DESENVOLVIMENTO DE UM SIMULADOR DE LANÇAMENTO OBLÍQUO COMO APOIO AO ENSINO DE FÍSICA

Autores: Guilherme Oliveira Barbosa de Andrade[1], Luiz Miguel Souza Araújo[1], Matheus Gonçalves da Silva Carvalho[1], Guilherme Miguel Pereira[1] e Hélio Lourenço Esperidião Ferreira[1].

[1]Colégio Técnico "Antônio Teixeira Fernandes" , Rua Paraibuna, 70, CEP 12245-021– Centro – São José dos Campos / SP

guioliveiraba29@gmail.com, mgsilvac01@gmail.com, luizm.univap@gmail.com, guilhermedev1204@gmail.com, helioesperidiao@gmail.com

Resumo

O lançamento oblíquo envolve movimentos simultâneos nas direções horizontal e vertical e pode ser difícil de visualizar apenas por meio de fórmulas e desenhos. Este trabalho apresenta o desenvolvimento de um simulador educacional criado por quatro alunos do 1º ano do Ensino Médio Técnico em informática, utilizando a Unity 2017 e a linguagem C#. No programa, o usuário ajusta o ângulo de lançamento, informa a velocidade inicial desejada e visualiza uma prévia da trajetória antes de executar o movimento. A interface também apresenta a gravidade e as componentes horizontal e vertical da velocidade inicial, permitindo relacionar os valores numéricos ao comportamento observado. O simulador possibilita comparar diferentes configurações de lançamento e foi desenvolvido para servir como ferramenta complementar em aulas de Física, auxiliando professores na demonstração da influência do ângulo e da velocidade inicial sobre a trajetória.

Palavras-chave: Lançamento oblíquo. Simulação computacional. Ensino de Física. Unity. C#.

Área do Conhecimento: Ensino Médio Técnico.

Introdução

O lançamento oblíquo é um movimento bidimensional no qual um objeto possui, no instante inicial, uma componente de velocidade horizontal e outra vertical. Quando a resistência do ar é desprezada, o movimento horizontal pode ser analisado separadamente do movimento vertical, que sofre a ação da aceleração da gravidade. A combinação desses dois movimentos produz uma trajetória parabólica (Moebs; Ling; Sanny, 2016).

Assim sendo, surgiu a ideia de desenvolvimento de um programa em Unity 2D, em que o usuário informa a velocidade inicial e ajusta o ângulo de lançamento. A partir desses valores, o simulador calcula as componentes horizontal e vertical da velocidade e representa visualmente a trajetória. Mesmo quando as equações são compreendidas, a passagem dos números para uma representação visual do movimento pode gerar dúvidas. Por esse motivo, uma simulação pode funcionar como apoio para comparar situações e observar de forma imediata o efeito causado pela alteração de uma variável.

Nesse contexto, recursos computacionais podem contribuir para tornar a representação do lançamento oblíquo mais acessível, principalmente quando permitem modificar os parâmetros do movimento e observar as alterações produzidas na trajetória. A possibilidade de visualizar diferentes situações a partir da mudança do ângulo e da velocidade inicial permite relacionar os cálculos realizados com o comportamento do objeto durante o lançamento. Dessa forma, o uso de uma ferramenta interativa pode complementar a abordagem tradicional do conteúdo, criando situações em que os conceitos físicos possam ser observados de maneira mais direta.

O objetivo deste trabalho foi desenvolver um simulador computacional de lançamento oblíquo que permita ajustar o ângulo e a velocidade inicial, visualizar previamente a trajetória de uma esfera e simular o lançamento, buscando criar uma ferramenta complementar que possa ser utilizada por professores para demonstrar e discutir esse conteúdo com os alunos durante aulas de física.

Metodologia

Foi utilizada Unity 2017 como ambiente de desenvolvimento e a linguagem C# para a programação das funções do simulador. Embora a Unity seja conhecida principalmente para criação de jogos, a plataforma também permite desenvolver aplicações e experiências interativas (Unity Technologies, 2018), sendo utilizada neste trabalho apenas como base para a interface e para a representação visual da simulação.

A primeira etapa foi organizar uma interface simples, de modo que o usuário pudesse iniciar a simulação sem precisar conhecer o funcionamento interno do programa. A tela inicial apresenta o título do simulador e os comandos para iniciar, ou encerrar a aplicação.

Em reuniões realizadas com o orientador deste projeto, definiu-se que ao clicar no botão “Iniciar”, na tela principal, o plano do movimento será representado por uma malha com marcações de distância. O usuário deverá informar a velocidade inicial em um campo numérico e ajustar o ângulo de lançamento por meio das setas direcionais do teclado. Na parte superior da interface gráfica são mostrados os valores da aceleração da gravidade, da velocidade inicial, das componentes horizontal e vertical e do ângulo selecionado. A Figura 1 apresenta a interface com o ângulo igual a 0° e velocidade inicial de 15 m/s.

Figura 1 - Interface de configuração dos parâmetros do lançamento.



Fonte: Autores (2026).

A fundamentação física utilizada para interpretar o lançamento considera a decomposição da velocidade inicial V0 em duas componentes. Para um ângulo θ medido a partir da horizontal, segundo Moebs, Ling e Sanny (2016), essas componentes podem ser expressas pelas relações:

$$Vx0 = V0 \cdot \cos(\theta)$$

$$Vy0 = V0 \cdot \text{sen}(\theta)$$

No simulador, o valor de g apresentado na interface é 9,81 m/s². O movimento da esfera é calculado diretamente pelas equações x(t) e y(t), usando as componentes Vx0 e Vy0 obtidas a partir da velocidade inicial e do ângulo. Assim, a posição horizontal é atualizada por $x(t) = Vx0 \cdot t$ e a posição vertical por $y(t) = Vy0 \cdot t - (g \cdot t^2)/2$. A prévia da trajetória é construída a partir desses mesmos cálculos, antes da execução do lançamento. A Unity é utilizada para representar os valores, a malha e a esfera na tela, enquanto o cálculo do movimento é realizado pelo script em C#, sem depender do mecanismo de física da Unity.

A malha apresentada na interface é utilizada como referência visual para o movimento. As posições físicas são calculadas em metros pelas equações do lançamento, enquanto a malha serve para facilitar a visualização das distâncias e não altera os cálculos. A Unity é uma plataforma de desenvolvimento com editor visual para organizar cenas, objetos e elementos de interface em aplicações 2D e 3D (Unity Technologies, 2018). A plataforma permite programar comportamentos por meio de scripts; neste projeto, eles foram escritos em C#. O C# foi utilizado para ler os valores informados pelo usuário, atualizar os dados da tela e controlar a trajetória e o movimento da esfera (Microsoft, 2026).

A Unity é uma plataforma de desenvolvimento com editor visual para organizar cenas, objetos e elementos de interface em aplicações 2D e 3D (Unity Technologies, 2018). A plataforma permite programar comportamentos por meio de scripts; neste projeto, eles foram escritos em C#. O C# é uma linguagem de programação de uso geral baseada em princípios de orientação a objetos (Microsoft,

2026). No simulador, foi utilizado para ler os valores informados pelo usuário, atualizar os dados da tela e controlar a trajetória e o movimento da esfera.

Resultados

O desenvolvimento resultou em um simulador funcional capaz de receber uma velocidade inicial, permitir a alteração do ângulo, apresentar as componentes da velocidade e desenhar uma previsão da trajetória. A alteração dos parâmetros produz representações diferentes na tela, permitindo observar de forma direta a relação entre os valores selecionados e o movimento apresentado.
Para validar quantitativamente o funcionamento do simulador, foi utilizada a configuração de velocidade inicial de 20 m/s e ângulo de 64°, apresentada na Figura 2. A partir das equações do lançamento oblíquo, com g = 9,81 m/s², foram obtidos tempo de voo de 3,66 s, altura máxima de 16,47 m e alcance horizontal de 32,13 m. Esses valores foram usados como referência para verificar a trajetória representada pelo simulador. A componente horizontal obtida foi 8,77 m/s e a componente vertical foi 17,98 m, valores que coincidem, dentro do arredondamento, com os apresentados na interface. A Tabela 1 reúne a comparação quantitativa.

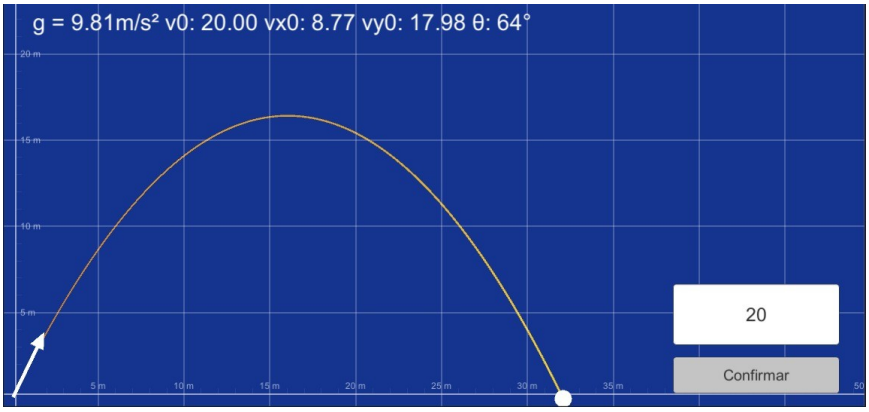
Tabela 1 - Comparação entre os valores analíticos e os valores da simulação para V0 = 20 m/s e θ = 64°.

Grandeza	Valor analítico	Valor da simulação
Tempo de voo	3,66 s	3,66 s
Altura máxima	16,47 m	16,47 m
Alcance horizontal	32,13 m	32,13 m

Fonte: Autores (2026).

Na configuração mostrada na Figura 2, por exemplo, foi utilizada velocidade inicial de 20 m/s e ângulo de 64°. A interface apresenta Vx0 igual a aproximadamente 8,77 m/s e Vy0 igual a aproximadamente 17,98 m/s. A trajetória exibida forma uma curva parabólica, com subida até uma altura máxima e posterior descida até o nível horizontal de referência.

Figura 2 - Prévia da trajetória para velocidade inicial de 20 m/s e ângulo de 64°.

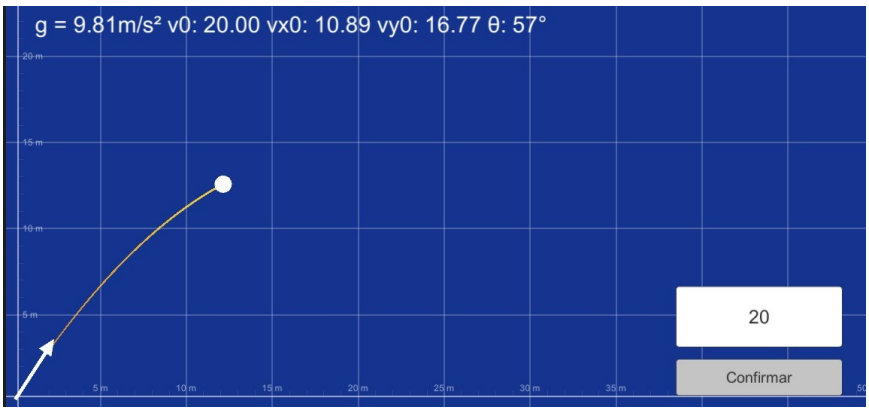


Fonte: Autores (2026).

Também foi possível acompanhar o objeto durante a execução do movimento. A Figura 3 mostra uma situação com velocidade inicial de 20 m/s e ângulo de 57°, em que o objeto ainda se encontra na parte ascendente da trajetória. Nesse caso, a tela apresenta as componentes Vx0 e Vy0 juntamente com o ângulo, permitindo que o usuário relacione a decomposição da velocidade com a direção do movimento.

Os resultados apresentados correspondem ao desenvolvimento, ao funcionamento do software e à verificação quantitativa do modelo físico implementado. A comparação realizada para a configuração de 20 m/s e 64° mostrou concordância entre os valores calculados pelas equações e os valores usados para representar a trajetória no simulador. Ainda não foram realizados testes formais com turmas, ou questionários, para medir o efeito do simulador sobre a aprendizagem. Assim, o trabalho não busca afirmar que a ferramenta melhora o desempenho escolar, mas demonstrar que ela permite representar o fenômeno e criar situações que podem ser exploradas durante uma aula.

Figura 3 - Execução do lançamento com velocidade inicial de 20 m/s e ângulo de 57°.



Fonte: Autores (2026).

Os resultados apresentados correspondem ao desenvolvimento e ao funcionamento do software simulador de lançamento oblíquo. Ainda não foram realizados testes formais com turmas, ou questionários, para medir o efeito do simulador sobre a aprendizagem. Assim, o trabalho não busca afirmar que a ferramenta melhora o desempenho escolar, mas demonstrar que ela permite representar o fenômeno e criar situações que podem ser exploradas durante uma aula.

Discussão

Como limitação, a versão atual representa um modelo idealizado do lançamento e ainda não teve sua aplicação pedagógica avaliada com uma turma. A validação realizada neste trabalho verifica o funcionamento físico do modelo, mas não mede seus efeitos sobre a aprendizagem. Uma aplicação possível é utilizar a prévia da trajetória como parte de uma questão de exercício simulado. Antes de executar o movimento, o professor pode solicitar que os estudantes indiquem o que esperam observar, após uma mudança de ângulo ou velocidade. Depois, a simulação pode ser executada e o resultado visual comparado com as previsões. Esse tipo de utilização aproxima a ferramenta da proposta de exploração e discussão conceitual associada às simulações educacionais (Wieman; Adams; Perkins, 2008).

O programa também pode ser associado aos cálculos realizados em sala. Após encontrar Vx0 e Vy0 a partir de uma velocidade inicial e de um ângulo, os alunos podem inserir a mesma configuração no simulador e observar a trajetória correspondente. Da mesma forma, os dados mostrados na parte superior da interface podem ser utilizados pelo professor para explicar por que um mesmo valor de V0 pode gerar componentes diferentes quando o ângulo é alterado.

A representação adotada é intencionalmente simples: uma grade com escalas, uma seta indicando a direção inicial, os valores numéricos e a curva do lançamento. Essa escolha mantém o foco nas grandezas físicas, sem adicionar elementos de jogo que poderiam desviar a atenção do objetivo principal do projeto. A Unity foi utilizada como ferramenta de desenvolvimento da aplicação, enquanto a finalidade do produto é exclusivamente a simulação e o apoio ao ensino.

Como continuidade, pretende-se aprimorar a ferramenta e avaliar seu uso em situações de ensino, permitindo verificar quais recursos são mais úteis para professores e estudantes. Essas etapas poderão contribuir para ajustar a interface e definir novas funções sem perder a simplicidade do simulador. A validação pedagógica, entretanto, ainda depende de testes com estudantes e instrumentos de avaliação adequados.

Conclusão

Foi desenvolvido um simulador educacional de lançamento oblíquo utilizando a Unity 2017 e programação em C#. A aplicação permite definir a velocidade inicial, ajustar o ângulo, visualizar as componentes horizontal e vertical da velocidade, observar uma prévia da trajetória e acompanhar a execução do movimento.

A organização da interface possibilita comparar diferentes configurações sem alterar a proposta principal do projeto, que é representar visualmente conceitos trabalhados em Física. Por isso, o simulador pode ser utilizado pelo professor como recurso complementar para relacionar fórmulas, valores numéricos e a trajetória observada, principalmente em atividades que envolvam comparação de ângulos e velocidades.

Em trabalhos futuros, podem ser realizados testes orientados por professores, coleta de opiniões dos usuários e inclusão de recursos como exibição do tempo de voo, altura máxima e alcance calculado, desde que essas informações sejam apresentadas de forma clara e compatível com o nível dos estudantes.

Como continuidade, pretende-se aprimorar a ferramenta e avaliar seu uso em situações de ensino, permitindo verificar quais recursos são mais úteis para professores e estudantes. Essas etapas poderão contribuir para ajustar a interface e definir novas funções sem perder a simplicidade do simulador.

Referências

MICROSOFT. Um tour pela linguagem C#. Microsoft Learn, 2026. Disponível em: https://learn.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/overview. Acesso em: 25 ago. 2026.

MOEBS, W.; LING, S. J.; SANNY, J. University Physics Volume 1. Houston: OpenStax, 2016. Seção 4.3: Projectile Motion. Disponível em: https://openstax.org/books/university-physics-volume-1/pages/4-3-projectile-motion. Acesso em: 25 ago. 2026.

UNITY TECHNOLOGIES. Unity User Manual (2017.4). 2018. Disponível em: https://docs.unity.cn/2017.4/Documentation/Manual/UnityManual.html. Acesso em: 25 ago. 2026.

WIEMAN, C.; ADAMS, W.; PERKINS, K. PhET: Simulations That Enhance Learning. Science, v. 322, n. 5902, p. 682-683, 2008. DOI: 10.1126/science.1161948.

Agradecimentos

Agradecemos a toda a equipe do colégio que nos deu apoio durante o desenvolvimento do projeto.