

O uso do python na construção de simuladores computacionais: proposições e potencialidades para o ensino de Física^{+,*}

Lucas Queiroz Galvão¹

Licenciando em Física – Universidade Federal do Oeste da Bahia
Barreiras – BA

Suiane Ewerling da Rosa¹

Grupo de Diálogos FormAtivos em Educação em Ciências
Universidade Federal do Oeste da Bahia
Barreiras – BA

Wanisson Silva Santana¹

Clebson dos Santos Cruz¹

Grupo de Informação Quântica e Física Estatística
Universidade Federal do Oeste da Bahia
Barreiras – BA

Resumo

O uso das tecnologias da informação e comunicação tem se mostrado uma alternativa de extrema importância e relevância para o ensino de Física, em especial no contexto atualmente vivido, marcado pela pandemia do novo coronavírus (SARS-CoV-2), e que implicou na adaptação de tarefas de diferentes naturezas, como na educação. Nesse sentido, este trabalho tem como intuito discutir o potencial das simulações computacionais, especificamente em python e vpython, como recursos de desenvolvimento de simuladores relacionadas a fenômenos da natureza abordados no estudo da mecânica. Para tanto, é apresentada, inicialmente, discussões acerca do tema em questão e, em seguida, a lógica de construção desses simuladores, assim como exemplos propositivos para um laboratório virtual de mecânica com o python e vpython de modo a salientar as suas múltiplas possibilidades de

⁺The use of python in the production of computer simulators: propositions and potentialities for Physics teaching

^{*} Recebido: 11 de junho de 2021.

Aceito: 30 de novembro de 2021.

¹ E-mails: lucas.g5847@ufob.edu.br; suiane.rosa@ufob.edu.br; wanisson.santana@ufob.edu.br; clebson.cruz@ufob.edu.br;

produção e utilização. Este trabalho busca inserir os/as professores/as e estudantes no atual contexto da programação a partir do desenvolvimento dos seus próprios simuladores, apresentando o python e vpython como ferramentas potencializadoras para o processo de ensino-aprendizagem da Física em diferentes níveis de ensino.

Palavras-chave: *Ensino de Física; Simulador Computacional; Python; Vpython.*

Abstract

The use of information and communication technologies has shown an effective alternative and relevance to Physics teaching, especially in the current context marked by the pandemic of the new coronavirus (SARS-CoV-2), and that implied in the adaptation of tasks of different natures, as in education. Accordingly, this work aims to discourse the potential of computer simulations, specifically in python and vpython, as resources of development of simulators related to natural phenomena addressed in the study of mechanics. For this purpose, it is presented, firstly, discussions about the topic in question and then the construction logic of these simulators, and prepositional examples for a virtual mechanics laboratory with python and vpython to saliently highlight their multiple productions and use possibilities. This paper seeks to insert teachers in the programming context with the development of their own simulators, presenting python and vpython as potentializing tools for the teaching-learning process of Physics at different levels of education.

Keywords: *Physics Teaching; Computer Simulator; Python; Vpython.*

I. Introdução

A importância de um ensino difundido da Física nos mais diversos espaços educacionais é inquestionável, seja por possibilitar aos/às estudantes a compreensão, sob o olhar da ciência, de como funcionam os fenômenos da natureza, seja por apresentá-los/as a um conhecimento que possibilite investigar, se apropriar e atuar sobre temas da área científico-tecnológica, contribuindo para uma formação mais plena desses sujeitos, para a constituição de cidadãos e cidadãs críticos/as e alfabetizados/as científico e tecnologicamente. No entanto, não é novidade que muitas são as dificuldades sentidas pelos/as estudantes na compreensão de fenômenos físicos (FIOLHAIS; TRINDADE, 2003). Motivados por essas

demandas e influenciados pelo crescente desenvolvimento das tecnologias da informação e comunicação (TIC), diferentes estudos na área (FIOLHAIS; TRINDADE, 2003; CENTENARO *et al.*, 2017; PASTORIO, 2018) têm sido desenvolvidos com intuito de investigar métodos e abordagens que possibilitem novas formas de pensar, planejar e estruturar ações educativas. Entretanto, é inegável que muitos impasses e desafios nesse contexto ainda são presentes

O processo de ensino-aprendizagem da Física, em especial os fundamentados por um sistema de ensino tradicional², mecanizado e centrado no/a professor/a, é, a muito tempo, alvo de problematizações na área. Todavia, essas abordagens ainda são frequentes no cotidiano escolar (MOREIRA, 2021). Estudiosos da área, como os já citados, destacam a falta de métodos e ferramentas educacionais mais modernas como um dos motivos que tem implicado na aprendizagem dos/as estudantes e no distanciamento com relação ao atendimento de diferentes demandas presentes no ensino da Física. Dentre esses apontamentos, Fiolhais e Trindade (2003), já no início dos anos 2000, anunciavam que, entre as razões dos insucessos da aprendizagem em Física, estariam alguns métodos de ensino utilizados por parte dos/as professores/as, métodos desajustados em relação às “teorias de aprendizagem mais recentes e não utilização dos meios mais modernos” (p. 259). Passado mais de quinze anos das problematizações anunciadas por esses autores, estudos recentes ainda revelam as mesmas problemáticas; isso porque muitos/as professores/as insistem no uso exclusivo de métodos tradicionais de ensino (HEINECK; VALIATI; ROSA, 2017), evitando novas abordagens que possibilitariam aos/as estudantes uma maior motivação e engajamento com relação à área e uma melhor apropriação dos conteúdos. No entanto, como evidenciam Lawson e McDermott (1987), não é de se admirar limitações na aprendizagem se conhecimentos complexos só forem apresentados de forma verbal e/ou textual.

Em um recente artigo, Moreira (2021) aponta a ausência de laboratórios virtuais e metodologias relacionadas a recursos tecnológicos digitais como um dos principais desafios para o ensino de Física. Ao longo do texto, o autor argumenta que esses recursos fazem parte da realidade dos/as estudantes e, por conseguinte, devem ser aderidos às aulas, seja por meio de simuladores computacionais, modelagem computacional e/ou experimentos virtuais. No caso dos simuladores computacionais, ênfase deste trabalho, as potencialidades se mostram ainda mais significativas, uma vez que, dentre outros pontos, são capazes de representar aspectos do mundo real de forma relativamente satisfatória (PASTORIO, 2018). Com eles, é possível ter uma maior interatividade com os conteúdos apresentados em sala, visto que “o programa é capaz de oferecer não apenas uma animação isolada de um fenômeno em causa;

² Sistema de ensino ao qual “o professor utiliza apenas o quadro negro e o giz, e parece deter o conhecimento, enumera informações para os estudantes que tentam absorver e interpretar essas informações” (PASTORIO, 2018, p. 36). É importante ressaltar que este trabalho não pretende tecer críticas ao sistema de ensino tradicional, mas analisar como o uso exclusivo de métodos tradicionais limita o processo de ensino-aprendizagem e propósitos educacionais ampliados.

mas uma vasta gama de animações alternativas selecionadas através de um input de parâmetros” (MEDEIROS; MEDEIROS, 2002, p. 79).

Portanto, este trabalho, incluído no contexto de pesquisas em ensino de Física, que se atenta às inquietações evidenciadas, tem como finalidade apresentar as ferramentas computacionais python, sua biblioteca gráfica vpython e o *framework* tkinter como recursos de desenvolvimento de simulações computacionais relacionadas a fenômenos da natureza abordados no estudo da mecânica. Para tanto, analisamos seus potenciais e desafios para resolução dos problemas referentes à falta de interação nas aulas e de implicações voltadas para o ensino-aprendizagem da Física. Complementar à análise, apresentamos alguns exemplos de simuladores construídos com python, vpython e tkinter de modo a explicitar esses aspectos, a saber: lançamento oblíquo, movimento retilíneo uniforme e o sistema com polia fixa. 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. Já a escolha pelo vpython se dá, principalmente, pela possibilidade de o programador incrementar, de forma prática, elementos geométricos tridimensionais em sua interface, atribuindo-lhes cor, tamanho, posição, massa e velocidade.

Atentando-se a essas potencialidades, entendemos como importante também identificar se essas ferramentas têm sido trabalhadas no contexto de pesquisas da área. Nesse sentido, realizamos um levantamento em periódicos e eventos nacionais da área de Ensino de Física que utilizam simuladores computacionais construídos em python e vpython com intuito de investigar a presença e, também, limitações e potencialidades frente ao tema investigado. Cabe dizer, portanto, que o foco deste trabalho não é de uma apresentação sistemática da produção da área, como na linha de pesquisa de revisão ou investigação da literatura, mas tem como objetivo apresentar ferramentas computacionais python, sua biblioteca gráfica vpython e o *framework* tkinter como recursos tecnológicos e digitais potencialmente significativos para o ensino da Física.

II. Simuladores Computacionais no Ensino de Física: perspectivas educacionais e possibilidades para a área

Os recursos tecnológicos e digitais vêm ganhando, paulatinamente, um maior espaço no cotidiano da população como ferramentas auxiliares em tarefas de diversas naturezas, tornando-se, em certos casos, essenciais para a realização dessas. Com a pandemia acometida pelo novo coronavírus (SARS-CoV-2), responsável pela doença Covid-19, por exemplo, tais recursos se mostraram indispensáveis para dar continuidade a diversas atividades de extrema importância social, econômica e política no país, tais como a educação. Nesse contexto, algumas instituições de ensino aderiram ao seu uso, fomentando uma popularização ainda maior dessas ferramentas nos ambientes educacionais por meio de atividades remotas. No

entanto, apesar do seu atual engajamento, é importante salientar que houve, em paralelo com a computação, um longo processo de investigação e produção científico-tecnológico responsável por moldar as suas atuais características.

Como marco inicial desse processo, a criação do computador pessoal durante a segunda metade da década de 1970 (LAURINDO; CARVALHO, 2003) já apontava para possibilidades de uso de TIC em diversos espaços. Anos mais tarde, concomitante à democratização desses computadores, as TIC ganharam uma massiva popularização. Com esses avanços, outros recursos, para além dos computadores *desktop*, foram sendo desenvolvidos, ampliando as suas possibilidades de produção. Segundo Fiolhais e Trindade (2003), esses fatos foram capazes de proporcionar “novas oportunidades de usar tecnologias da informação na educação e de concretizar com elas novas formas de aprendizagem” (p. 260). Nesse sentido, surge uma nova perspectiva em relação ao ensino, levando autores, como os já citados, a classificarem o aparecimento dos computadores no ensino como uma nova revolução educacional.

Para além disso, o surgimento da internet (rede global de computadores) durante a década de 1980 se destaca como outro importante marco para a consolidação das TIC. Ao final dessa década, foi criada a World Wide Web (WWW), que só ganhou popularidade nos anos 1990 (FIOLHAIS; TRINDADE, 2003). Com ela, tornou-se possível o compartilhamento de dados em todo o lugar do planeta, o que gerou uma difusão ainda maior das TIC e suas aplicações no ensino. Outrossim, ainda nesse período, os computadores foram ganhando melhorias gráficas e uma maior capacidade de processamento à medida em que ficavam cada vez mais baratos (FIOLHAIS; TRINDADE, 2003; LAURINDO; CARVALHO, 2003).

No início do século XXI, a portabilidade dos computadores progrediu a tal ponto que se tornou possível levar um computador com aspectos superiores aos desenvolvidos durante as décadas anteriores no bolso (*smartphones*) por um preço relativamente acessível à maioria. Com esses equipamentos cada vez mais próximos do cotidiano das pessoas, somados às vantagens proporcionadas pelas redes móveis, grande parte da população vem sendo capaz de ter acesso a uma imensa gama de informações de forma rápida e fácil (PASTORIO, 2018). Logo, tornou-se comum, por exemplo, ver uma grande quantidade de estudantes em uma sala de aula possuindo ao menos um computador (*smartphone*, *notebook*, *desktop*, entre outros).

Com o surgimento dessas tecnologias digitais, cunhou-se um novo termo, que tem sido cada vez mais utilizado, para se referir a esses recursos: Tecnologias Digitais de Informação e Comunicação (TDIC). Enquanto as TIC englobam recursos que vão desde os analógicos até os digitais, as TDIC dão ênfase exclusiva ao digital, a exemplo dos computadores e da internet (AFONSO, 2002). Referente a isso, Baranauskas e Valente (2013) destacam que as TDIC contribuem significativamente na “na constituição da sociedade e em nossas vidas, em termos de valores, atitudes, convenções e práticas sociais e econômicas que engendram” (p. 2). Essa relação, segundo os autores, que não é neutra, reflete em

implicações sociais, ao mesmo tempo que nos coloca como responsáveis pelos direcionamentos e consequências oriundas dela.

Tendo em vista a atual abrangência dos meios digitais, alguns autores utilizam o termo TDIC (a exemplo de: FEITOSA, 2015; JOSÉ; ANGOTTI; BASTOS, 2016; RANGEL; SANTOS; RIBEIRO, 2012; JAIME; FILHO; LEONEL, 2021), enquanto outros, por questões de maior abrangência e consolidação da área, optam ainda pelo termo TIC (a exemplo de: PASTORIO, 2018; FERNANDES; RODRIGUES; FERREIRA, 2015; MACÊDO *et al.*, 2013). Dessa forma, por integrar aspectos da realidade vivenciada pelos/as estudantes, as TIC ou TDIC têm feito parte de estudos e investigações no contexto de pesquisas educacionais, sinalizando contribuições sobre os processos de ensino-aprendizagem nas diferentes componentes curriculares da educação básica (ROSA; SOSO; DARROZ, 2018).

No entanto, apesar da atual abrangência, salienta-se que a completa acessibilidade aos meios digitais não é um objetivo alcançado, visto que, segundo uma pesquisa realizada em 2018 pelo Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação (CETIC, 2018), 42% das casas no Brasil não possuem um computador em casa, o que demonstra uma diferença socioeconômica relevante em relação à possibilidade de acesso a essas ferramentas tecnológicas digitais. Atualmente, verifica-se que o cenário não teve mudanças expressivas, posto que um dos maiores desafios para dar continuidade às aulas *online* durante a pandemia acometida pela Covid-19 foi, justamente, a impossibilidade de parte dos/as estudantes terem acesso a meios digitais que permitam a realização das práticas educativas remotas. Portanto, ainda existem desafios para a construção de uma infoinclusão³.

Entretanto, cabe destacar que o constante aumento na adesão desses recursos nos diferentes espaços sociais tem crescido ao ponto de termos como “sociedade da informação” serem amplamente utilizados atualmente. Isso levou as escolas, como instituições comprometidas em formar cidadãos e cidadãs críticos/as e alfabetizados/as científico e tecnologicamente, a terem como um de seus maiores desafios “adaptar a educação às tecnologias de informação e comunicação tais como a internet, a televisão, o rádio, os softwares, que funcionam como meios educativos formais ou informais” (MACÊDO *et al.*, 2013, p. 172). No Brasil, por exemplo, a inserção do programa de inclusão digital no Plano de Desenvolvimento da Educação em 2005 foi implementada com o intuito de proporcionar “a instalação de computadores e laboratórios multimídia em todas as escolas públicas e a produção de material de apoio de suporte para os/as professores/as através do Portal do Professor” (PASTORIO, 2018, p. 35). Atualmente, os impactos da pandemia têm implicado na utilização gradual de recursos tecnológicos digitais para a continuidade das aulas, compelindo os/as professores/as a se familiarizarem com tais ferramentas, assim como

³ Processo democrático de inclusão digital dos indivíduos. Cabe destacar ainda que a infoinclusão está para além do simples acesso a recursos tecnológicos e digitais, sendo necessário uma inserção qualitativa no contexto informacional (SANTOS *et al.*, 2007).

identificarem os seus alcances e limitações para a adaptação do Ensino Remoto Emergencial (JAIME; FILHO; LEONEL, 2021).

Nesse contexto, devido à imensa gama de possibilidades de utilização dos recursos tecnológicos e digitais em sala de aula, seja como hipertextos que proporcionem uma leitura fluida aos/às estudantes, multimídias que oferecem uma melhor percepção visual e sonora de determinados problemas ou modelagem e simulação computacional capazes de dinamizar as aulas, surgem novas propostas didático-pedagógicas que as inserem em diversas atividades metodológicas com a intenção de potencializar o ensino. Tendo em vista seus aspectos e objetivos, os simuladores computacionais ganham ênfase em áreas de ensino que possibilitem o uso de modelagem, tais como as ciências exatas/naturais. Na disciplina de Física, por exemplo, em que se é exigido um alto grau de abstração dos objetos de conhecimento por trás dos fenômenos investigados/trabalhados, o seu uso se mostra um possível caminho para uma abordagem mais ampla desses fenômenos, principalmente, nos casos em que os experimentos se tornam inexecutáveis de serem reproduzidos na prática devido às suas condições físicas incapazes de serem replicadas de maneira observável em experimentos.

Além disso, Gaddis (2000), citado por Pastorio (2018), destaca que os simuladores computacionais “englobam uma vasta classe de tecnologias, do vídeo à realidade virtual, que podem ser classificadas em certas categorias gerais baseadas fundamentalmente no grau de interatividade entre aprendiz e computador” (p. 39). Assim, um bom simulador computacional tem que ser capaz de oferecer ao/à usuário/a, dentre outros aspectos, uma boa interatividade, permitindo a eles inserir variáveis por meio do comando *input* e observar os resultados na tela por meio de animações alternativas (MEDEIROS; MEDEIROS, 2002). Ou seja, os/as estudantes inserem dados nesses programas e, logo após, é mostrado como determinada situação deve ocorrer baseando-se neles. De acordo com Medeiros e Medeiros (2002), diversos fenômenos podem ser modelados nesses simuladores, a exemplo dos lançamentos, pois:

Para ilustrar o movimento de um projétil, uma simulação computacional permite ao estudante a escolha de parâmetros relevantes tais como a velocidade inicial e o ângulo do tiro, para os quais o programa fornece as respectivas animações geradas a partir de grandes bancos de dados (p. 79).

Desse modo, é possível que o/a estudante acompanhe o fenômeno e perceba a sua variação à medida que novos dados são inseridos. Com isso, os fenômenos físicos ganham uma abordagem mais dinâmica, assim como as aulas se tornam mais atraentes aos/às estudantes. Outrossim, para os casos em que a realização de alguns experimentos na física é inviável, seja por serem muito caros, ou muito pequenos, ou ainda muito grandes, os simuladores surgem como possibilidade, pois com apenas um computador de qualidade e o programa ideal, é possível modelar diversos fenômenos nas mais variadas escalas (PASTORIO, 2018).

Na utilização dos simuladores computacionais, cabe destacar três práticas amplamente adotadas: (i) o uso de simuladores já construídos; (ii) a construção de simuladores a partir de ambientes pré-programados; (iii) a construção de simuladores a partir da linguagem de programação. No primeiro caso (i), a ênfase é dada exclusivamente ao uso de simuladores “prontos”, que são comumente encontrados em laboratórios virtuais, a exemplo do Phet. Com esse tipo de atividade, questões referentes à interatividade dos/as estudantes e dinamicidade das aulas se fazem presentes a partir da inserção dos dados solicitados e geração de diferentes animações (MEDEIROS; MEDEIROS, 2002). Todavia, no caso das atividades de construção de simuladores computacionais, o processo de ensino-aprendizagem da Física pode ganhar ainda mais potencialidades. Isso porque as simulações são desenvolvidas pelo/a próprio/a usuário/a, o que torna necessário um conhecimento ainda mais aprofundado do fenômeno abordado para simular o seu comportamento. Assim, enquanto nesse tipo de atividade o/a usuário/a tem uma maior liberdade criativa para investigar aspectos presentes na transposição de um fenômeno da realidade para o ambiente virtual, no uso de simuladores “prontos” ele é limitado pelas possibilidades de utilização da simulação já desenvolvida.

Ainda sobre a construção de simuladores, a atividade caracterizada pela utilização de ambientes pré-programados (ii) é marcada pela assimilação de ícones específicos para a inserção de figuras geométricas, construção de gráficos, desenho de trajetórias, dentre outros. Ou seja, nesse tipo de atividade, a iconografia é predominante, sem a necessidade de o/a usuário/a saber a linguagem de programação por trás do simulador (GIORDAN, 2005). Com isso, ele/a pode se concentrar em construir sua própria narrativa na transposição do fenômeno para o ambiente virtual, sem, contudo, ignorar seus aspectos. Aplicativos como Modellus e STELLA destacam-se como ambientes que permitem esse tipo de construção.

No caso da atividade de construção de simuladores a partir da linguagem de programação (iii), o/a usuário/a cria sua própria ferramenta para o desenvolvimento das simulações, transpondo as características do fenômeno abordado por meio da linguagem de programação (GIORDAN, 2005). Nessa atividade, o desenvolvimento dos simuladores perpassa por um processo de interação entre a programação e a Física, visto que é preciso conhecer aspectos do fenômeno abordado para a construção das simulações. É importante salientar que, nesse caso, as possibilidades de criação de simuladores são ainda maiores se comparado a ambientes pré-programados, uma vez que o/a usuário/a é limitado apenas pelas possibilidades de desenvolvimento de *softwares* oferecidas pela linguagem, enquanto que nos ambientes pré-programados ele é limitado pelos recursos que esse ambiente oferece.

Dentre as linguagens de programação que se destacam por facilitarem a construção dos simuladores, o python, em conjunto com a sua biblioteca gráfica vpython, ganham uma ênfase maior devido à sua capacidade de inserir objetos tridimensionais dinâmicos em sua interface, além de possuir uma sintaxe robusta e várias bibliotecas capazes de auxiliar na criação de códigos com aplicabilidades na Matemática e na Física (BORCHERDS, 2007). Logo, por meio do vpython, pode-se criar simulações capazes de representar os fenômenos da

natureza de forma intuitiva e com uma ampla gama de possibilidades a serem trabalhadas nas aulas.

Todavia, apesar das possibilidades de criação e utilização desses simuladores, cabe ressaltar que os recursos digitais e tecnológicos não são e não devem ser tratadas como a solução para todos os impasses presentes no ensino de Física, mas como ferramentas capazes de contribuir significativamente se usadas de maneira adequada. Pois, segundo Medeiros e Medeiros (2002), é preciso refletir “a construção do conhecimento em um contexto mais amplo que englobe os conteúdos e os seus processos de construção” (p. 77). Logo, os simuladores computacionais não podem ser aplicados e trabalhados de modo reducionista; é preciso que os/as estudantes estejam cientes de que elas são ferramentas de apoio, tendo como base algoritmos baseados nos conhecimentos trabalhados em sala de aula.

Desse modo, considerando as suas potencialidades e limitações, os simuladores computacionais se mostram uma opção capaz de amenizar alguns impasses presentes no atual modelo de ensino de Física. Com as várias formas de aplicabilidade desses instrumentos no processo de ensino-aprendizagem, os métodos de ensino ganham novas abordagens, podendo se tornar mais atraentes tanto para os/as professores/as que podem avaliá-las como ferramentas adequadas ao ensino, quanto para os/as estudantes que, em sua maior parte, podem ter contato com meios e recursos de ensino diversificados.

III. Caracterização do Trabalho

Com vista às discussões realizadas, denota-se que os simuladores computacionais apresentam características capazes de potencializar o processo de ensino-aprendizagem da Física, proporcionando abordagens mais amplas de uma situação problema. Eles são capazes de promover uma maior dinamicidade às aulas, tornando possível o desempenho de um papel ainda mais ativo por parte dos/as estudantes na construção do conhecimento, e, em alguns casos, uma investigação para além da sala de aula. Em se tratando do python e a sua biblioteca gráfica vpython, ênfases deste trabalho, as possibilidades de construção desses simuladores são inúmeras, pois, além de proporcionar uma maior liberdade para criar e manipular figuras geométricas tridimensionais, o python concentra uma vasta gama de módulos e bibliotecas embasadas em equações físicas e matemáticas. Outro ponto de destaque é a facilidade em se trabalhar com a linguagem devido à sua sintaxe robusta e código aberto, que permite aos/às usuários/as ter acesso livre aos algoritmos, possibilitando-os/as entender, na prática, como eles foram desenvolvidos – baseando-se nas leis da Física –, bem como podendo modificá-los posteriormente.

Para uma maior fundamentação da pesquisa e identificação da presença, alcance e utilização da linguagem de programação python e a sua biblioteca gráfica vpython como estratégia didático-pedagógica no contexto de práticas educativas do ensino de Física, realizamos um levantamento de trabalhos que abordam o uso e/ou a construção de simuladores em python publicados nos principais periódicos e anais de eventos da área de

ensino de Física no contexto nacional. Para tanto, consideramos 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), buscados a partir das seguintes palavras-chave: python e vpython. Como resultado, encontramos um total de sete trabalhos. No SNEF, foram encontrados dois trabalhos: (1) Noções de programação estruturada em python no Ensino de Física: cultura lúdica e dificuldades de Aplicação (PARIZOTTO; SOARES, 2017); (2) Um programa em python para experimentos de Física automatizados com o Arduino (MARTINS; CORDEIRO; LOUREIRO, 2019). No EPEF, apenas um trabalho: (3) Projetor estelar: uma ferramenta tecnológica para o ensino de astronomia e observação do céu a olho nu (SILVA; PEDERSEN; CARVALHO-NETO, 2020). Na RBEF, encontramos um número maior, tendo 4 trabalhos: (4) Software MUFCosm como ferramenta de estudo dos modelos da cosmologia padrão (CUZINATTO; MORAIS, 2014); (5) Visualização da forma de onda e conteúdo harmônico da corrente elétrica alternada em eletrodomésticos (DIONÍSIO; SPALDING, 2016); (6) Aplicação do problema restrito de três corpos no estudo do movimento de astros do sistema solar (MACEDO; JUNIOR, 2018); (7) Análise do movimento de um ponto de luz sobre um plano inclinado (FREITAS-LEMOS *et al.*, 2019). Por outro lado, no CBEF não foi encontrado nenhum trabalho com essas características.

A fim de discutir os alcances, potencialidades e limitações desses trabalhos, apresentaremos, a seguir, um breve relato deles, tendo por objetivo caracterizar as investigações e propostas desenvolvidas. No entanto, destacamos novamente que o nosso intuito não está em realizar uma sistematização da área, mas evidenciar perspectivas de trabalhos e propostas alinhadas ao tema de interesse. Ou seja, o intuito não está em descrever detalhadamente as práticas/propostas, mas situar os objetivos centrais delas frente ao uso do python e vpython na construção de simuladores computacionais, apresentando um breve panorama do tema no lócus investigado.

No primeiro trabalho (1), os autores apresentam os resultados obtidos durante a etapa inicial da efetivação de uma proposta didático-pedagógica que traz como objetivo associar os conteúdos ministrados na disciplina de Física da educação básica com a criação de jogos através da linguagem de programação python. No trabalho (2), os autores apresentam a aplicação de um programa computacional autoral, associado à robótica (arduino), desenvolvido em python para três diferentes experimentos: pêndulo simples, carga e descarga de um capacitor e absorção de radiação eletromagnética. Já no trabalho (3), com o intuito de potencializar o ensino da astronomia observacional em espaços não formais, os autores apresentaram uma TIC, denominada de Projeto Estelar, capaz de explorar o céu noturno a olho nu. O Projeto Estelar é composto por um motor fixo, um motor móvel, um Arduino Uno, um laser, um computador e um programa desenvolvido em python.

No que se refere aos trabalhos da RBEF, há uma ênfase maior para o uso de simuladores computacionais na Educação Superior. Assim, no quarto trabalho (4), os autores

buscam retratar as distintas fases da evolução do universo na escala cosmológica através do modelo unificado para o fluido cosmológico, utilizando o programa MUFCosm, construído em python, como possível ferramenta para a abordagem desse fenômeno. No programa, é possível construir gráficos capazes de modelar equações que representam o comportamento do universo nas diferentes etapas da sua evolução. No trabalho (5), os autores apresentam um projeto desenvolvido utilizando o arduíno, um circuito elétrico e um programa computacional construído em python capaz de, através da medição da corrente elétrica alternada de alguns eletrodomésticos, gerar gráficos que possibilitem a visualização da forma de onda da corrente desses equipamentos, assim como dos seus respectivos espectros de frequência. No trabalho (6), tendo em vista a pouca abordagem do Problema Restrito de Três Corpos (PRTC) em situações reais e contextualizadas nas aulas de Física na graduação – em especial, na disciplina de Mecânica Clássica –, os autores trazem um programa escrito em código python capaz de abordar esse fenômeno de forma didaticamente ampla, explorando as trajetórias, valores e comportamentos orbitais dos astros. Por fim, no sétimo trabalho (7), tomando como referência de abordagem pedagógica o método *Problem Based Learning* (PBL), os autores propõem a análise do movimento de um ponto de luz sobre um plano inclinado por meio de um experimento realizado por estudantes de uma turma de Física Experimental I do curso de engenharia do Instituto Tecnológico de Aeronáutica (ITA). Como outra forma de abordar o problema, foi feita uma simulação do fenômeno por meio de um programa desenvolvido por eles utilizando a biblioteca vpython da linguagem de programação python.

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.

Motivados por esses fatores, propomos para este trabalho dois objetivos: descrever e caracterizar o python e a sua biblioteca gráfica vpython como ferramentas adequadas ao ensino da Física, apresentando suas potencialidades e limitações; e desenvolver simuladores computacionais a partir dessas, sinalizando parâmetros para o ensino de Física de modo a fundamentar propostas educativas. Com vista a esses objetivos, são discutidas possíveis contribuições do trabalho na inserção do/a professor/a no contexto científico-tecnológico necessário para se gerar o entendimento dessa linguagem de programação como possibilidade de mediação do conhecimento na abordagem da Física.

IV. Resultados e discussões

Em um primeiro momento, buscamos retratar o python, em conjunto com as suas bibliotecas vpython e tkinter, de modo a demonstrar a lógica de programação e as potencialidades da construção de simuladores computacionais por meio dessa linguagem. Em um segundo momento, utilizando esses recursos, apresentamos simuladores computacionais que integram o laboratório virtual de mecânica, salientando seus fatores potencializadores e limitantes no ensino da Física, perpassando por sua construção.

Para possibilitar uma perspectiva ainda mais ampla de desenvolvimento desses simuladores, apresentamos tutoriais que abordam a instalação das ferramentas computacionais necessárias, noções básicas de python e vpython e a lógica utilizada para construção dos simuladores de forma geral e específica. Esses tutoriais remontam um laboratório virtual⁴ de Física, que contém, além desses materiais, simuladores produzidos com o python e vpython para a abordagem de fenômenos explorados na mecânica. Para ter acesso aos simuladores e tutoriais, basta acessar o laboratório no link <<https://www.griquer.com/laborat%C3%B3rio-virtual>>.

No que concerne à disponibilidade dos códigos apresentados e à instalação das ferramentas necessárias ao desenvolvimento dos simuladores, é possível encontrá-los no link disponibilizado no site do laboratório virtual, assim como no apêndice. Neles, é explicitada a lógica de programação responsável pela produção de simuladores embasados em fenômenos físicos. Além disso, outras informações referentes ao python podem ser encontradas em seu site oficial⁵.

IV.1 Características gerais da construção dos simuladores

Entre as diversas linguagens de programação existentes, o python se destaca por ter uma sintaxe simples, um código aberto ao usuário, ser orientada a objeto e, apesar de robusta, poder se tornar, com facilidade, uma linguagem de alto nível se usada em conjunto com módulos e bibliotecas instaláveis (BORCHERDS, 2007). Por essas razões, ela se caracteriza como uma linguagem acessível no processo de ensino-aprendizagem de lógica de programação para iniciantes, visto que conceitos essenciais como sequência, condição e repetição são mais simples de serem trabalhados com ela (BORCHERDS, 2007). Referente ao levantamento realizado, destaca-se, por exemplo, que seis trabalhos tiveram contato direto com a linguagem de programação python (trabalho 1, 2, 3, 5, 6, 7), enquanto que apenas um utilizou um *software* construído em python (trabalho 4).

⁴ Ambientes de simulação de experimentos reais relacionados a fenômenos físicos, sendo acessado por meio das TIC. Nesse caso, o laboratório se trata de um conjunto de simuladores computacionais desenvolvidos pelos autores deste trabalho com o python e vpython.

⁵ Ver: <<https://www.python.org/>>

Para além disso, o python se destaca por possuir uma vasta gama de módulos e bibliotecas embasadas em um conjunto de expressões físicas e matemáticas capazes de proporcionar uma maior fundamentação no desenvolvimento de algoritmos associados a fenômenos da natureza, o que a torna uma linguagem computacional adequada para a área de Física (BORCHERDS, 2007). Dentre essas bibliotecas, existem aquelas focadas no desenvolvimento gráfico, seja de interfaces de usuário (tkinter), seja na modelagem de fenômenos (vpython), capazes de proporcionar a construção de simuladores computacionais de forma prática. Para utilizá-las no python, é preciso, após terem sido instaladas, apenas importá-las no algoritmo.

Em se tratando do vpython, ele se caracteriza por ser uma biblioteca gráfica do python capaz de proporcionar a inserção de figuras geométricas tridimensionais – elipsóides, paralelepípedos, pirâmides, trapézios, entre outros – em sua interface, permitindo que essas possam ser manipuladas de acordo com mudanças em seus parâmetros (Fig. 1). Assim, a alteração em seus valores pode modelar algumas situações que envolvem o movimento, uma vez que eles estão associados à posição, massa, tamanho e, até mesmo, velocidade. Nesse sentido, aplicando os modelos físicos correspondentes na construção dos seus algoritmos, é possível criar simuladores para tipos específicos de movimento, como os amplamente abordados na disciplina de Física. No levantamento dos trabalhos, é visto que o vpython tem sido pouco utilizado, tendo apenas um trabalho que explicita o seu uso na construção dos simuladores (trabalho 7).

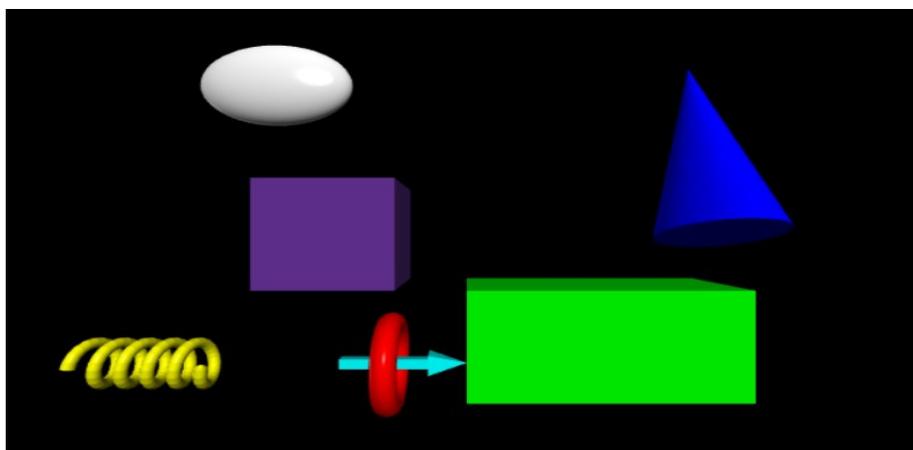


Fig. 1 – Figuras criadas a partir do python utilizando o módulo vpython.

Na construção dos simuladores, é preciso se atentar à interface do vpython em que as figuras são inseridas: um espaço cartesiano tridimensional com origem em seu centro. Para facilitar a inserção desses objetos na interface, muitos parâmetros relacionados a eles recebem valores do tipo vetorial nas três coordenadas espaciais, a exemplo de sua posição na interface e de suas dimensões. Outrossim, apesar de não serem necessários para uma criação inicial desses objetos, outros parâmetros, como cor, orientação do seu eixo e opacidade, são

interessantes para ilustrar melhor algumas simulações, sendo, em alguns casos, essenciais para uma melhor visualização dessas.

No que se refere ao tkinter, apesar de não ser necessariamente uma biblioteca fundamental para o desenvolvimento dos simuladores, ganha destaque como um *framework* capaz de proporcionar o desenvolvimento de interfaces gráficas de usuário (GUI – *Graphical User Interface*) no python de forma prática, uma vez que é de fácil usabilidade e contém múltiplos recursos. Logo, é possível criar janelas capazes de receber os dados inseridos pelo/as usuário/as, tornando os simuladores mais atraentes para os/as estudantes, já que estão adaptados a esse tipo de configuração proporcionada pelos programas atuais. Cabe destacar que, no levantamento realizado, apenas um (trabalho 2) deixou explícito o uso do tkinter na construção dos simuladores.

Com essas noções de programação, é possível que professores/as e estudantes de Física, seja da educação básica, seja da educação superior, desenvolvam simuladores similares aos apresentados no próximo tópico, ou alguns para além dos que já foram criados como, por exemplo, os apresentados no laboratório virtual. Ademais, eles também podem modificar o código desses simuladores, adequando-os às suas aulas de acordo com as suas necessidades. Com isso, é possível gerar uma maior inserção desses/as professores/as e estudantes no contexto da programação.

IV.2 Simulador Computacional: Lançamento Oblíquo

A fim de ilustrar as potencialidades do uso de simuladores construídos com o python e vpython em abordagens de conteúdos da Física, trazemos uma simulação do lançamento oblíquo, visto que esse é considerado um movimento de difícil abstração por parte dos/as estudantes, seja por envolver mais variáveis que outros movimentos da cinemática, seja por ser bidimensional. Nesse sentido, os/as estudantes podem observar o comportamento do fenômeno, compreendendo como valores, a exemplo do ângulo de lançamento e velocidade inicial, influem no movimento (Fig. 2). Outrossim, há também a possibilidade de se trabalhar com a independência do movimento nas componentes x e y, uma vez que a partícula se comporta como um movimento retilíneo uniforme na componente x (1) e como um movimento retilíneo uniformemente variado na componente y (2).

$$x = x_0 + v_0 \cos \theta t \quad (1)$$

$$y = y_0 + v_0 \sin \theta t + \frac{1}{2} g t^2 \quad (2)$$

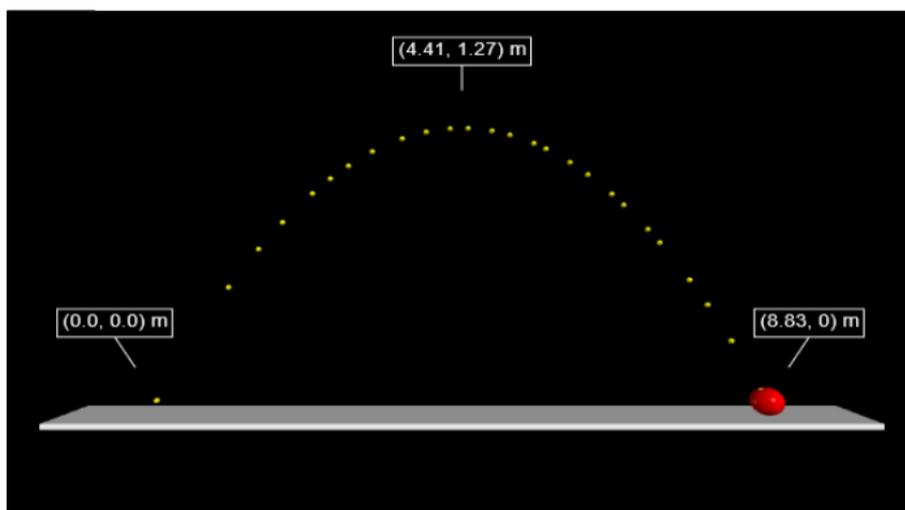
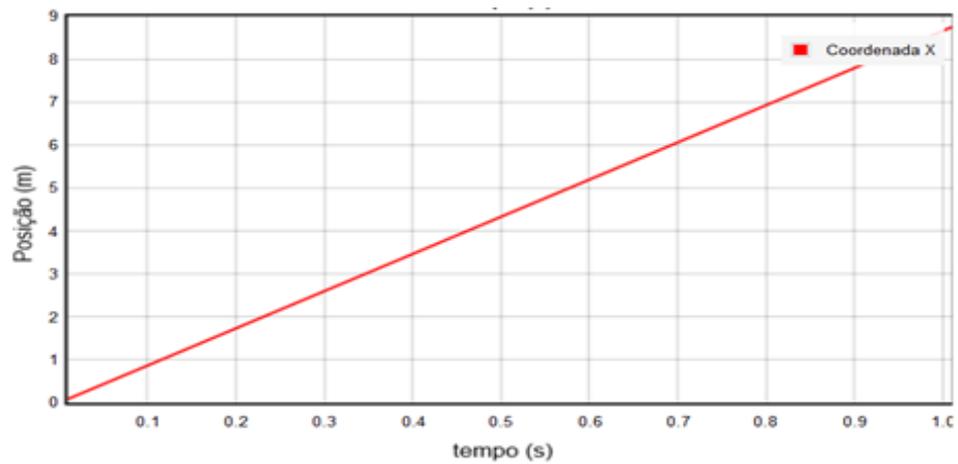


Fig. 2 – Simulador de lançamento oblíquo em python utilizando uma esfera, um paralelepípedo e três legendas.

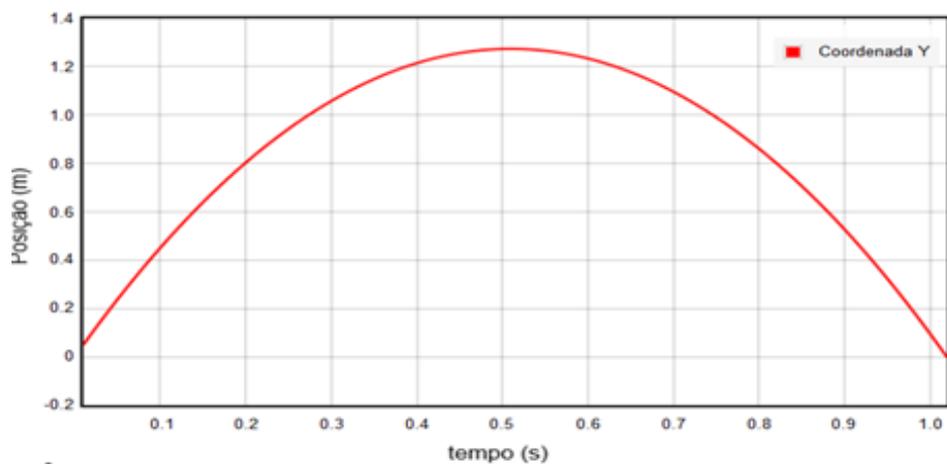
Na simulação (Fig. 2), parâmetros como a posição inicial, ponto de máximo e a posição final foram inseridos na interface de modo a proporcionar aos/às estudantes uma maior facilidade na abstração do movimento em suas principais partes, além de serem parâmetros fundamentais para a caracterização do seu estado de movimento. Ademais, para uma percepção ainda mais dinâmica do problema, são gerados gráficos em tempo real da posição da partícula na coordenada x versus o tempo (Fig. 3.a), da posição da partícula na coordenada y versus o tempo (Fig. 3.b) e da velocidade da partícula na coordenada y versus o tempo (Fig. 3.c), visto que o vpython proporciona esse tipo de construção. Com esses recursos, o/a estudante consegue visualizar o fenômeno em tempo real de uma forma relativamente completa, o que o auxilia a compreender a relação entre o movimento e a equação responsável por caracterizá-lo.

Para criar a simulação, foram utilizados apenas dois objetos – uma esfera⁶ para representar a partícula e um paralelepípedo para representar o solo – e três legendas – posição inicial, ponto de altura máxima e posição final. Para solicitar ao/à usuário/a os elementos necessários à reprodução do fenômeno, foi criada uma GUI utilizando o tkinter (Fig. 4). Com a captura desses dados (ângulo de lançamento, velocidade inicial, posição inicial) por meio do evento associado ao botão “simular”, é possível reproduzir o lançamento oblíquo a partir da mudança de posição da esfera em uma estrutura de repetição, tendo o tempo como contador, que varia de 0 até o tempo total de queda. Logo, essa mudança é definida pela equação da posição (1) no eixo x , e pela equação da posição (2) no eixo y , sendo evidenciada pela trajetória da partícula (ver tutorial de construção do simulador de lançamento).

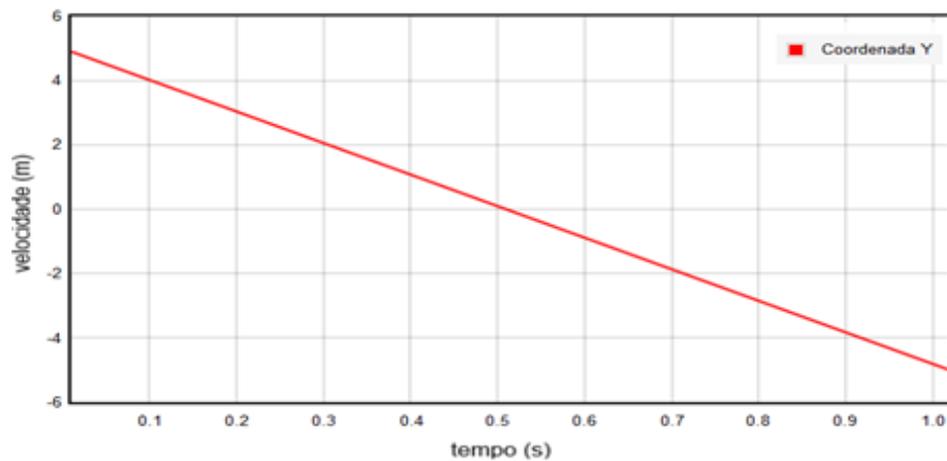
⁶ Destaca-se que a escolha da forma geométrica esférica para representar a partícula se deu por ser de uso convencional em abordagens da Física, tendo suas dimensões desprezíveis na simulação.



(a)



(b)



(c)

Fig. 3 – Gráficos referentes ao lançamento oblíquo. (a) Gráfico da posição da partícula na componente x em função do tempo. (b) Gráfico da posição da partícula na componente y em função do tempo. (c) Gráfico da velocidade da partícula na componente y em função do tempo.

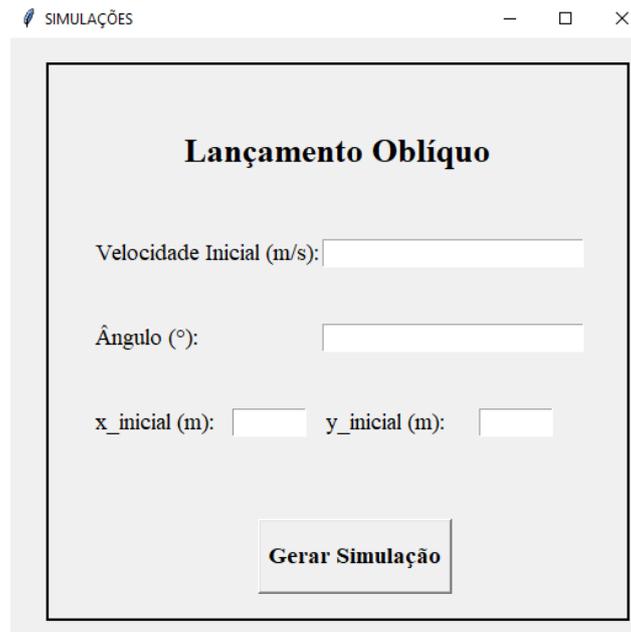


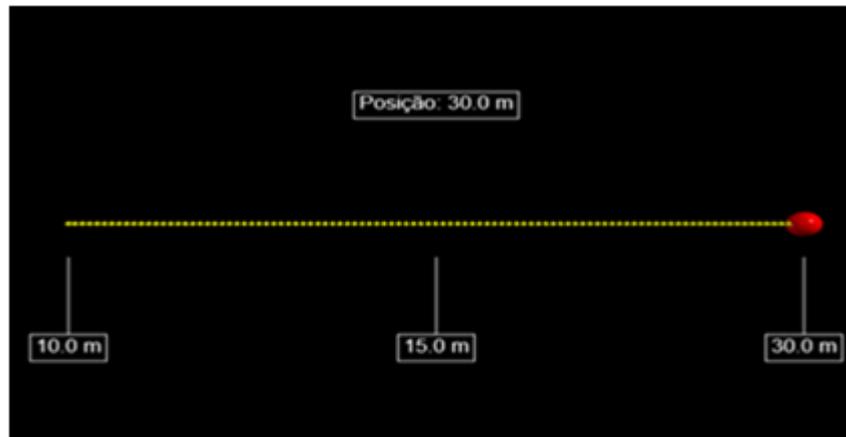
Fig. 4 – Interface gráfica de usuário (GUI) do simulador de lançamento oblíquo.

Com vista a essas características, o simulador tem potencial de trazer abordagens interativas e dinâmicas para as aulas de Física, inserindo os/as estudantes em um contexto científico-tecnológico ao qual a maioria faz parte. Embora percam o contato físico do laboratório, em contrapartida, ele permite uma maior aproximação subjacente às equações da mecânica. Nesse sentido, o/a professor/a deve estar atento às especificidades metodológicas das abordagens experimentais, evitando comparações reducionistas, tendo em vista que os simuladores computacionais não devem ser tratados como substitutos aos experimentos realizados com aparatos físicos, já que ambos têm finalidades diferentes. Assim, atentando-se a esses aspectos, os/as professores/as podem, em conjunto com os/as estudantes, desenvolver outras simulações de forma autônoma com a mesma lógica de construção do simulador apresentado a partir da apropriação de conhecimentos científicos envolvidos nas abordagens dos conteúdos da Física, implicando em uma contextualização ainda maior dos fenômenos físicos vistos nas aulas.

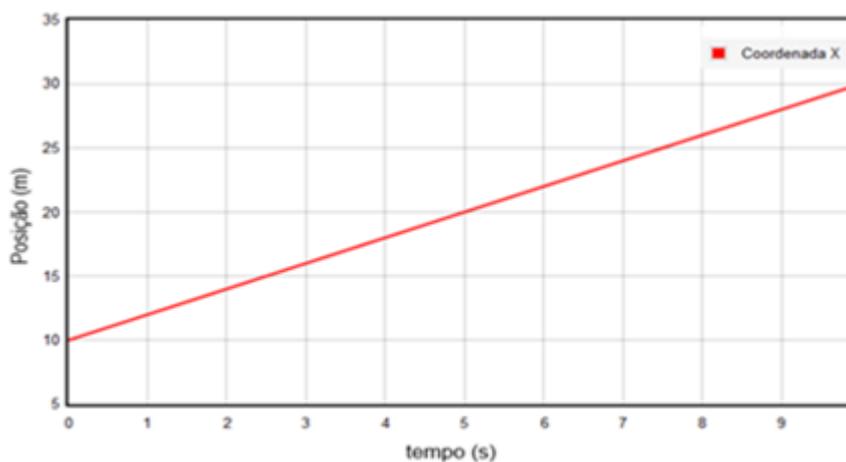
IV.3 Simuladores Computacionais: outras possibilidades para o ensino de Física

Para demonstrar as várias possibilidades de desenvolvimento de simulações na área da Física com o uso do python e vpython, destacamos outros dois simuladores, caracterizando o seu estado de movimento e suas especificidades. O primeiro se trata de uma partícula que realiza o movimento retilíneo uniforme (Fig. 5.a) a partir da posição inicial, variação do tempo, velocidade da partícula (3) inseridos pelo/a usuário/a (ver tutorial de construção do simulador de MRU). Com ele, alguns aspectos podem ser evidenciados durante a aula, tais como a velocidade constante ao longo do movimento (intervalos iguais entre os pontos da

trajetória) e a diferença entre o movimento progressivo e retrógrado, já que a partícula assume sentidos diferentes de acordo com o valor da velocidade (positiva ou negativa). Além disso, também é criado o gráfico em tempo real da posição da partícula em função do tempo, permitindo uma melhor ilustração desses conceitos (Fig. 5.b).



(a)



(b)

Fig. 5 – Simulador do movimento retilíneo uniforme. (a) Simulador de movimento retilíneo uniforme utilizando apenas uma esfera e três legendas. (b) Gráfico da posição da partícula em função do tempo.

$$x = x_0 + vt \quad (3)$$

Além do apresentado, destacamos outro simulador que representa um sistema de dois blocos conectados por um fio inextensível, que passa por uma polia fixa, tendo um bloco suspenso pelo fio (bloco azul) e outro apoiado em cima de uma parede (bloco vermelho) (Fig. 6). Com os dados da altura em relação ao solo do bloco suspenso e a massa de ambos os blocos, é possível achar a aceleração do sistema (4) e, conseqüentemente, o tempo de queda do bloco suspenso (5). Com esse simulador, o conjunto de forças no sistema e as suas interações podem ser visualizados por meio do resultado obtido a partir de dados inseridos

pelo/a próprio/a usuário/a, ficando em maior evidência através dos vetores que agem nos blocos, indicando as direções das forças que atuam nas massas, possibilitando uma melhor assimilação dos conceitos da Física presentes no movimento.

$$a = \frac{m_b - m_a}{m_a + m_b} g \quad (4)$$

$$t = \sqrt{\frac{2h}{a}} \quad (5)$$

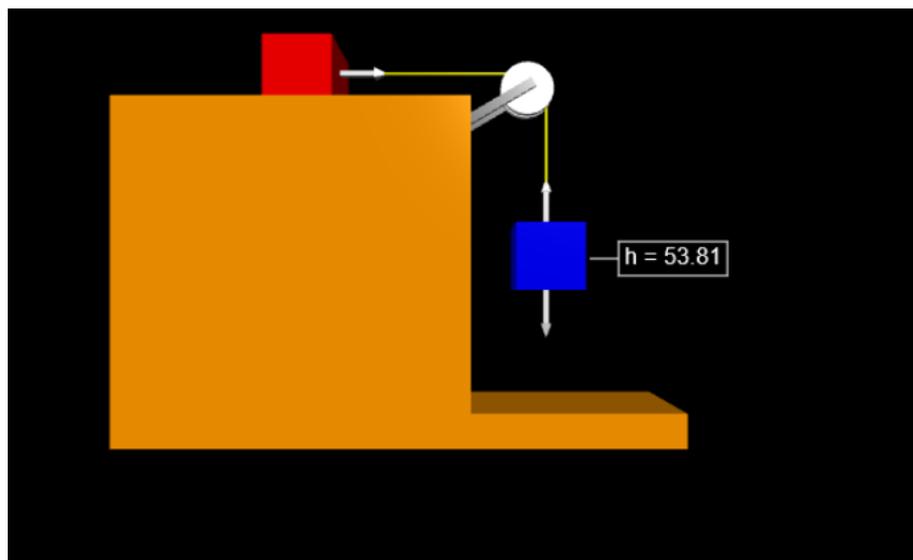


Fig. 6 – Simulador de sistema de blocos com uma polia fixa utilizando cilindros, paralelepípedos, legendas e vetores.

Com vista aos simuladores apresentados, as possibilidades de seu desenvolvimento a partir do python e vpython são inúmeras, uma vez que, a partir da manipulação das formas geométricas tridimensionais desejadas, são gerados diversos simuladores para os fenômenos abordados na Física. Por fim, outros simuladores referentes ao conteúdo de mecânica podem ser encontrados no laboratório virtual, que foi construído de modo a desenvolver possibilidades, para além das tradicionais, de se trabalhar com esses fenômenos abordados na Física, em especial nesse novo contexto, que nos possibilitou pensar em alternativas frente ao acesso presencial dos laboratórios com aparatos físicos.

V. Considerações finais

A construção de simuladores com o uso do python, juntamente com a sua biblioteca gráfica vpython, demonstra ser um recurso educacional com potencial para o ensino da Física, como evidenciado ao longo do trabalho. Dentre as razões para tal afirmativa, destaca-se a facilidade em trabalhar com o python na construção de algoritmos, a possibilidade de criar

figuras geométricas tridimensionais de uso dinâmico com o vpython e, sobretudo, o desenvolvimento de simuladores que potencializam, de forma significativa, o processo de ensino-aprendizagem devido à sua interatividade proporcionada e uma maior liberdade para a criação dos simuladores. Com eles, os/as professores/as podem aplicar metodologias e recursos de ensino para além das tradicionais, enquanto que os/as estudantes podem ter contato com recursos tecnológicos e digitais mais próximos da realidade da maioria.

Entretanto, ainda que essas potencialidades sejam evidenciadas, o número de trabalhos que abordam o desenvolvimento de simuladores computacionais com esses recursos tem se mostrado baixo. Destaca-se ainda que esse pequeno número de trabalhos têm uma ênfase quase exclusiva na Educação Superior. No levantamento realizado, apenas dois, dos sete trabalhos encontrados, são voltados à Educação Básica. E, dentre esses trabalhos, nenhum discutia questões referentes à cinemática.

Além dessas questões, o levantamento demonstra a pouca preocupação dos trabalhos em descrever potencialidades e limitações trazidas pela linguagem de programação python, que é justamente a responsável pela transposição do fenômeno da realidade ao ambiente virtual (GIORDAN, 2005). Apesar de compreendermos os avanços das atividades de uso de simulações já desenvolvidas (i) e a construção de simuladores por meio de ambientes pré-programados (ii) no ensino de Física, assumimos neste trabalho a importância de um olhar integrador entre a Física e a programação, seja por exigir do/a usuário/a um conhecimento aprofundado dos conceitos da Física para a construção dos simuladores se comparado a outras atividades, seja pelo fato de, em tempos de “sociedade da informação”, a linguagem de programação ter se mostrado fundamental em atividades de diversas naturezas.

Contudo, apesar de os alcances de utilização do python e vpython se mostrarem significativos para o ensino da Física, pontuamos a falta de conhecimento em programação como uma limitação na efetivação de propostas que utilizem essas ferramentas. No entanto, entendemos que a proposta apresentada sinaliza uma possibilidade de inserção dos professores nesse contexto, uma vez que simples ajustes nos códigos já disponibilizados (no laboratório e no apêndice) e consultas rápidas aos tutoriais já proporcionam um gradual contato com a programação.

Portanto, com noções de programação em python, as possibilidades de produção de simulações computacionais são diversas. Logo, o python e vpython demonstram grandes potencialidades de uso em metodologias que envolvam os simuladores para o ensino da Física. No cenário atual, por exemplo, com a pandemia acometida pelo novo coronavírus (SARS-CoV-2), esses simuladores ganham grande destaque devido à possibilidade de uma maior diversidade de recursos a serem utilizados nas aulas remotas e híbridas, bem como pelo potencial de interação com simulações computacionais nesse contexto. E ainda, com o uso dessas simulações atenuado pelo atual período, as vantagens trazidas por elas têm grande potencial de serem explicitadas durante o processo de ensino-aprendizagem, acarretando uma

maior aplicabilidade desses recursos no ensino da Física em momentos posteriores à pandemia.

Referências bibliográficas

AFONSO, C. A. Internet no Brasil: alguns dos desafios a enfrentar. **Informática Pública**, v. 4, n. 2, p. 169-184, 2002.

BARANAUSKAS, M. C. C.; VALENTE, J. A. Editorial. **Tecnologias, Sociedade e Conhecimento**, v. 1, n. 1, p. 1-5, 2013.

BORCHERDS, P. H. Python: a language for computational physics. **Computer Physics Communications**, p. 199-201, fev. 2007.

CENTENARO, F.; SAUERWEIN, I.; SAUERWEIN, R.; PASTORIO, D.; ALVES, J. As TIC no ensino de física: uma proposta de atividade didática com o uso de uma simulação computacional acerca do movimento de um pêndulo simples. **Revista de Enseñanza de la Física**, v. 29, n. extra, p. 353 -359, nov. 2017.

CENTRO REGIONAL DE ESTUDOS PARA O DESENVOLVIMENTO DA SOCIEDADE DA INFORMAÇÃO - CETIC. *In: TIC Domicílios - 2018*. [S. l.], 2018. <<https://cetic.br/pt/tics/domicilios/2019/domicilios/A1/>>. Acesso em: 3 mai. 2021.

CUZINATTO, R.; MORAIS, E. Software MUFCosm como ferramenta de estudo dos modelos da cosmologia padrão. **Revista Brasileira de Ensino de Física**, [S. l.], v. 36, p. 1-7, 6 fev. 2014.

DIONÍSIO, G.; SPALDING, L. Visualização da forma de onda e conteúdo harmônico da corrente elétrica alternada em eletrodomésticos. **Revista Brasileira de Ensino de Física**, v. 39, n. 1, p. 1-15, 6 ago. 2016.

FEITOSA, E. TDIC Para Ensinar e Aprender Física. In: XII Congresso Nacional de Educação, 2015, Curitiba. **Anais...** Paraná: Pontifícia Universidade Católica do Paraná, 2015, p. 1-14.

FERNANDES, G.; RODRIGUES, A.; FERREIRA, C. Módulos temáticos virtuais: Uma proposta pedagógica para o ensino de ciências e o uso das TICs. **Caderno Brasileiro de Ensino de Física**, v. 32, n. 3, p. 934-962, out. 2015.

FIOLHAIS, C.; TRINDADE, J. Física no computador: o computador como ferramenta no ensino e na aprendizagem das Ciências Física. **Revista Brasileira de Ensino de Física**, [S. l.], v. 25, p. 259-272, 3 set. 2003.

FREITAS-LEMOS, P.; VILELA, D.; GUARNIERI, M.; PRADO, R.; MEDEIROS, T.; GEERMANO, J. Análise do movimento de um ponto de luz sobre um plano inclinado. **Revista Brasileira de Ensino de Física**, v. 41, n. 1, p. 1-12, 2019.

GIORDAN, M. O Computador na Educação em Ciências: Breve Revisão Crítica acerca de Algumas Formas de Utilização. **Ciência & Educação**, v. 11, n. 2, p. 279-304, 2005.

HEINECK, R.; VALIATI, E. R. A.; ROSA, C. T. W. Software Educativo no Ensino de Física: análise quantitativa e qualitativa. **Revista Iberoamericana de Educación**, v. 42, n. 6, p. 1-12, 2007.

JAIME, D.; FILHO, L.; LEONEL, A. Ensino de física em tempos pandêmicos: um olhar para a fala dos estudantes de uma escola da rede federal. **Revista Tecnologias Educacionais em Rede**, v. 2, n. 2, p. 1-20, 2021.

JOSÉ, W.; ANGOTTI, J.; BASTOS, F. Ensino de Física por meio de questões do PISA associadas a Temas Estruturadores e Conceitos Unificadores. **Caderno Brasileiro de Ensino de Física**, v. 33, n. 2, p. 333-354, set. de 2016.

LAWSON, R. A.; MCDERMOTT, L. C. Student understanding of the work- energy and impulse- momentum theorems. **American Journal of Physics**, v. 55, n. 9, p 811–817, 1987.

LAURINDO, F. J. B; CARVALHO M. M. Outsourcing e geração de valor na indústria dos computadores pessoais: estudo de múltiplos casos. **Gestão e Produção**, v. 1, n. 3, p. 363-377, 2003.

MACEDO, G.; JUNIOR, A. Aplicação do Problema Restrito de Três Corpos no estudo do movimento de astros do sistema solar. **Revista Brasileira de Ensino de Física**, v. 40, n. 4, p. 1-8, 2018.

MACÊDO, J. PEDROSO, L.; VOELZKE, M.; ARAÚJO, M. Levantamento das abordagens e tendências dos trabalhos sobre as tecnologias da Informação e comunicação apresentados no XIX Simpósio Nacional de Ensino de Física. **Caderno Brasileiro de Ensino de Física**, v. 31, n. 1, p. 167-197, 2013.

MARTINS, N.; CORDEIRO, R.; LOUREIRO, M. Um programa em python para experimentos de Física automatizados com o Arduino. In: SIMPÓSIO NACIONAL DE ENSINO DE FÍSICA, XXIII, 2019, Salvador. **Anais...** São Paulo: Sociedade Brasileira de Física, 2019, p. 1-8.

MEDEIROS, A.; MEDEIROS, C. Possibilidades e limitações das simulações computacionais no ensino de Física. **Revista Brasileira de Ensino de Física**, v. 24, p. 77-86, 2 jun. 2002.

MOREIRA, M. A. Uma análise crítica do ensino de Física. **Estudos avançados**, v. 32, n. 94, p. 73-80, 2018.

MOREIRA, M. A. Desafios no ensino de física. **Revista Brasileira de Ensino de Física**, v. 43, n. 1, p. 1-8, 2021.

PASTORIO, D. **Processos avaliativos reflexivos integrados a tarefas contínuas no âmbito do ensino superior em Física**. 2018. 223 f. Tese (Doutorado em Educação em Ciências: Química da Vida e Saúde) - Universidade Federal de Santa Maria, Santa Maria, RS.

PARIZOTTO, G.; SOARES, M. Noções de Programação Estruturada em Python no ensino de Física: cultura lúdica e dificuldades de aplicação. In: SIMPÓSIO NACIONAL DE ENSINO DE FÍSICA, XXII, 2017, São Carlos. **Anais...** São Paulo: Sociedade Brasileira de Física, 2017, p. 1-8.

PYTHON SOFTWARE FOUNDATION. **Python**. [S. l.], 2001. Disponível em: <<https://www.python.org/>>. Acesso em: 1 jun. 2021.

RANGEL, F.; SANTOS, L.; RIBEIRO, C. Ensino de Física mediado por tecnologias digitais de informação e comunicação e a literacia científica. **Caderno Brasileiro de Ensino de Física**, v. 29, p. 651-677, out. de 2012.

ROSA, C.; SOSO, F.; DARROZ, L. Tecnologias Digitais de Informação e Comunicação na voz de professores de Física do ensino médio. **Revista Educere et Educare**, v. 13, p. 1-24, nov. de 2018.

SANTOS, P.; BOTTENTUIT, A.; JORENTE, M.; GROSSI, A. INFOINCLUSÃO DIGITAL: uma experiência na cidade de Marília (SP). **Revista Brasileira de Biblioteconomia e Documentação**, v. 3, n. 2, p. 129-140, 2007.

SILVA, M.; PEDERSEN, F.; CARVALHO-NETO, J. Projeto Estelar: uma ferramenta tecnológica para o ensino de astronomia e observação do céu a olho nu. In: ENCONTRO DE PESQUISA E ENSINO DE FÍSICA, XVIII, 2020, Florianópolis. **Anais...** São Paulo: Sociedade Brasileira de Ensino de Física, 2020, p. 1-8.

WESENDONK, F. **O uso da experimentação como recurso didático no desenvolvimento do trabalho de professores de física do ensino médio.** 2015. Dissertação (Mestrado em Educação para a Ciência) - Universidade Estadual Paulista Júlio de Mesquita Filho, Bauru, São Paulo.

Apêndices

Box 1: Código referente ao movimento retilíneo uniforme

```
from vpython import *
from tkinter import *
graf_pos = graph(xtitle='tempo (s)', ytitle='Posição (m)')
curva_pos = gcurve(color=color.red, label = 'Coordenada X')
def ajuda():
    newWindow = Toplevel(janela)
    newWindow.geometry('520x480+100+150')
    newWindow.title('AJUDA')
    newWindow['pady']=20
    newWindow['padx']=30
    lab_title2 = Label(newWindow, text='Movimento Retilíneo Uniforme',
        font='times 20 bold', height=3)
    lab_title2.grid(row=0, column=0)
    a = 'O movimento retilíneo uniforme é caracterizado por um corpo se
movendo
    em
    a = a + 'velocidade constante durante toda a sua trajetória
(retilínea),
    resultando em uma
    a = a + 'velocidade instantânea igual à velocidade média. Com a
posição
    inicial, velocidade inicial e
    a = a + 'intervalo de tempo (tempo inicial e tempo final), é possível
se
    ter uma descrição exata do movimento.
    a = a + '\n\nNa simulação é possível visualizar a trajetória do corpo
formada por pontos que
    a = a + 'aparecem em intervalos de tempos iguais. Nela é possível
verificar
    que o espaço de um ponto
    a = a + 'a outro é simétrico, o que indica que a variação da posição
do
    corpo em relação ao tempo é'
```

```

a = a + ' constante para toda trajetória. (Em alguns casos, no início
da_
,--simulação a trajetória pode conter '
a = a + ' algumas incoerências devido a um atraso na inicialização da_
,--simulação). Outrossim, é possível '
a = a + ' visualizar a origem, o final e o ponto médio da trajetória,
assim_
,--como a posição em que o corpo '
a = a + ' se encontra em instantes diferentes, possibilitando uma
melhor_
,--compreensão das escalas '
a = a + ' utilizada em cada caso. '
lab_desc = Label(newWindow, font='times 12 ', text=a, wraplength=450)
lab_desc.grid(row=1, column=0)

#evento associado ao botão Gerar Simulação
def bt_clic():
    try:
        x_inicial=float(ed_i_pos.get())
        v=float(ed_i_vel.get())
        t_inicial=float(ed_ti.get())
        t_final=float(ed_tf.get())
        #Simulação
        if (v<0):
            p = 10
        else:p=-10
        #Perceba que vamos inserir os parâmetros da trajetória própria_
        ,--esfera
        delt_t = t_final - t_inicial #tempo de execução da simulação_
        ,--(variação do tempo)
        esfera = sphere(pos = vec(p, 0, 0), radius = 0.4, color =
color.
        ,--red, opacity=1, make_trail=True, interval=10,
trail_type="points",_
        ,--trail_color =color.yellow)
        texto_x_inicial = label(pos=vec(p, 0, 0), text= str(x_inicial)
+ "_
        ,--m", xoffset=0, yoffset=-100, space=30, height=16, border=4)
#indica o
        texto_x_medio = label(pos=vec(0, 0, 0), text=_
        ,--str((v*delt_t+x_inicial)/2)+ " m", xoffset=0, yoffset=-100,
space=30,_
        ,--height=16, border=4, font='sans') #indica o ponto médio
        texto_x_final = label(pos=vec(p*-1, 0, 0), text=_
        ,--str(v*delt_t+x_inicial)+ " m", xoffset=0, yoffset=-100,
space=30,height= 16,
        ,--border=4, font='sans') #indica o ponto final

```

```

        texto_posicao = label(pos=vec(0, 4, 0), text= "Posição: " +
        ,→str(x_inicial)+ " m", space=30, height=16, border=4,
font=' sans' )
        t = 0
        e = abs(v*delt_t)/20 #fator de controle de escala
        janela.destroy()
        while (t < delt_t):
            rate (100)
            esfera.pos = vec(p + (v*t)/e, 0, 0)#Perceba a inserção do fator
            ,→de escala apenas na variação da posição
            curva_pos.plot(pos=(t_inicial + t, x_inicial + v*t)) #plotagem
            ,→do gráfico
            texto_posicao.text = 'Posição: ' + str(round (x_inicial + v*t,
            ,→2))+ ' m'
            t+=0.01
    except:
        lb1['text']='Valores Informados Inválidos!!!'

#O código abaixo se refere exclusivamente à criação de janelas do TKinter
janela = Tk()
janela.geometry(' 530x500+400+150')
janela.title('SIMULAÇÕES')
janela['pady']=20
janela['padx']=30
container = Frame(janela, highlightbackground='black', highlightthickness=2)
container['pady']=20
container['padx']=35
container.grid(row=0, column=0)
lab_title = Label(container, text='Movimento Retilíneo Uniforme', font='times
,→20 bold', height=3)
lab_i_pos = Label(container, text='Posição Inicial (m):', font='times 14 ',
,→height=3)
lab_i_vel = Label(container, text='Velocidade Inicial (m/s):', font='times 14
,→', height=3)
lab_ti = Label(container, text='t_inicial(s):', font='times 14 ', height=3)
lab_tf = Label(container, text='t_final (s):', font='times 14 ', height=3)
lab_title.grid(row=0, column=0, columnspan=4)
lab_i_pos.grid(row=1, column=0, columnspan=2, sticky=W)
lab_i_vel.grid(row=2, column=0, columnspan=2, sticky=W)
lab_ti.grid(row=3, column=0, sticky=W)
lab_tf.grid(row=3, column=2, sticky=W)
ed_i_pos = Entry(container, width=26,font='times 12')
ed_i_vel = Entry(container, width=26,font='times 12')

```

```

ed_ti = Entry(container, width=7, font='times 12')
ed_tf = Entry(container, width=7, font='times 12')
ed_i_pos.grid(row=1, column=2, columnspan=2, sticky=W)
ed_i_vel.grid(row=2, column=2, columnspan=2, sticky=W)
ed_ti.grid(row=3, column=1, sticky=W)
ed_tf.grid(row=3, column=3, sticky=W)
lbl = Label(container, text='', height=2, font='times 12 italic')
lbl.grid(row=4, column=1, columnspan=2)
bt = Button(container, text = 'Gerar Simulação', font='times 14 bold',
pady=12,
,--command=bt_clic)
bt.grid(row=5, column=1, columnspan=2, sticky=S)
buttonExample = Button(container, text='Ajuda', font='times 12 bold',
,--command=ajuda) #Botão de ajuda
buttonExample.grid(row=5, column=0, columnspan=2, sticky=W)
janela.mainloop()

```

Box 2: Código referente ao lançamento oblíquo

```

from tkinter import *
from vpython import *
import math
#posição da partícula na coordenada x versus o tempo
graf_pos_x = graph(xtitle="tempo (s)", ytitle="Posição (m)")
curva_pos_x = gcurve(color=color.red, label = "Coordenada X")
#posição da partícula na coordenada y versus o tempo
graf_pos_y = graph(xtitle="tempo (s)", ytitle="Posição (m)")
curva_pos_y = gcurve(color=color.red, label = "Coordenada Y")
#velocidade da partícula na coordenada y versus o tempo
graf_vel_y = graph(xtitle="tempo (s)", ytitle="velocidade (m/s)")
curva_vel_y = gcurve(color=color.red, label = "Coordenada Y")

def ajuda():
    newWindow = Toplevel(janela)
    newWindow.geometry('520x480+100+150')
    newWindow.title('AJUDA')
    newWindow['pady']=20
    newWindow['padx']=30
    lab_title2 = Label(newWindow, text='Lançamento Oblíquo', font='times
20,
,--bold', height=3)
    lab_title2.grid(row=0, column=0)
    a = '0 lançamento oblíquo se caracteriza por ser um movimento
curvilíneo,
,--em que um corpo é '
    a = a + 'lançado a partir de um ângulo com uma velocidade inicial,
,--realizando uma trajetória '

```

```

a = a + 'parabólica. Nele, a velocidade na componente x é constante,
,-enquanto que na componente y '
a = a + 'é acelerado devido justamente à aceleração da gravidade. '
a = a + '\n\nNa simulação é possível visualizar a trajetória do corpo,
,-formada por pontos que '
a = a + 'aparecem em intervalos de tempos iguais - como dito, uma
,-trajetória parabólica. '
na
a = a + 'É possível verificar também que o espaço de um ponto a outro
,-componente x é igual e '
do
a = a + 'na componente y varia, o que indica que a variação da posição
,-corpo em relação ao tempo '
a = a + 'é constante em x e acelerado em y. (Em alguns casos, no
início da
,-simulação a trajetória '
a = a + 'pode conter algumas incoerências devido a um atraso na
,-inicialização da simulação). '
a = a + 'Outrossim, ao longo da trajetória, é marcado o posição
inicial,
,-posição de máximo e posição final. '
lab_desc = Label(newWindow, font='times 12 ', text=a, wraplength=450)
lab_desc.grid(row=1, column=0)

def bt_clic():
    try:
        i_vel=float(ed_i_vel.get())
        tet=float(ed_i_tet.get())
        x_pos=float(ed_i_x.get())
        y_pos=float(ed_i_y.get())
        g = 9.81 #valor da gravidade
        t = math.sin(math.radians(tet))*(i_vel/g) #tempo de subida
        y_max = y_pos + ((i_vel*math.sin(math.radians(tet)))**2)/(2*g)
        #altura
        ,-máxima
        x_y_max = x_pos + i_vel*math.cos(math.radians(tet))*t #correspondente
        ,-em x da altura máxima
        x_vel = i_vel*math.cos(math.radians(tet)) #Velocidade em x
        if (y_pos > 0):
            t += (2*y_max/g)**(1/2) #tempo de subida + tempo de descida (tempo
            ,-total)
        else: t = t*2 #tempo de subida + tempo de descida (tempo total)

        if (t <= 0.5): #Variável de controle do tempo para tempos muito curtos
            c = 5

```

```

elif (t <= 1):
    c = 10
elif (t <= 2):
    c = 20
elif (t <= 3):
    c = 30
elif (t <= 4):
    c = 40
elif (t <= 5):
    c = 50
else:
    c = 100

delt_sx = i_vel*math.cos(math.radians(tet))*t #variação da posição
,→total em x
x_max = x_pos + delt_sx #alcance total
ey = y_max/10 #variável controle de escala (Coordenada Y)
ex = delt_sx/20 #variável controle de escala (Coordenada X)
i_pos = (y_pos*10)/y_max - 5 #posição inicial relativa da partícula na
,→interface do vpython
ball = sphere(pos = vec(-10, i_pos, 0), radius = 0.5, color = color.
,→red, make_trail=True, trail_type='points', trail_color
=color.yellow,
,→interval = c/10)
L_origem = label(pos=ball.pos, text='(' +str(x_pos)+'
'+str(y_pos)')'
,→m', yoffset=50, xoffset = -30, space=30, height=14, border=4,
font='sans')
L_medio = label(pos=vec(((x_y_max-x_pos) -(x_max-x_pos)/2)*(20/
,→delt_sx), 5, 0), text='(' +str(round(x_y_max, 2))+'
'+str(round(y_max,
,→2))+'') m', yoffset=50, xoffset = 0, space=30, height=14, border=4,
,→font='sans')
L_fim = label(pos=vec(10, -5, 0), text='(' +str(round(x_max, 2))+'
0)'
,→m', yoffset=50, xoffset = 30, space=30, height=14, border=4,
font='sans')
solo = box(pos=vector(0, -5.6, 0), size=vector(26, 0.2, 2))
dt = 0.01
ts = 0 #passagem do tempo
janela.destroy()
while (ts<t):
    rate (c) #controle do tempo (c=100, implica em tempo real)
    curva_pos_y.plot(pos=(ts, y_pos + i_vel*math.sin(math.
,→radians(tet))*ts - (g/2)*(ts**2))) #gráfico posição em y

```

```

        curva_pos_x.plot(pos=(ts, x_pos + i_vel*math.cos(math.
        ,-radians(tet))*ts)) # gráfico posição em x
        curva_vel_y.plot(pos=(ts, i_vel*math.sin(math.radians(tet)) -
        ,-g*ts)) # gráfico velocidade em y
        ball.pos.y = (((i_vel*math.sin(math.radians(tet))*ts) - ((g/
        ,-2)*(ts**2)))/ey) + i_pos #mudança de posição em y
        ball.pos.x = ((i_vel*math.cos(math.radians(tet))*ts )/ex) -10
        ,-#mudança de posição
    except:
        lbl['text'] = 'Valores Informados Inválidos!!!'

janela = Tk()
janela.geometry(' 520x490+400+150')
janela.title(' SIMULAÇÕES')
janela['pady'] = 20
janela['padx'] = 30
container = Frame(janela, highlightbackground='black', highlightthickness=2)
container['pady'] = 20
container['padx'] = 35
container.grid(row=0, column=0)
lab_title = Label(container, text='Lançamento Oblíquo', font='times 20 bold',
    ,-height=3)
lab_i_pos = Label(container, text='Velocidade Inicial (m/s):', font='times 14
    ,-height=3)
lab_i_vel = Label(container, text='Ângulo (°):', font='times 14 ', height=3)
lab_ti = Label(container, text='x_inicial (m):', font='times 14 ', height=3)
lab_tf = Label(container, text='y_inicial (m):', font='times 14 ', height=3)
lab_title.grid(row=0, column=0, columnspan=4)
lab_i_pos.grid(row=1, column=0, columnspan=2, sticky=W)
lab_i_vel.grid(row=2, column=0, columnspan=2, sticky=W)
lab_ti.grid(row=3, column=0, sticky=W)
lab_tf.grid(row=3, column=2, sticky=W)
ed_i_vel = Entry(container, width=26, font='times 12')
ed_i_tet = Entry(container, width=26, font='times 12')
ed_i_x = Entry(container, width=7, font='times 12')
ed_i_y = Entry(container, width=7, font='times 12')
ed_i_vel.grid(row=1, column=2, columnspan=2, sticky=W)
ed_i_tet.grid(row=2, column=2, columnspan=2, sticky=W)
ed_i_x.grid(row=3, column=1, sticky=W)
ed_i_y.grid(row=3, column=3, sticky=W)
lbl = Label(container, text='', height=2, font='times 12 italic')
lbl.grid(row=4, column=1, columnspan=2)
bt = Button(container, text = 'Gerar Simulação', font='times 14 bold',
    pady=12,
    ,-command=bt_clic)

```

```

bt.grid(row=5, column=1, columnspan=2, sticky=S)
buttonExample = Button(container, text='Ajuda', font='times 12 bold',
    _command=ajuda)
buttonExample.grid(row=5, column=0, columnspan=2, sticky=W)
janela.mainloop()

```

Box 3: Código referente ao sistema com polia fixa

```

from vpython import *
from tkinter import *
def ajuda():
    newWindow = Toplevel(janela)
    newWindow.geometry('520x490+100+150')
    newWindow.title('AJUDA')
    newWindow['pady']=20
    newWindow['padx']=30
    lab_title2 = Label(newWindow, text='SISTEMA COM POLIA FIXA', font='times
    20
    _bold', height=3)
    lab_title2.grid(row=0, column=0)
    a = 'O sistema se trata de dois blocos ligados por um fio inextensível, '
    a = a + 'que passa por uma polia fixa, tendo um bloco suspenso pelo fio e
    _
    _outro'
    a = a + 'apoiado a uma parede. A partir da massa dos blocos e da altura em
    _
    _que o bloco '
    a = a + 'suspenso se encontra em relação ao solo, é possível caracterizar
    o_
    _estado '
    a = a + 'de movimento do sistema com a sua aceleração. '
    a = a + '\n\nNa simulação, é também possível verificar os vetores do peso
    e_
    _da tração nas cordas.'
    lab_desc = Label(newWindow, font='times 12 ', text=a, wraplength=450)
    lab_desc.grid(row=1, column=0)

def bt_clic():
    try:
        Parede = box(pos=vector(-5, 0.5, 0), size=vector(10, 10, 3), color=color.
        _orange)
        Solo = box(pos=vector(-2, -4, 0), size=vector(16, 1, 3), color=color.
        _orange)
        #Construção da Polia (Conjunto de Cilindros e dois Box)
        polia_a = cylinder(pos=vector(1.7, 6.2, 0), axis=vector(0, 0, 0.1), _
        _radius=0.6, color = color.yellow)
    1

```

```

polia_b = cylinder(pos=vector(1.7, 6.2, 0.1), axis=vector(0, 0, 0.2),
, radius=0.8, color = color.white)
polia_c = cylinder(pos=vector(1.7, 6.2, -0.2), axis=vector(0, 0, 0.2),
, radius=0.8, color = color.white)
barra_polia_b = box(pos=vector(0.8, 5.7, 0.39), size=vector(2.5, 0.
, -3, 0.1), axis = vector(0.5, 0.3, 0), color=color.gray(0.5))
barra_polia_c = box(pos=vector(0.8, 5.7, -0.29), size=vector(2.5, 0.
, -3, 0.1), axis = vector(0.5, 0.3, 0), color=color.gray(0.5))
#Cordas
corda_a = cylinder(pos=vector(-7, 6.75, 0), axis=vector(9, 0, 0),
, radius=0.05, color = color.yellow)
corda_b = cylinder(pos=vector(2.3, 5, 0), axis=vector(0, 1, 0), radius=0.
, -0.05, color = color.yellow)
#Blocos
massa_a = box(pos=vector(-8, 6.5, 0), size=vector(2, 2, 2), color=color.
, red, mass = 0)
massa_b = box(pos=vector(2.3, 4, 0), size=vector(2, 2, 2), color=color.
, blue, mass = 0)
#Vetores
vetor_a = arrow(pos = massa_a.pos + vec(0, 0.25, 0), axis = vec(2.5,
, -0, 0), shaftwidth=0.15)
vetor_b = arrow(pos = massa_b.pos, axis = vec(0, 2.5, 0),
, shaftwidth=0.15)
vetor_peso = arrow(pos = massa_b.pos, axis = vec(0, -2.5, 0),
, shaftwidth=0.15)
#Dados
massa_a.mass = float(ed_mass_a.get())
massa_b.mass = float(ed_mass_b.get())
h = float(ed_h.get())
#Legenda com a Altura
h_label = label(pos=vec(3.3, 3, 0), text='Altura: y = ' + str(h),
, xoffset=50, yoffset=0, space=30, height=16, border=4, font='sans')
g = 9.81 #gravidade
a = massa_b.mass*g/(massa_a.mass+massa_b.mass) #aceleração
t_total = (2*h/a)**(1/2) #tempo total
janela.destroy()
if (t_total <= 0.5): #Variável de controle do tempo para tempos
, muito curtos
    c = 5
elif (t_total <= 1):
    c = 10
elif (t_total <= 2):
    c = 20
elif (t_total <= 3):
    c = 30
elif (t_total <= 4):
    c = 40

```

```

elif (t_total <= 5):
    c = 50
else:
    c = 100

e = h/6.5 #controle de escala
dt=0.01
t=0
while (t < t_total):
    rate (c) #Controle do Tempo
    t+=dt
    #Mudança de Posição em A
    massa_a.pos.x = (((a/2)*(t**2))/e) - 8
    corda_a.pos.x = -7 + (((a/2)*(t**2))/e)
    corda_a.axis.x = 9 - (((a/2)*(t**2))/e)
    vetor_a.pos = massa_a.pos + vec(0, 0.25, 0)
    #Mudança de Posição em B
    massa_b.pos.y = 4 - (((a/2)*(t**2))/e)
    corda_b.pos.y = 5 - (((a/2)*(t**2))/e)
    corda_b.axis.y = 1 + (((a/2)*(t**2))/e)
    vetor_b.pos = massa_b.pos
    vetor_peso.pos = massa_b.pos
    h_label.text = 'h = ' + str(h - round((a/2)*(t**2), 2))
    h_label.pos=massa_b.pos

    #Definindo os Valores Finais dos Vetores
    vetor_a.shaftwidth = 0.01
    vetor_b.shaftwidth = 0.01
    vetor_peso.shaftwidth = 0.01
    h_label.text = 'h = 0'

except:
    lbl['text']='Valores Informados Inválidos!!!'

janela = Tk()
janela.geometry('650x430+400+150')
janela.title('SIMULAÇÕES')
janela['pady']=20
janela['padx']=30
container = Frame(janela, highlightbackground='black', highlightthickness=2)
container['pady']=20
container['padx']=35
container.grid(row=0, column=0)
lab_title = Label(container, text='SISTEMA COM POLIA FIXA', font='times 20,
-bold', height=3)
lab_mass_a = Label(container, text='Insira a Massa (kg) do Bloco Suspenso: ',

```

```

└─
,--font='times 14 ', height=2)
lab_mass_b = Label(container, text='Insira a Massa (kg) do Bloco Apoiado: ', └─
,--font='times 14 ', height=2)
lab_h = Label(container, text='Insira a Altura de Queda (m): ', font='times 14
└─
,--', height=2)
lab_title.grid(row=0, column=0, colspan=4)
lab_mass_a.grid(row=1, column=0, colspan=2, sticky=W)
lab_mass_b.grid(row=2, column=0, colspan=2, sticky=W)
lab_h.grid(row=3, column=0, colspan=2, sticky=W)
ed_mass_b = Entry(container, width=26, font='times 12')
ed_mass_a = Entry(container, width=26, font='times 12')
ed_h = Entry(container, width=26, font='times 12')
ed_mass_a.grid(row=1, column=2, colspan=2, sticky=W)
ed_mass_b.grid(row=2, column=2, colspan=2, sticky=W)
ed_h.grid(row=3, column=2, colspan=2, sticky=W)
lb1 = Label(container, text='', height=1, font='times 12 italic')
lb1.grid(row=4, column=1, colspan=2)
bt = Button(container, text = 'Gerar Simulação', font='times 14 bold',
pady=12, └─
,--command=bt_clic)
bt.grid(row=5, column=1, colspan=2, sticky=S)
buttonExample = Button(container, text='Ajuda', font='times 12 bold', └─
,--command=ajuda)
buttonExample.grid(row=5, column=0, colspan=2, sticky=W)
janela.mainloop()

```



Direito autoral e licença de uso: Este artigo está licenciado sob uma [Licença Creative Commons](https://creativecommons.org/licenses/by-nc-nd/4.0/).