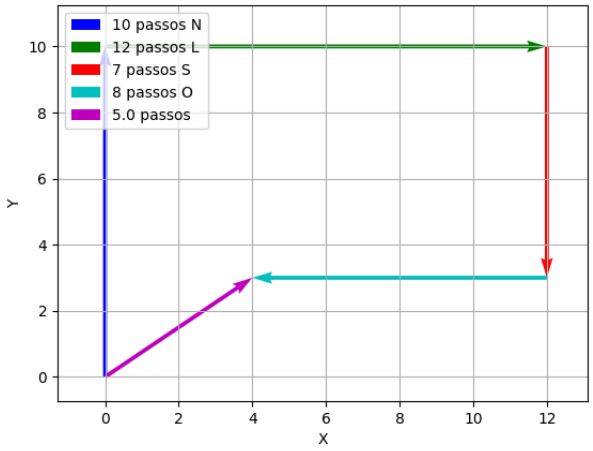
ANEXOS

Será disponibilizado ao alunos via Google Classroom os códigos discutidos em aula.

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/1Ano/Atividade_6.py



https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/1Ano/Atividade_7.py



https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/1Ano/Atividade_8_Engrenagens.py

https://github.com/Blodhor/Programando_Em_Python_EnsinoMedio/blob/main/1Ano/Atividade_9.py

MODELO DE PLANEJAMENTO/PLANO DE AULA PARA CADA ENCONTRO

IDENTIFICAÇÃO

Instituição Concedente: Colégio Impacto Centro de Ensino LTDA – ME

Professor: Bruno Camargo Braga

Componente Curricular: Física

Unidade temática: Investigação Científica e Processos Criativos.

Tema de estudo: Avaliação e depoimento

Grupos 1 e 2: 1º ano do Ensino Médio	**Tempo:** 2 horas	**Data:** 03/06/2025

COMPETÊNCIAS E HABILIDADES

(EMIFCG03) Utilizar informações, conhecimentos e ideias resultantes de investigações científicas para criar ou propor soluções para problemas diversos;
(EMIFCG05) Questionar, modificar e adaptar ideias existentes e criar propostas obras ou soluções criativas, originais ou inovadoras, avaliando e assumindo riscos para lidar com as incertezas e colocá-las em prática.

OBJETIVOS DE APRENDIZAGEM

Utilizando a linguagem de programação Python (versão 3.11), ser capaz de:

1. Conseguir construir e ler corretamente gráficos;
2. Resolver sistemas de equações.

RECURSOS/MATERIAIS DIDÁTICOS

Quadro, smartphone, Google Classroom, Google Colab, Pydroid 3.

PROCEDIMENTOS METODOLÓGICOS

1º momento (10 min): Saudações iniciais, chamada e anotação da ata de sala.

2º momento (55 min): **Reaplicação da avaliação diagnóstica**

3º momento (55 min): **Deixe os alunos que não terminaram a avaliação faze-la. Indique aos que terminaram para preencher o depoimento de participação via Google Forms.**

AVALIAÇÃO

Atividades recomendadas:

- Refazer a avaliação diagnóstica;
- Responder o depoimento de participação.

REFERÊNCIAS

1. G. Van Rossum, F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace; 2009.
2. J. D. Hunter. *Matplotlib: A 2D Graphics Environment.*Computing in Science & Engineering. v. 9, n. 3, p. 90-95, 2007.
3. A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, A. Scopatz. *SymPy: symbolic computing in Python.* PeerJ Computer Science. v. 3, p. e103, 2017. Disponível em: https://doi.org/10.7717/peerj-cs.103.

ANEXOS

A avaliação diagnóstica será a mesma descrita no encontro 1.

Descrição do formulário do depoimento de participação:

1. Nome do aluno(a):
2. Como você descreveria sua experiência geral neste projeto?
3. Quais foram os aspectos mais positivos do projeto?
4. Houve alguma dificuldade que você enfrentou antes de começar o projeto (ex: dificuldade em resolver equações ou utilizar gráficos)? Como o programa ajudou a supera-la?
5. Como você se sente em relação ao seu desempenho escolar após participar do projeto?
6. Alguma sugestão ou comentário adicional que gostaria de compartilhar?
7. **Permissão para Uso de Depoimento:**
   Você autoriza o uso do seu depoimento em materiais de divulgação da universidade Uniasselvi?
   a. Sim
   b. Não