

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE DO CAMPUS ARARANGUÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Claudio André Korb

**Desenvolvimento de um sistema para simulação de fenômenos físicos para aplicação na
educação em turmas de Ensino Médio.**

Araranguá

2022

Claudio André Korb

Desenvolvimento de um sistema para simulação de fenômenos físicos para aplicação na educação em turmas de Ensino Médio.

Trabalho Conclusão do Curso de Graduação em Engenharia de Computação do Centro de Ciências, Tecnologias e Saúde da Universidade Federal de Santa Catarina como requisito para a obtenção do Título de Bacharel em Engenharia de Computação
Orientador: Prof^ª Olga Yevseyeva, Dr^ª

Araranguá

2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Korb, Claudio André

Desenvolvimento de um sistema para simulação de fenômenos físicos para aplicação na educação em turmas de Ensino Médio. / Claudio André Korb ; orientadora, Olga Yevseyeva, 2022. 61 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Câmpus Araranguá, Graduação em Engenharia de Computação, Araranguá, 2022.

Inclui referências.

1. Engenharia de Computação. 2. Desenvolvimento de Software. 3. Simulação física. 4. Educação. 5. Desenvolvimento web. I. Yevseyeva, Olga. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Computação. III. Título.

Claudio André Korb

Desenvolvimento de um sistema para simulação de fenômenos físicos para aplicação na educação em turmas de Ensino Médio.

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Engenharia de Computação, e foi aprovado em sua forma final pelo Curso de Engenharia de Computação.

Araranguá, 26 de Julho de 2022.

Prof.^a Analúcia Schiaffino Morales, Dr.^a.
Coordenadora do Curso

Banca Examinadora:

Prof.^a Olga Yevseyeva, Dra.
Orientadora
Universidade Federal de Santa Catarina

Prof.^a Luciana Bolan Frigo, Dra.
Universidade Federal de Santa Catarina

Prof. Marcelo Zannin Da Rosa, Dr.
Universidade Federal de Santa Catarina

RESUMO

O seguinte trabalho de conclusão de curso apresenta o desenvolvimento de um sistema de simulação de fenômenos físicos para aplicação na educação de turmas de física do ensino médio. O principal objetivo do trabalho é desenvolver um simulador que forneça ferramentas para o auxílio do ensino de física e aplicá-lo em sala de aula. Foi realizada uma pesquisa bibliográfica acerca do tema para identificar o que já foi desenvolvido e quais resultados podem ser esperados através do uso desta tecnologia para a educação. O protótipo, representado por uma aplicação *web* com *back-end* em *node.js*, simula o movimento de objetos em queda livre na superfície de diferentes corpos celestes. Também foi desenvolvida uma prova que apresenta questionamentos acerca do conteúdo estudado para aferir o domínio dos alunos sobre o mesmo. O simulador foi utilizado por uma turma de 30 alunos de ensino médio de uma escola estadual de Santa Catarina e as opiniões dos utilizadores foram coletadas através de uma pesquisa de satisfação de software. A interação dos alunos com o sistema demonstrou-se positiva e, de acordo com o professor responsável, apresentou-se um aumento no interesse dos educandos pelo tema estudado.

Palavras-chave: Simulação. Física. Ensino médio. Aplicação web.

ABSTRACT

The following undergraduate thesis presents the development process for a system that simulates physical phenomena for application on High School physics classes. The main objective of the study is to develop a simulator that can provide the necessary auxiliary tools for physics education and to apply it on real a high school classroom. A bibliographic research was conducted to gather information about what was already developed on the theme and what results can be expected with the utilization of this kind of technology in education. The prototype, represented by a web application with a node.js back-end, simulates the movement of free falling objects on the surface of different celestial bodies. A test that presents questions about the subjects of study with the intention of evaluating the knowledge of the students was also developed. The simulator was used by a High School classroom of 30 students in a public school on the State of Santa Catarina and the opinions of the users were collected through a software satisfaction survey. The interaction of the students with the system proved positive, and according to the teacher responsible for the classes, there was a noticeable increase in interest about the subject of study for the students.

Keywords: Simulations. Physics. High School. Web application

LISTA DE FIGURAS

Figura 1 - Uso diário de internet mundialmente per capita por dispositivo (em minutos).....	19
Figura 2 – Laboratório de colisões do PhET.....	24
Figura 3 – Sobre de margem causada por campo input de tamanho incorreto.....	25
Figura 4 – Simulação de um sistema massa-mola duplo.....	26
Figura 5 – Simulação de movimento uniformemente acelerado em uma dimensão.....	27
Figura 6 – Experimento do mergulhador de penhasco.....	28
Figura 7 – Parâmetros da simulação.....	29
Figura 8 – Simulador de queda livre do Laboratório de Física da Universidade Federal do Ceará.....	30
Figura 9 – Diagrama de classes da tela de simulação livre.....	37
Figura 10 – Diagrama de classes da tela de prova (simplificado).....	38
Figura 11 – Tela de login da aplicação.....	39
Figura 12 – Tela de prova com questão a ser respondida.....	40
Figura 13 – Alert Box informando resultado.....	41
Figura 14 – Simulação exibida após cada questão.....	41
Figura 15 – Resultado final da prova.....	42
Figura 16 – Tela de simulação livre.....	42
Figura 17 – Gráfico de utilização de tecnologia para acesso à internet.....	46
Figura 18 – Gráfico de utilização de navegadores.....	46
Figura 19 – Avaliação da facilidade de utilização do sistema.....	47
Figura 20 – Avaliação do desempenho do sistema.....	48
Figura 21 – Avaliação da precisão das simulações.....	49
Figura 22 – Avaliação da interface do simulador.....	50
Figura 23 – Avaliação da qualidade das simulações.....	50
Figura 24 – Gráfico sobre utilização de simuladores em salas de aula.....	51
Figura 25 – Gráfico sobre fixação do conteúdo.....	51
Figura 26 – Formato da pergunta no sistema.....	53

LISTA DE ABREVIATURAS E SIGLAS

CETIC Centro Nacional de Estudos para o Desenvolvimento da Sociedade da Informação

TIC Tecnologias de Informação e Comunicação

PhET Physics Education Technology Project

API Application Programming Interface (Interface de programação de Aplicação)

MVP Minimal Viable Product (Produto mínimo viável)

UML Unified Modeling Language (Linguagem de Modelagem Unificada)

RF Requisito funcional

S.I Sistema Internacional de unidades

ID Identificador

SUMÁRIO

1 INTRODUÇÃO	19
1.1 OBJETIVOS.....	21
1.1.1 Objetivo Geral	21
1.1.2 Objetivos Específicos	21
2 UM ESTUDO DO ESTADO DA ARTE	22
2.1 Uso de simuladores em sala de aula.....	22
2.2 Simuladores disponíveis.....	23
2.2.1 Physics Education Technology Project	23
2.2.2 MyPhysicsLab	25
2.2.3 Ophysics	26
2.2.4 Interactives.ck12.org	27
2.2.5 Physics Classroom	28
2.2.6 Laboratório virtual de física da Universidade Federal do Ceará	29
3 DESENVOLVIMENTO DO PROTÓTIPO	32
3.1 ETAPAS DE DESENVOLVIMENTO.....	32
3.2 Requisitos DO SISTEMA.....	32
3.3 Diagrama de classes.....	38
3.4 TECNOLOGIAS UTILIZADAS.....	40
3.4.1 JavaScript	40
3.4.2 Node.js	40
3.4.3 Express	41
3.4.4 Socket.io	41
3.4.5 P5.js	42
3.5 Telas do sistema.....	43
3.5.1 Tela de login	43
3.5.2 Tela de prova	44
3.5.3 Tela de simulação livre	46
3.6 DESCRIÇÃO DO FENÔMENO FÍSICO ATRAVÉS DO SOFTWARE.....	47

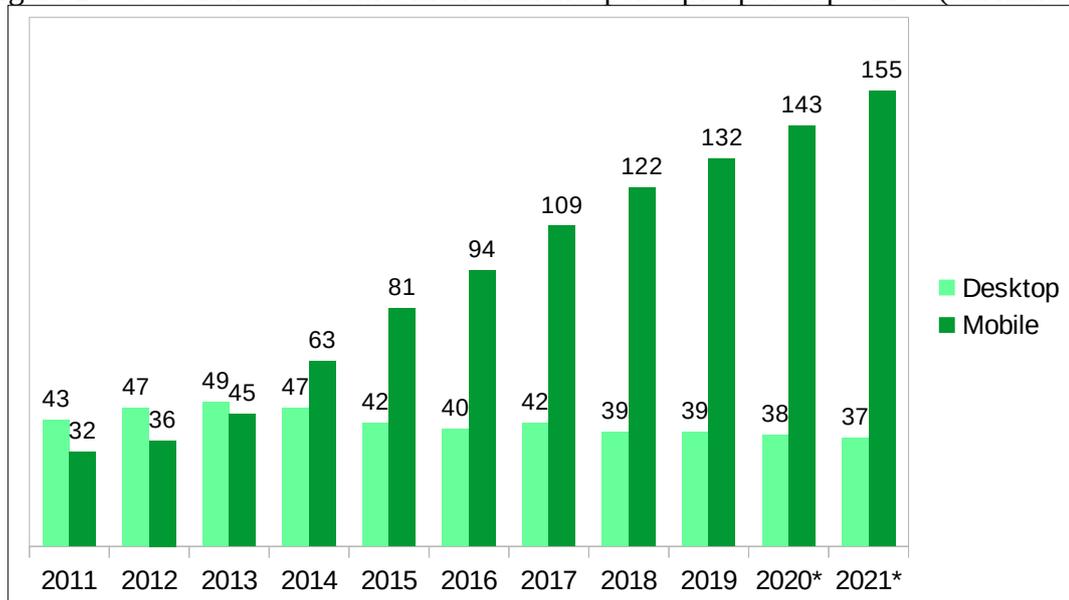
4 RESULTADOS E DISCUSSÕES.....	49
4.1 PESQUISA DE SATISFAÇÃO DO SOFTWARE.....	49
4.1.1 Metodologia da pesquisa.....	49
4.1.2 Sobre os métodos de acesso.....	49
4.1.3 Sobre a usabilidade e desempenho.....	51
4.1.4 Sobre a precisão das simulações.....	52
4.1.5 Sobre a interface.....	53
4.1.6 Sobre o auxílio no aprendizado.....	54
4.2 MELHORIAS FUTURAS.....	56
4.2.1 Novas simulações.....	56
4.2.2 Cadastro de turmas e interface do professor.....	56
4.2.3 Melhorias identificadas por usuários.....	57
4.3 CONSIDERAÇÕES FINAIS.....	58
REFERÊNCIAS.....	59
APÊNDICE A – DIAGRAMA DE CLASSES COMPLETO.....	61

1 INTRODUÇÃO

Com a tecnologia em desenvolvimento crescente nas últimas décadas, é notável a presença quase que natural dos dispositivos tecnológicos na sociedade moderna. Segundo uma pesquisa publicada em 2019 pela empresa alemã de estatística especializada em dados mercadológicos, Statista, o tempo médio gasto na internet mundialmente aumentou de 75 para 171 minutos por dia entre 2011 e 2019 (STATISTA, 2019).

Um aumento ainda mais significativo é encontrado quando analisamos os dados de dispositivos: a mesma pesquisa indica que a quantidade de minutos gastos na internet diariamente em dispositivos móveis já supera a dos *desktops* desde 2014, e a diferença só vem aumentando desde então.

Figura 1 - Uso diário de internet mundialmente per capita por dispositivo (em minutos)



Fonte: Statista

É seguro afirmar que hoje os dispositivos móveis já são utilizados com maior frequência do que os computadores pessoais por boa parte da população, portanto soluções tecnológicas implementadas para mobile têm a capacidade de alcançar uma grande quantidade de usuários.

Uma pesquisa realizada pelo Centro Nacional de Estudos para o Desenvolvimento da Sociedade da Informação (CETIC) em 2018, indicou que 97% dos estudantes de escolas urbanas utilizam telefones celulares para o acesso à internet, sendo que 84% dos mesmos afirmaram que os *smarthphones* são o principal equipamento que utilizam para o acesso à *web* (CETIC.BR, CGI.BR e NIC.BR, 2018).

Qualquer tecnologia tão difundida está fadada a criar alguns problemas sociais. Alguns estudos já foram conduzidos para tentar mitigar uma reclamação que parece ser recorrente entre professores: as tecnologias de computação estão desviando o foco dos alunos, que passam mais tempo na internet do que estudando (JIANHONG, 2010).

A hegemonia dos celulares com acesso à internet traz a tona a possibilidade de uma nova metodologia de ensino: a utilização dos telefones como ferramenta pedagógica, uma maneira de auxiliar o professor a transmitir conhecimento para uma geração que nasceu na era da informação. Segundo Abdulayeva, o uso de smartphones pode gerar resultados proveitosos no ensino de física experimental, sendo um substituto interessante para os laboratórios tradicionais, através de simulações de fenômenos físicos (ABDULAYEVA, 2021).

O trabalho abordará as fases de desenvolvimento do protótipo de um simulador de fenômenos físicos online. A versão desenvolvida simula apenas o fenômeno em estudo (corpos em queda livre), ideias de expansões futuras são discutidas no capítulo 4.

A ideia do projeto veio de uma demanda do curso de Mestrado Profissional em Ensino de Física da Universidade Federal de Santa Catarina, campus Araranguá, que se trata de um projeto de simulador (físico) de impacto de corpos em queda livre.

Durante a demonstração do projeto, enfatizou-se a necessidade de criar um mecanismo que pudesse ser facilmente reproduzido por professores para aplicação em sala de aula, focando na simplicidade do projeto e na redução dos custos de produção do mesmo.

Porém, segundo os autores, mesmo com um projeto barato e de fácil reprodução, a confecção do mesmo demanda um certo trabalho e o mínimo de habilidade manual e de manuseio de ferramentas por parte do professor (considerando que o mesmo terá interesse de apresentar o simulador físico em sala de aula).

Dessa ideia, surgiu a opção da criação, junto ao modelo físico, de um protótipo de simulador virtual, que pudesse ser acessado por qualquer pessoa com um computador ou *smartphone* e acesso à internet, que representasse fenômenos semelhantes aos descritos pelo produto educacional desenvolvido no mestrado e que fornecesse ferramentas de avaliação do aprendizado ao professor.

Dito isso, tratou-se os autores do projeto como os clientes finais da aplicação, aqueles que solicitaram e que definiram os requisitos do sistema. Durante o documento serão descritas as fases de desenvolvimento do protótipo do simulador, que incluem reuniões de alinhamento e exibição das versões iniciais do produto.

Na seção 2 apresenta-se uma revisão bibliográfica, um estudo do estado da arte do que é discutido sobre o uso de simuladores em salas de aula, bem como experimentos já existentes e seus resultados.

Na seção 3 apresenta-se a metodologia utilizada para o desenvolvimento do protótipo, as práticas de Engenharia de *Software* utilizadas e também exibimos alguns diagramas que representam o funcionamento do sistema.

Na seção 4 encontram-se os resultados obtidos bem como discussões gerais sobre os mesmos, também apresentamos as sugestões de melhoria para o projeto e para trabalhos futuros

1.1 OBJETIVOS

Nas seções abaixo estão descritos o objetivo geral e os objetivos específicos deste TCC.

1.1.1 Objetivo Geral

Desenvolver uma solução tecnológica que forneça ao professor ferramentas para auxiliar o processo de ensino de física, incluindo um simulador de fenômenos físicos que seja de fácil utilização.

1.1.2 Objetivos Específicos

Os objetivos específicos do trabalho são:

1. Fazer um levantamento acerca dos recursos tecnológicos já existentes que possam suprir (ou não) a demanda identificada;
2. Desenvolver um protótipo para aplicação em salas de aula que possa ser utilizado pela maior parte da população;
3. Estudar os efeitos da aplicação do simulador, através de uma pesquisa de satisfação de *software* apresentada aos alunos;
4. Aprimorar o conhecimento relacionado ao desenvolvimento de *software* e suas boas práticas.

2 UM ESTUDO DO ESTADO DA ARTE

Existem estudos que buscam entender os efeitos da utilização de simulações computacionais em sala de aula na transmissão de conhecimento. A seguir apresentam-se alguns dos estudos encontrados utilizando o repositório IEEE Xplore. Também são listados alguns dos principais simuladores disponíveis na internet.

2.1 USO DE SIMULADORES EM SALA DE AULA

Para a pesquisa, foi utilizado o filtro “physics simulation AND education AND high school”, limitando os resultados para o período de 2000 à 2022. Foram encontrados 211 resultados que foram classificados e a seguir apresentamos alguns dos mais relevantes.

Ben Ouahi e seus coautores apresentam uma pesquisa de opinião com professores de física e química do ensino fundamental e médio de escolas do Marrocos sobre a utilização de simulações do repositório Physics Education Technology Project (PhET), ou projeto tecnológico de educação em física, da Universidade do Colorado em Boulder (BEN OUAHI et. al., 2020).

A maioria dos pesquisados informou que utiliza as simulações disponíveis no PhET, utilizando um método investigativo de educação, onde os alunos são incentivados a aprender através de desafios e problemas apresentados. Os autores apresentam também três fatores importantes que dificultam a utilização das simulações: falta de recursos tecnológicos em sala de aula; falta de experiência e conhecimento sobre os benefícios da utilização de simulações no ensino; problemas inerentes ao sistema de educação utilizado (especificamente salas de aula lotadas).

Gerlič e S. Ülen realizam uma pesquisa com estudantes de ensino médio de escolas da Eslovênia. O objetivo do estudo é comparar dois métodos de ensino: o tradicional, utilizando livros, fórmulas e exercícios de fixação e o chamado 'método conceitual', utilizando o computador como ferramenta de ensino, através de pequenas simulações de fenômenos físicos em aplicações *web*. (GERLIČ e S. ÜLEN, 2011)

Os resultados obtidos mostram que estudantes introduzidos ao método conceitual apresentaram um grau de conhecimento superior em todas as 5 competências analisadas (conhecimento, análise, comparação, inferência e avaliação), tanto em conceitos de elétrica quanto de magnetismo.

Os autores finalizam com: “Concluimos que simulações computacionais podem ter um papel importante no processo educacional moderno, portanto não existem motivos para

não incorporar Tecnologias de Informação e Comunicação (TIC) na educação contemporânea” (GERLIČ e S. ÜLEN, 2011).

Hattori e seus coautores, apresentam uma ideia interessante para trazer parte da interatividade proporcionada por métodos computacionais ao mundo real. Utilizando um *software* de modelagem 3D chamado Blender, os autores apresentam um conceito onde as simulações realizadas no computador podem ser reproduzidas no mundo real através da impressão 3D. (HATTORI et. al., 2020).

Muitos *softwares* de modelagem 3D modernos apresentam a funcionalidade de criar animações e até simulações simples, utilizando linguagens de alto nível como o Python. Os autores exploram essas possibilidades através do programa Blender.

O artigo apresenta o exemplo do lançamento horizontal de uma esfera. Utilizando o *software* é possível criar a animação e definir visões de câmera diferentes para acompanhar o movimento. É possível simular os efeitos de aceleração gravitacional, velocidade e resistência do ar utilizando *scripts* em Python. Os autores recomendam então imprimir o modelo da esfera utilizando uma impressora 3D e recriar o experimento utilizando câmeras no mundo real.

O *paper* apenas introduz o conceito e indica que pode ser promissor no ensino de física, porém não apresenta uma pesquisa ou estudo que mostre sua eficácia em sala de aula. Outro ponto importante a se ressaltar é a necessidade de uma impressora 3D, uma ferramenta cara e de difícil utilização, o que dificilmente estará amplamente disponível para escolas de ensino fundamental e médio no Brasil nos próximos anos.

2.2 SIMULADORES DISPONÍVEIS

Existem diversos simuladores de fenômenos físicos disponíveis na internet. Porém uma pesquisa entre os resultados mais populares do Google.com indicou que muitos deles não atendem todos os requisitos levantados pelos clientes do sistema.

Para a pesquisa dos simuladores disponíveis na internet, focou-se nos que disponibilizam conteúdo gratuito dentre os principais resultados gerados por uma pesquisa com a chave “physics simulations”.

2.2.1 Physics Education Technology Project

Primeiramente foram analisadas as simulações disponíveis no Physics Education Technology Project (PhET), um projeto da Universidade do Colorado em Boulder. Um

diferencial desta é que está disponível inteiramente em português no site da universidade. Existem diversas simulações disponíveis, entre elas: montagem de circuitos AC, Transformadas de Fourier, simulações de gases, molas, colisão de corpos rígidos, etc.

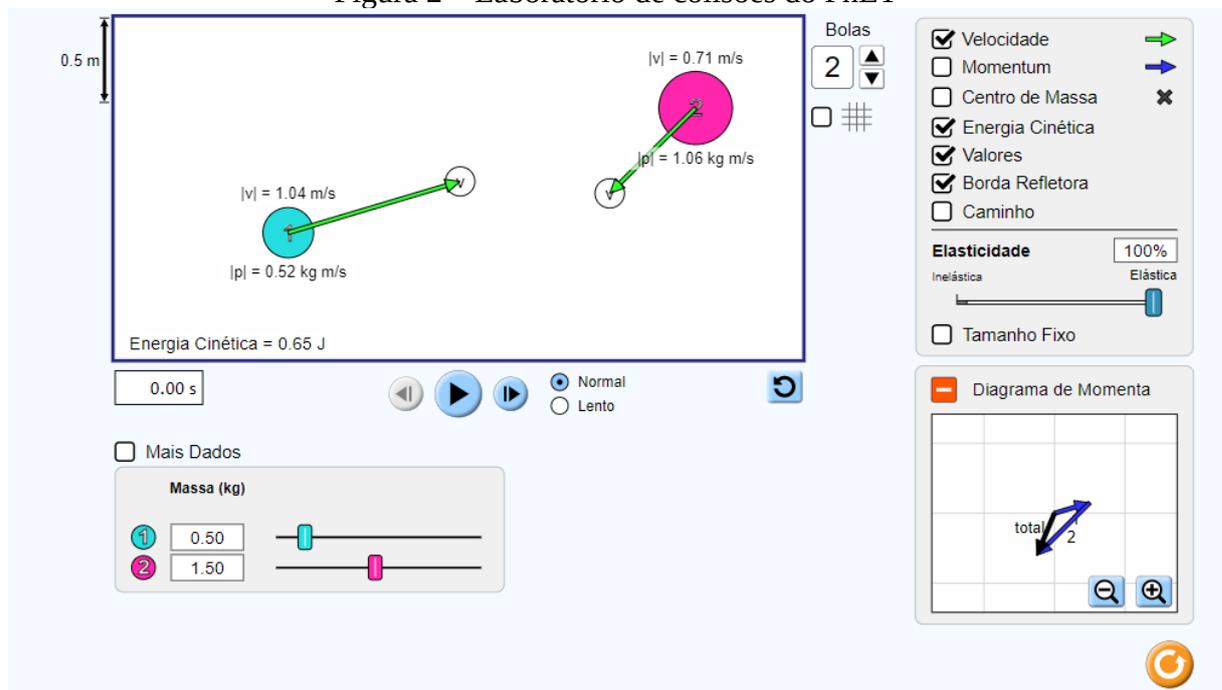
Outro ponto positivo do *website* é o fato de ser totalmente compatível com navegadores *mobile*, uma vez que suas animações funcionam utilizando HTML5, tecnologia que é facilmente exibida em qualquer navegador moderno.

Um experimento interessante disponível no *website* é o do laboratório de colisões. Nele podemos explorar conceitos como momento, conservação da energia cinética, velocidade, entre outros. Pode-se alterar o nível de elasticidade das colisões, variando de inelástica (coeficiente de restituição = 0) à elástica (coeficiente de restituição = 1).

Alguns dos conceitos aplicados no protótipo podem ser explorados no laboratório de colisões do PhET, porém por tratar de movimento uniforme, não temos a concepção de aceleração e de aplicação de forças externas em um corpo.

Não foi encontrada uma experiência que aplicasse a Segunda Lei de Newton em corpos em queda livre na biblioteca do PhET.

Figura 2 – Laboratório de colisões do PhET



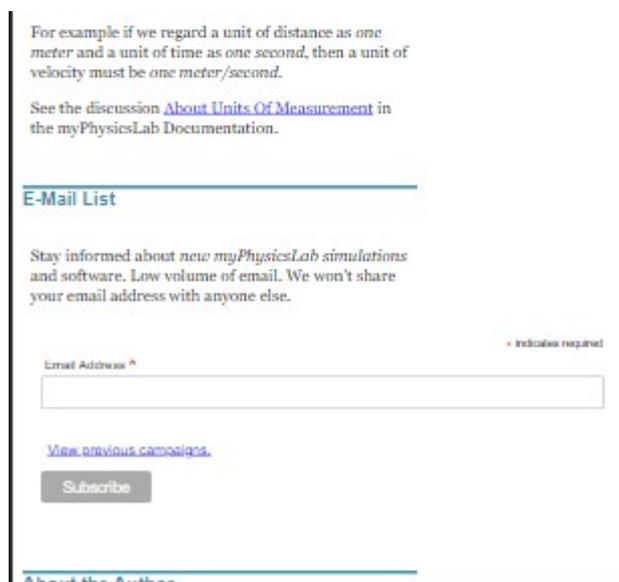
Fonte: University of Colorado Boulder

2.2.2 MyPhysicsLab

A segunda biblioteca de simuladores analisada foi a do MyPhysicsLab. Nela encontramos 50 experimentos diferentes que aplicam variados conceitos, entre eles: Molas, pêndulos, colisão de corpos rígidos, atração molecular, etc.

Assim como o PhET, o *website* em questão e suas experiências funcionam em dispositivos móveis, porém a interface não é otimizada para os mesmos, apresentando sobras de margem devido a elementos com tamanho não padronizado.

Figura 3 – Sobra de margem causada por campo input de tamanho incorreto

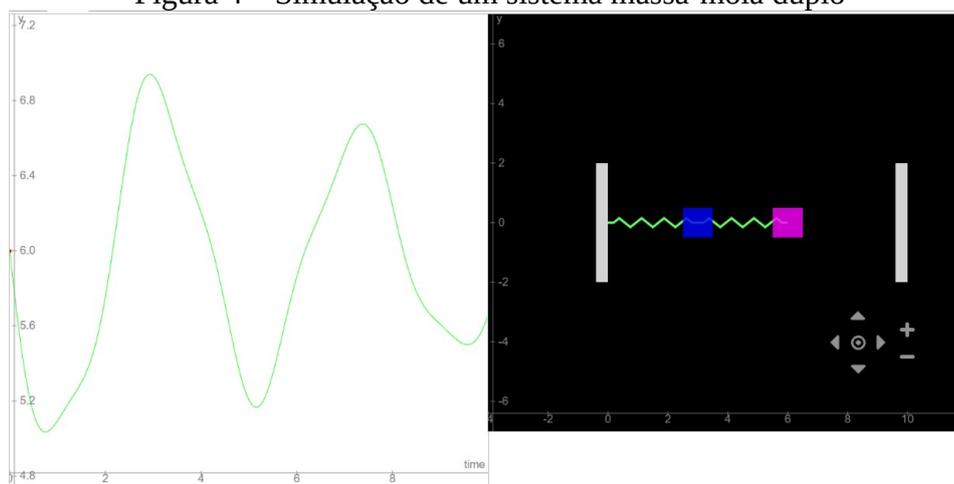


Fonte: myphysicslab.com

Percebeu-se que boa parte das simulações disponíveis (19 de 50) envolvem variações de pêndulos (simples, duplos, caóticos, com mola...) e não foi encontrado um experimento que explorasse todos os conceitos aplicados no protótipo.

As simulações são simples de um ponto de vista visual, porém apresentam recursos interessantes, como a construção de gráficos com as diferentes variáveis disponíveis em cada experimento. A figura 4 mostra um exemplo de experimento encontrado no site. Um sistema massa-mola duplo, no qual podemos plotar a posição da massa rosa ao longo do tempo.

Figura 4 – Simulação de um sistema massa-mola duplo



Fonte: myphysicslab.com

2.2.3 Ophysics

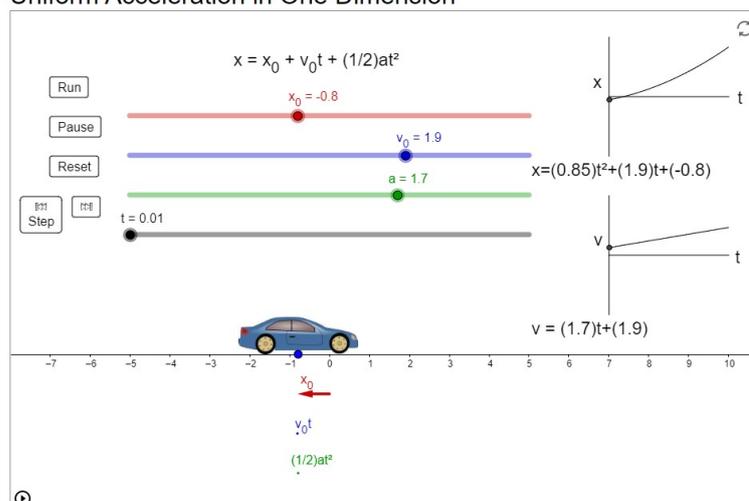
O *website* ophysics.com foi o terceiro analisado. As simulações encontradas em sua biblioteca são em sua maioria estáticas, e envolvem a manipulação de variáveis para construção de gráficos. A quantidade de conceitos explorada é ampla, variando de cinemática, forças e conservação de energia mecânica até luz, ondas, elétrica, e física moderna.

Boa parte dos experimentos é desenvolvido utilizando a aplicação GeoGebra, um *software* de matemática dinâmica, que permite – entre outras funções – a criação de gráficos interativos. O aplicativo disponibiliza uma Interface de Programação de Aplicação (API) para aqueles que desejam integrar suas soluções em suas aplicações. O uso não comercial é de livre distribuição e a utilização do mesmo para fins comerciais requer uma licença e um acordo colaborativo.

A biblioteca do site conta com 103 experimentos, divididos em 11 categorias, sendo que a maior parte das simulações (53%) estão concentradas em 3 delas: Ondas, luz e cinemática. O site não tem opção de seleção de idioma, estando disponível apenas em inglês.

Não foi encontrada uma simulação que explorasse os conceitos aplicados no protótipo. O experimento mais próximo envolve movimento uniformemente acelerado e a função horária da posição.

Figura 5 – Simulação de movimento uniformemente acelerado em uma dimensão
Uniform Acceleration in One Dimension



Fonte: ohphysics.com

A utilização do software GeoGebra torna as animações fluídas e visualmente agradáveis, apesar de simples. O site e suas animações funcionam bem em dispositivos móveis e todos os experimentos apresentam uma breve descrição dos fenômenos abordados no mesmo.

2.2.4 Interactives.ck12.org

O próximo repositório analisado é o do *website* interactives.ck12.org. A página apresenta dois tipos de simuladores: os de física e os de química. Por se tratar do escopo do trabalho, focaremos nos experimentos de física.

A biblioteca do site é bem completa, abordando conteúdos de cinemática à elétrica, passando inclusive por conceitos simples de radioatividade e física moderna.

Não é disponibilizada uma tradução para português, sendo os idiomas disponíveis inglês, coreano, alemão, chinês, grego e polonês.

Existe um experimento semelhante ao estudado neste trabalho, mas de uma maneira mais lúdica, o que parece ser a temática abordada pela maioria das simulações do site. Nele podemos observar um mergulhador olímpico pulando da encosta de um penhasco até atingir um corpo d'água abaixo do mesmo. Na questão das variáveis, podemos alterar a altura do penhasco, variando entre cinco valores preestabelecidos (de 20 à 40 metros de altura) e a resistência do ar, que pode ser nenhuma, típica ou alta.

Infelizmente os valores utilizados para a resistência do ar não são fornecidos e é difícil deduzi-los uma vez que os únicos resultados apresentados estão em formato gráfico que foca no formato das curvas e não nos valores absolutos.

Figura 6 – Experimento do mergulhador de penhasco



Fonte: interactives.ck12.org

O conceito de energia cinética não é explorado, tema que é importante para o protótipo desenvolvido e como não se pode alterar o valor de massa do mergulhador (pois o mesmo não importa para a simulação apresentada) não é possível deduzí-la com os valores apresentados. Também não é possível alterar o valor de velocidade inicial nem da aceleração gravitacional.

No geral, as animações do site são bonitas e funcionais, o site é responsivo e também funciona em dispositivos móveis.

2.2.5 Physics Classroom

O próximo *website* analisado foi o physicsclassroom.com. Além das simulações apresentadas, o site conta com um serviço chamado *Concept Builder* ou construtor de conceitos (tradução livre). O serviço busca dar ferramentas para professores de modo a auxiliar no ensino de conceitos específicos de física, apresentando aos estudantes questões selecionadas aleatoriamente que devem ser respondidas em três níveis de dificuldade diferentes. O serviço é pago, variando de U\$0,80 à U\$1,00 por aluno ativo.

As simulações disponíveis no site são gratuitas e divididas em 12 tópicos: cinemática unidimensional, leis de Newton, vetores, projéteis e movimento em duas dimensões; forças em duas dimensões; momento e colisões; trabalho e energia; movimento circular e gravitação;

equilíbrio e rotação; eletricidade estática; circuitos elétricos; magnetismo; ondas e som; luz e cores; reflexão e espelhos, e química.

Existe uma simulação que explora conceitos semelhantes aos do protótipo desenvolvido, porém durante vários testes não foi possível fazer a mesma funcionar. Analisando as variáveis que se pode modificar percebe-se que os temas abordados são parecidos com os procurados.

Figura 7 – Parâmetros da simulação

Axis Convention: Down is + ... Up is - Change

Quantity	Value	Constraints
Δ time (s):	10	Minimum Value: 0.001 s
Init. Height (m):	50	Must be > 1 m
Init. Velocity (m/s):	1	Use of +/- sign must follow axis convention.
g (N/kg):	9.81	Use of +/- sign must follow axis convention.
Mass (kg):	10	Minimum Value: 1 kg
Profile Area (m ²):	1	Must be > 0 m ²
Drag Coefficient:	1	View Wikipedia page for guidance.
Air Dens. (kg/L):	1.22	View Wikipedia page for guidance.

Fonte: physicsclassroom.com

Juntamente com as simulações, existem pequenas descrições dos conceitos abordados e algumas apresentam atividades pré-desenvolvidas, caso os usuários desejem uma experiência direcionada além das simulações livres. O site não está disponível em português

2.2.6 Laboratório virtual de física da Universidade Federal do Ceará

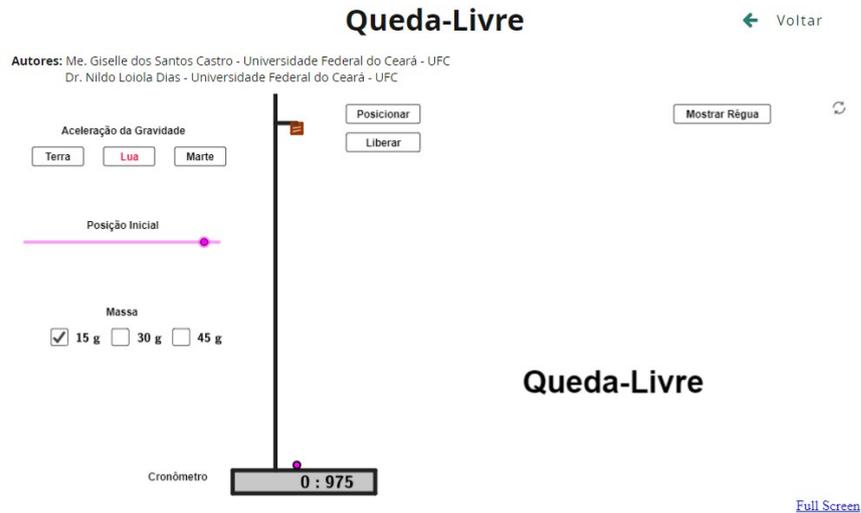
Dentre os simuladores disponibilizados por instituições de ensino brasileiras, destaca-se o Laboratório virtual de física da Universidade Federal do Ceará, que conta com 47 diferentes simulações, abordando temas de mecânica, ondulatória, termodinâmica, eletricidade e magnetismo, óptica e física moderna.

Além dos simuladores online, também estão disponíveis roteiros para a execução das experiências em sala de aula, tanto utilizando as simulações do site quanto materiais de baixo custo para reprodução dos experimentos.

O site está disponível inteiramente em português e, apesar de não contar com uma versão para dispositivos móveis, suas simulações funcionam no *mobile*, pois são desenvolvidas utilizando o aplicativo GeoGebra.

A biblioteca do site conta com uma simulação de corpos em queda livre, porém as variáveis disponíveis são limitadas, contando com apenas 3 valores de massa, utilizadas para demonstrar a influência (ou falta de influência) no tempo de queda. Também não são apresentados valores de velocidade ou de energia cinética.

Figura 8 – Simulador de queda livre do Laboratório de Física da Universidade Federal do Ceará



Fonte: <https://www.laboratoriovirtual.fisica.ufc.br/>

Na tabela abaixo encontra-se um resumo dos principais simuladores disponíveis na internet e sua relação com as funcionalidades desejadas no protótipo.

Tabela 1 - Principais simuladores disponíveis na internet

Site	Grátis	Simulador de queda livre	Português	Mobile
PhET	Sim	Não	Sim	Sim
MyPhysicsLab	Sim	Não	Não	Sim*
OPhysics	Sim	Não	Não	Sim
interactives.ck	Sim	Sim	Não	Sim
PhysicsClassroom	Sim	Sim**	Não	Sim
Laboratório de Física UFC	Sim	Sim	Sim	Sim*

* Simulações funcionam no *mobile*, mas layout do site não é muito amigável

** Não foi possível fazer a simulação funcionar

Fonte: Autor

Entre os estudados, destacam-se as simulações disponíveis em interactives.ck, que incluem um experimento sobre aplicação da 2ª Lei de Newton sobre corpos em queda, apesar de focar na relação entre resistência do ar e velocidade de queda. Porém, como muitos dos outros *websites*, não está disponível em português.

Também não foi encontrada uma solução que apresentasse uma ferramenta de avaliação, conforme solicitado nos requisitos do projeto. A solução que mais se aproxima são os roteiros de simulação fornecidos pelo laboratório virtual da Universidade Federal do Ceará.

3 DESENVOLVIMENTO DO PROTÓTIPO

O protótipo foi desenvolvido com base nas necessidades levantadas pelo mestrando responsável pelo projeto do curso de mestrado em ensino de física da Universidade Federal de Santa Catarina e pela professora orientadora do mesmo.

Durante as etapas de desenvolvimento do protótipo, o mestrando e orientadora foram tratados como os clientes solicitantes do *software* e serão tratados nas seções seguintes apenas como ‘clientes’.

3.1 ETAPAS DE DESENVOLVIMENTO

O desenvolvimento do protótipo foi dividido nas seguintes etapas:

1. **Reunião inicial para reconhecimento da demanda:** Onde foi identificada a oportunidade para o desenvolvimento de uma solução para auxiliar o professor no processo de ensino de física;
2. **Levantamento de requisitos junto ao cliente:** Onde foram definidos os requisitos mínimos de projeto;
3. **Modelagem do software utilizando ferramentas de UML:** Onde o sistema foi construído de maneira conceitual, através das ferramentas de modelagem;
4. **Desenvolvimento do protótipo inicial:** Onde o produto mínimo viável (MVP) foi desenvolvido utilizando as tecnologias disponíveis;
5. **Apresentação ao cliente:** Onde o MVP foi apresentado e foram discutidas as melhorias necessárias para aplicação em sala de aula;
6. **Melhorias solicitadas:** Onde foram desenvolvidas as melhorias solicitadas;
7. **Aplicação em sala de aula e coleta de *feedback*:** Onde o protótipo foi aplicado em sala de aula e as opiniões dos alunos foram coletadas.

3.2 REQUISITOS DO SISTEMA

Os requisitos funcionais representam funções a serem realizadas pelo software. Normalmente seguem um padrão, contendo os elementos: descrição da função a ser realizada; origem do requisito; informações de entrada e/ou saída; restrições lógicas aplicáveis (WAZLAWICK, 2011).

Os requisitos do projeto foram definidos em reuniões com os clientes. Durante os encontros, foram levantados os seguintes requisitos funcionais:

1. Simular o fenômeno de queda livre em corpos na superfície da Terra;
2. Permitir a mudança do valor da aceleração gravitacional local;
3. Permitir a mudança do valor da massa do corpo de queda;
4. Permitir a mudança da velocidade inicial do corpo de queda;
5. Permitir a mudança da altura inicial do corpo de queda;
6. Mostrar valores de velocidade e energia cinética do corpo de queda a cada instante de tempo;
7. Permitir a identificação dos alunos no sistema;
8. Apresentar questionamentos sobre os experimentos para avaliar o aprendizado dos alunos;
9. Apresentar nota final de avaliação para os alunos;

Abaixo, apresentam-se as descrições de cada um dos requisitos funcionais (RF). As descrições explicam os RFs e também apresentam as restrições (requisitos não funcionais) relacionadas a cada um deles. As especificações seguem um padrão apresentado na referência WAZLAWICK 2011, página 28:

1. Simular o fenômeno de queda livre em corpos na superfície da Terra

Descrição: O sistema deverá apresentar uma simulação gráfica do fenômeno de um corpo em queda livre quando na presença de uma força gravitacional. O usuário insere as variáveis de entrada e o programa deverá simular as condições físicas e apresentar os resultados de maneira visual.

Usuário:

- O aluno.

Informações de entrada:

- Velocidade inicial do corpo;
- Altura da queda;
- Massa do corpo;
- Aceleração gravitacional.

Informações de saída:

- Uma simulação visual do fenômeno estudado.

Restrições lógicas: Não há

Restrições tecnológicas:

- Deve ser desenvolvido na linguagem de programação javascript;
- Deve ser compatível (funcional) e responsivo tanto em plataformas *desktop* como *mobile*;
- A simulação deverá funcionar com uma taxa de atualização de 120 Hz (atualizações por segundo);
- Os valores das variáveis deverão ser apresentados no Sistema Internacional de Unidades (S.I.).

2. Permitir a mudança do valor da aceleração gravitacional local

Descrição: O sistema deverá apresentar valores preestabelecidos para aceleração gravitacional da simulação em formato de lista contendo nome, imagem e valor da aceleração gravitacional com base no valor aproximado da aceleração gravitacional de quatro corpos celestes:

- Terra: (10 m/s²)
- Marte: (4 m/s²)
- Júpiter: (25 m/s²)
- Lua: (1.6 m/s²)

Usuário:

- O aluno.

Informações de entrada:

- Corpo celeste desejado

Informações de saída:

- Sistema carregará o novo valor da aceleração gravitacional para a simulação.

Restrições lógicas: Não há

Restrições tecnológicas: Não há

3. Permitir a mudança do valor da massa do corpo de queda

Descrição: O sistema deverá permitir que o usuário insira um valor para a massa do objeto de queda.

Usuário:

- O aluno.

Informações de entrada:

- Valor da massa do corpo de queda.

Informações de saída:

- Sistema carregará o novo valor da massa do corpo de queda para a simulação.

Restrições lógicas:

- O sistema deverá considerar apenas números inseridos na entrada;
- O sistema deverá considerar qualquer valor inserido como estando em kilogramas (kg).

Restrições tecnológicas:

- O valor deverá ser recebido através de um campo de *input* de texto.

4. Permitir a mudança do valor da velocidade inicial do corpo de queda.

Descrição: O sistema deverá permitir que o usuário insira um valor para a velocidade inicial do corpo de queda.

Usuário:

- O aluno.

Informações de entrada:

- Valor da velocidade inicial do corpo de queda.

Informações de saída:

- Sistema carregará o novo valor da velocidade inicial do corpo de queda para a simulação.

Restrições lógicas:

- O valor da velocidade inicial deverá ser maior ou igual a zero;
- Valor da velocidade inicial deverá ser considerado como estando em metros por segundo (m/s);
- A velocidade inicial será sempre considerada em direção perpendicular ao solo;
- O sistema deve considerar apenas números inseridos na entrada.

Restrições tecnológicas:

- O valor deverá ser recebido através de um campo de *input* de texto;
- Deverá ser possível ignorar a entrada de velocidade inicial, considerando a mesma igual à zero.

5. Permitir a mudança da altura inicial do corpo de queda

Descrição: O sistema deverá permitir que o usuário insira um valor para a altura da queda.

Usuário:

- O aluno.

Informações de entrada:

- Valor da altura da queda.

Informações de saída:

- Sistema carregará o novo valor da altura da queda ao simulador.

Restrições lógicas:

- O valor da altura da queda deverá estar entre 1 e 100;
- O valor da altura deverá ser considerado em metros.

Restrições tecnológicas:

- O valor deverá ser recebido através de um campo de *range* (*slider*).

6. Mostrar valores de velocidade e energia cinética do corpo de queda a cada instante de tempo.

Descrição: O sistema deverá mostrar os valores da velocidade de queda a cada instante de tempo.

Usuário: Não há.

Informações de entrada: Não há.

Informações de saída:

- Valores de energia cinética e velocidade do corpo de queda, exibidos no frame de animação

Restrições lógicas: Não há

Restrições tecnológicas:

- Deve haver um *switch* para controlar a exibição dos valores;
- O valor da energia cinética deverá ser exibido em Joules (J), com precisão de duas casas decimais;
- O valor da velocidade deverá ser exibido em metros por segundo (m/s), com precisão de duas casas decimais.

7. Permitir a identificação dos alunos no sistema.

Descrição: O sistema deverá possibilitar a identificação dos alunos, através de uma tela de login, solicitando o nome do estudante.

Usuário:

- O aluno.

Informações de entrada:

- *String* contendo o nome do aluno.

Informações de saída:

- Sistema carrega informação do nome do usuário.

Restrições lógicas:

- O sistema de avaliação não deverá permitir a entrada de usuários que não tenham realizado o login.

Restrições tecnológicas:

- O sistema deve armazenar a informação da sessão (estado) como questões, acertos e erros, e questão atual e relacioná-lo ao usuário.
- Caso um usuário devidamente logado e que já tenha iniciado uma avaliação saia do sistema, ao voltar deverá retornar ao estado que saiu.

8. Apresentar questionamentos sobre os experimentos para avaliar o aprendizado dos alunos.

Descrição: O sistema deverá apresentar perguntas aos alunos a respeito dos conceitos estudados e avaliá-los com base nas respostas.

Usuário:

- O aluno.

Informações de entrada:

- A resposta do aluno à uma questão apresentada.

Informações de saída:

- Sistema carrega resposta (correta ou incorreta) do usuário.

Restrições lógicas:

- Cada pergunta deverá ter apenas uma resposta correta;
- Perguntas em formato de múltipla escolha.

Restrições tecnológicas:

- O sistema deverá selecionar as perguntas de maneira aleatória utilizando um banco de questões em formato JSON ;
- As respostas das perguntas deverão ser armazenadas em arquivo JSON separado ao arquivo de perguntas, relacionando as mesmas através de um identificador único;
- Ao realizar o login, o sistema deverá atribuir 5 questões aleatórias ao aluno e manter esta lista em memória, de modo a atender à restrição tecnológica apresentada no RF7;
- O sistema de avaliação deverá funcionar em tela separada ao sistema de simulação livre.

9. Apresentar nota final de avaliação ao aluno.

Descrição: O sistema deverá apresentar a nota final do estudante com base nas suas respostas às perguntas apresentadas.

Usuário:

- O aluno.

Informações de entrada:

- Lista com as respostas (corretas e incorretas) do aluno.

Informações de saída:

- Nota final, apresentada em tela, juntamente ao nome do aluno.

Restrições lógicas:

- As notas deverão estar em um intervalo de 0 a 5;
- O sistema deverá apresentar a nota do aluno apenas no final da avaliação.

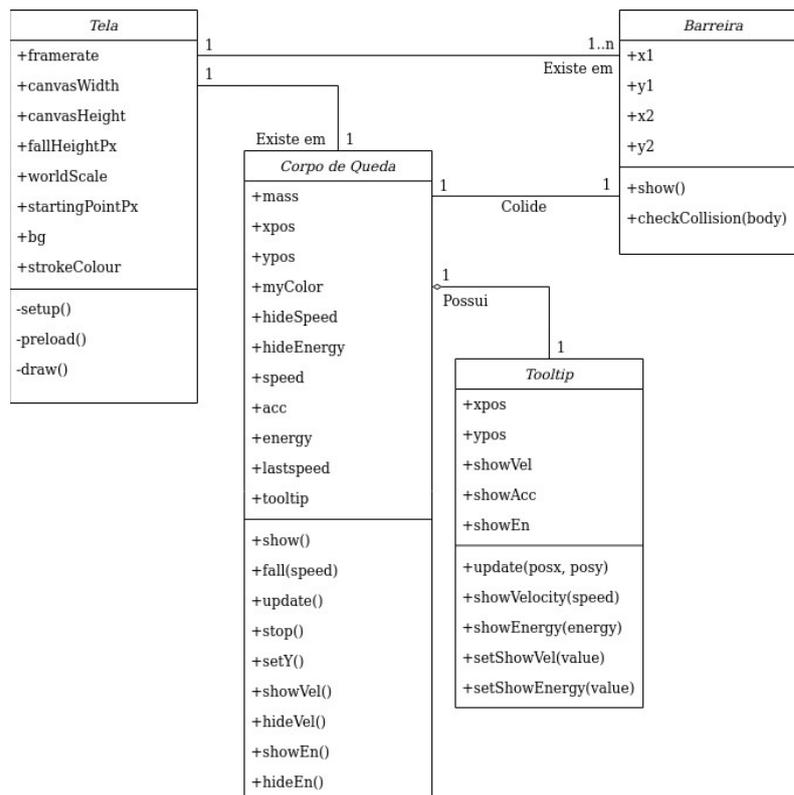
Restrições tecnológicas:

- O sistema deverá apresentar a nota do aluno em tela separada.

3.3 DIAGRAMA DE CLASSES

A seguir, apresentam-se os diagramas de classes referentes ao sistema. Neles podemos identificar as principais classes utilizadas no desenvolvimento do código.

Figura 9 – Diagrama de classes da tela de simulação livre

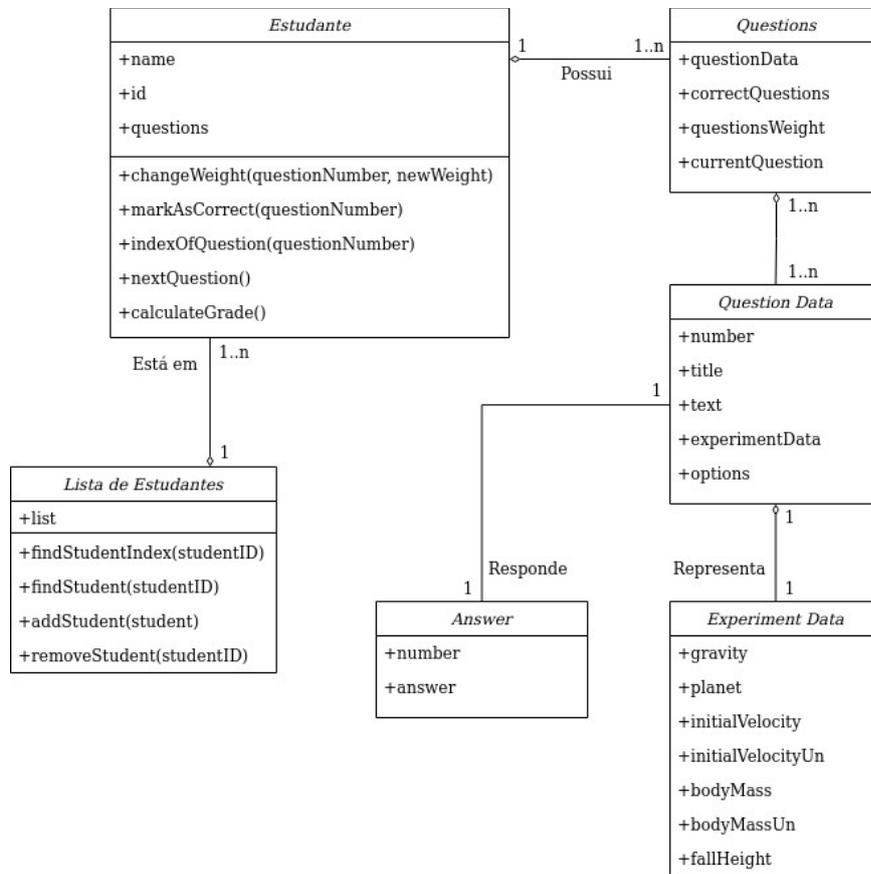


Fonte: Autor

No diagrama representado pela figura 9, a classe ‘Tela’ representa o *namespace* da tela de animação fornecida pela biblioteca p5.js e não uma classe propriamente dita. O corpo de queda é o objeto afetado pela aceleração gravitacional e que ‘cai’ em direção ao solo. O solo é representado pela classe ‘Barreira’ que é implementado através de uma linha (2 pontos

em um plano 2D). A classe ‘Tooltip’ identifica as informações de velocidade e energia cinética do corpo, exibidas em tela.

Figura 10 – Diagrama de classes da tela de prova (simplificado)



Fonte: Autor

A figura 10 mostra uma representação simplificada do diagrama de classes da tela de prova. Na prática, como a tela de prova também mostra a simulação como parte do fluxo, os dois diagramas se unem, uma vez que a classe ‘*Experiment Data*’ altera os parâmetros de simulação. O apêndice A apresenta o diagrama completo.

A classe ‘Estudante’ armazena as informações importantes sobre o aluno. A lista de estudantes armazena todos os estudantes ativos no momento (conexões inativas são excluídas periodicamente). A classe ‘Questions’, contida na classe ‘Estudante’ armazena as informações das perguntas fornecidas a um determinado aluno. ‘Question Data’ é uma classe contida na classe ‘Questions’ que representa os principais elementos de uma pergunta, incluindo seu identificador único. Como a resposta das perguntas é armazenada de maneira a parte, também é representada por uma segunda classe ‘Answers’. As perguntas e respostas se relacionam pelo identificador único ‘número’. Cada pergunta contém uma série de parâmetros, como velocidade inicial do corpo, aceleração gravitacional, altura da queda, etc. Esses parâmetros

são armazenados na classe ‘Experiment Data’. Essa classe é responsável por definir os parâmetros da simulação apresentada após a resposta de cada pergunta.

3.4 TECNOLOGIAS UTILIZADAS

Aqui apresentam-se algumas das principais tecnologias utilizadas no desenvolvimento do protótipo.

3.4.1 JavaScript

O sistema foi desenvolvido utilizando a linguagem de programação JavaScript, com back-end (representado pelo servidor) em node.js. A linguagem foi selecionada principalmente por sua ampla utilização e suporte. Uma pesquisa realizada pelo *website* Stack Overflow em 2022 indicou que JavaScript é a linguagem de programação mais popular entre os desenvolvedores, sendo utilizada por 65.36% dos respondentes (STACK OVERFLOW, 2022).

A questão do amplo suporte em navegadores, tanto *desktop* quanto *mobile* é importante e se relaciona com o RF1. Segundo a documentação encontrada no MDN Web Docs, *website* oficial de padrões de desenvolvimento da Mozilla Corporation, todos os navegadores modernos oferecem suporte para a linguagem, desde 2012 (MDN WEBDOCS, 2020).

3.4.2 Node.js

Node.js é um *runtime* JavaScript amplamente utilizado na indústria pela sua facilidade em criar aplicações de rede escaláveis. A plataforma utiliza um modelo de eventos que, diferente de outras soluções baseadas em máquinas de eventos, implementa o *loop* como parte do *runtime* ao invés de apresentá-lo como uma biblioteca a parte (OPENJS FOUNDATION, entre 2009 e 2022). Na prática isso significa que não é necessário iniciar o *loop* de eventos através de uma chamada de função, o mesmo inicia automaticamente quando o *script* de entrada é executado.

Uma das maiores vantagens da utilização do node.js para o programador é a grande quantidade de bibliotecas de desenvolvimento disponíveis, fruto de uma grande comunidade de desenvolvedores produzindo conteúdo de código aberto para ampla utilização.

O administrador de pacotes padrão do node.js é o *Node Package Manager* (npm). O npm é o maior arquivo de *software* do mundo (NPM Inc, 2022), contando com mais de 2 milhões de pacotes (em julho de 2022).

3.4.3 Express

O módulo express, disponível no npm, permite a criação de servidores HTTP (*Hyper Text Transfer Protocol*) de maneira fácil, rápida e robusta. O *framework* conta com uma série de funcionalidades para o desenvolvimento de servidores para aplicações *web* (OPENJS FOUNDATION 2017). Na prática, isso significa que se pode criar um servidor *web* pronto para receber conexões com poucas linhas de código:

Figura 11 - Código para criação de servidor web utilizando o Express

```
1 import express from 'express'
2 let app = express()
3 let port = 3000
4
5 app.get('/', (req, res) => {
6   res.send('Hello World!')
7 })
8
9 app.listen(port, () => {
10  console.log(`Example app listening on port ${port}`)
11 })
```

Fonte: Autor

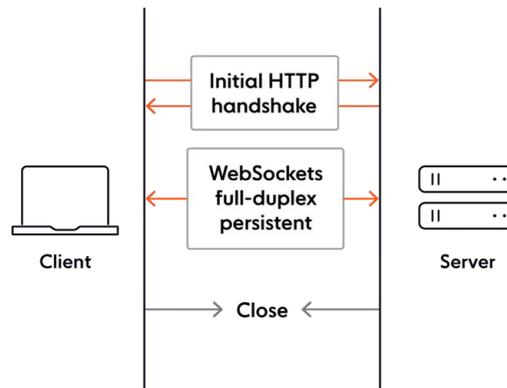
3.4.4 Socket.io

O socket.io é uma biblioteca disponível no npm, que implementa um sistema de comunicação bidirecional entre uma aplicação servidor e uma aplicação cliente. A biblioteca conta com ferramentas de confiabilidade para estabelecimento de conexões, suporte para reconexão automática, detecção de desconexão, entre outros (SOCKET.IO, 2022).

A biblioteca se baseia no conceito de *sockets*, uma abstração derivada do protocolo TCP/IP, onde cada ponto de conexão é chamado de *socket* (em português, soquete, como o final de uma conexão elétrica). Um servidor *web* abre um *socket* e o associa a uma porta de rede específica que passa a ‘escutar’ novas conexões. Uma aplicação cliente pode, por sua vez, se comunicar com essa porta através de um *socket* próprio, estabelecendo assim uma conexão ponto-a-ponto.

O grande benefício da utilização da biblioteca é a possibilidade de transmitir mensagens entre cliente e servidor sem lidar com requisições HTTP, permitindo comunicação em tempo real de baixa latência, sem redirecionamentos.

Figura 12– Modelo de comunicação WebSockets



Fonte: <https://ably.com/blog/introducing-the-websocket-handbook>

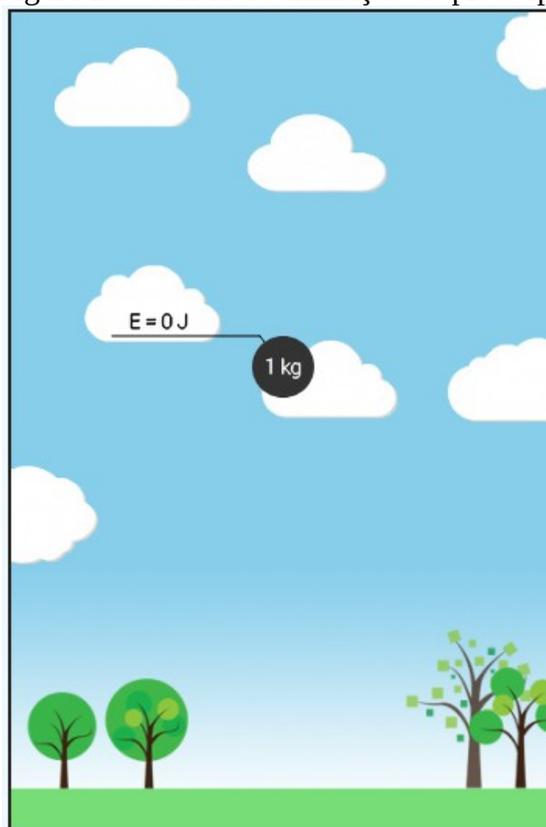
3.4.5 P5.js

P5.js é uma biblioteca para JavaScript que tem como objetivo promover o desenvolvimento criativo de código, promovendo uma plataforma intuitiva e acessível tanto para profissionais da indústria de *software* quanto para artistas, designers, professores, entre outros (YE, Quianquian; MCCARTHY, entre 2020 e 2022).

A biblioteca utiliza o elemento *canvas* do HTML5 para fornecer funcionalidades de desenho e animação em páginas *web*. Para o projeto, a biblioteca é importante por permitir a criação de animações de maneira fácil e rápida, sem a necessidade de escrita extensiva de código JavaScript.

O *framework* implementa seu próprio *loop* de animações que utiliza automaticamente uma taxa de atualização (*framerate*) predefinida pelo desenvolvedor, o que facilita o desenvolvimento. Também é possível definir vários parâmetros da animação através da função `setup()`, como tamanho da janela de animação, cor do plano de fundo, carregamento de imagens, etc.

Figura 13 – Janela de animação do protótipo



Fonte: Autor

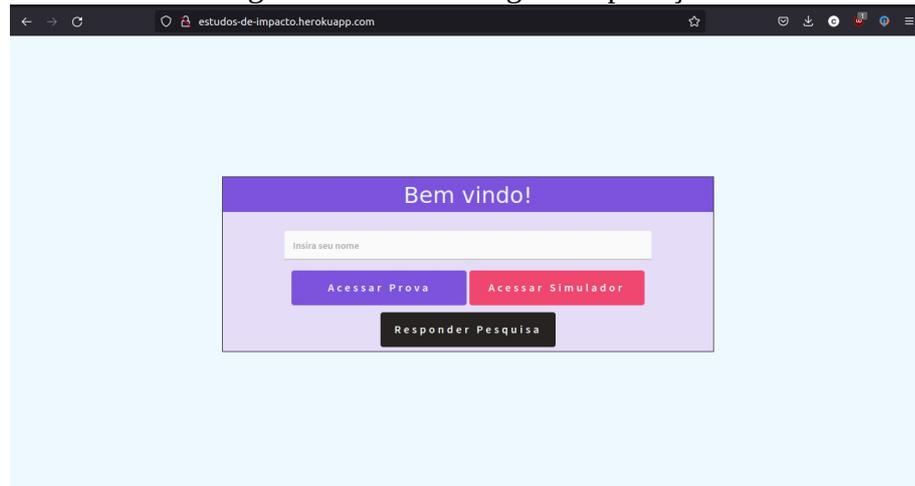
3.5 TELAS DO SISTEMA

Nas seções a seguir, são exibidas as telas do sistema. Durante o desenvolvimento, foi identificada a necessidade da criação de 3 diferentes telas para atender aos requisitos de projeto, são elas: Tela de login, Tela de prova e Tela de simulação livre. Todas as telas foram desenvolvidas em HTML5 com estilização em CSS puro. As imagens exibidas na seção, foram retiradas em um computador pessoal, as telas em dispositivos móveis são um pouco diferentes.

3.5.1 Tela de login

O objetivo da tela de login é a identificação do usuário. Atualmente a autenticação é feita de maneira simplificada, apenas com o nome do aluno. Ao realizar o login, um identificador (ID) único de sessão é gerado e armazenado em memória no servidor. Esse ID é utilizado para identificar o aluno e não o seu nome. Quando um aluno já autenticado volta à tela de login manualmente, ele é automaticamente reconhecido (a aplicação *client-side* armazena o ID através de um *cookie*) e é redirecionado à tela de prova (caso ainda não a tenha finalizado).

Figura 14 – Tela de login da aplicação



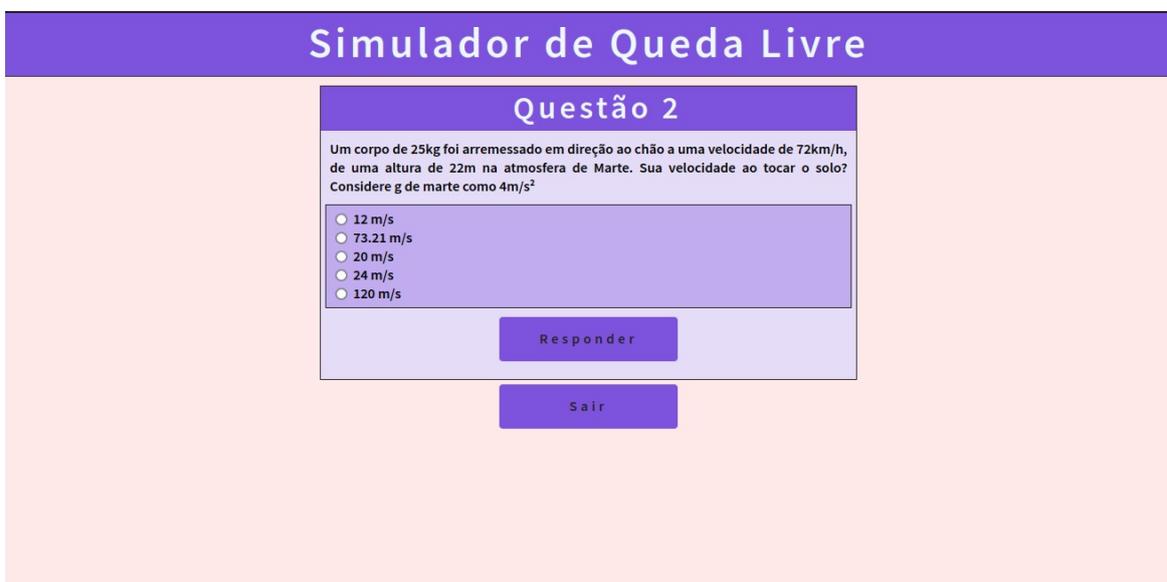
Fonte: Autor

A tela conta com um campo de *input* de texto para identificação do aluno e 3 botões. ‘Acessar Prova’ redireciona para a tela de prova (caso o usuário tenha inserido seu nome), ‘Acessar Simulador’ redireciona para a tela de simulação livre (não é necessária identificação) e ‘Responder Pesquisa’ redireciona para um questionário destinado a avaliar a satisfação dos usuários quanto ao *software* (apresentado na seção 4).

3.5.2 Tela de prova

A tela de prova apresenta as questões para avaliar o conhecimento do usuário. Após a resposta é exibida uma simulação em tempo real representando o fenômeno descrito pela pergunta.

Figura 15 – Tela de prova com questão a ser respondida

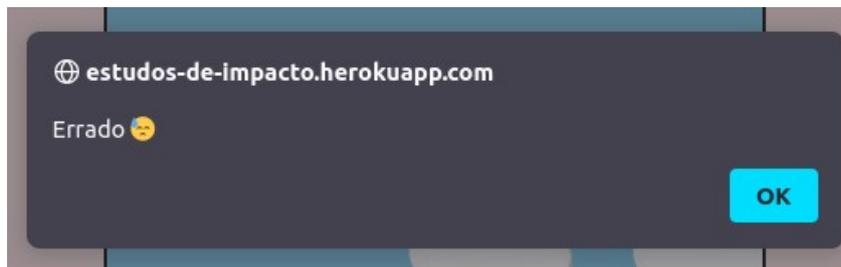


Fonte: Autor

A tela de prova conta com o número da questão atual, o texto da pergunta e as opções para múltipla escolha. Somente um dos *radio buttons* pode ser selecionado por vez. Ao clicar em ‘Responder’ a seleção é enviada ao servidor, que informa se a resposta está correta ou incorreta. As respostas não são armazenadas no *client side*, apenas no servidor.

Ao clicar em ‘Sair’ o usuário faz *logout* do sistema, excluindo todo seu progresso na prova. Uma mensagem é exibida ao clicar, informando que o progresso atual será perdido caso o usuário prossiga.

Figura 16 – *Alert Box* informando resultado



Fonte: Autor

Após informar o resultado (correto ou errado), o sistema mostra uma simulação representando o fenômeno estudado na questão.

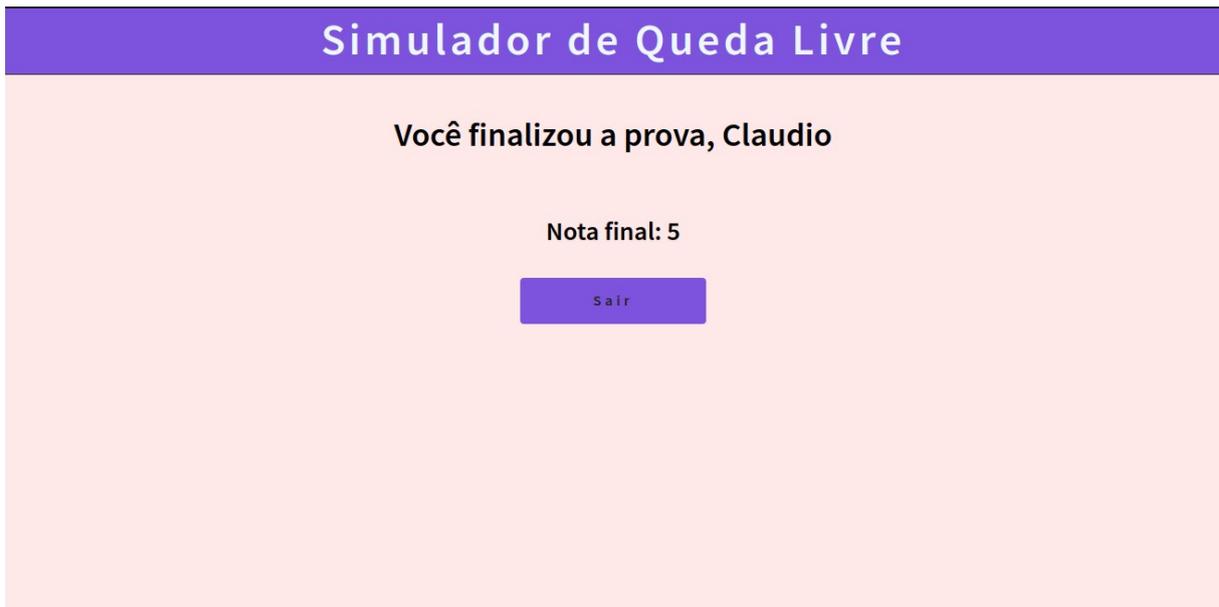
Figura 17 – Simulação exibida após cada questão



Fonte: Autor

A simulação mostra dados como a velocidade e energia cinética a cada instante de tempo e a massa do corpo. O usuário pode então clicar em ‘Replay’ para reexecutar a simulação ou em ‘Próxima pergunta’ para prosseguir com a prova.

Figura 18 – Resultado final da prova



Fonte: Autor

Ao responder a última pergunta, a tela exibe a nota final do usuário, com uma mensagem informando o final da prova. O usuário pode então clicar em 'Sair' para voltar à tela de login.

3.5.3 Tela de simulação livre

A tela de simulação livre é onde o usuário pode alterar os parâmetros de simulação para entender o comportamento de corpos em queda livre. É possível alterar vários parâmetros sobre o corpo de queda e sobre o ambiente.

Figura 19 – Tela de simulação livre



Fonte: Autor

Nos controles do corpo é possível alterar a altura de queda e massa do objeto. Também é possível dar uma velocidade inicial de queda e alterar a exibição da energia cinética e da velocidade do corpo.

Nos controles de atmosfera é possível selecionar em qual corpo celeste a simulação ocorrerá, alterando o valor da aceleração gravitacional e mudando também o plano de fundo do *frame* de animação.

Também foi incluído um botão que redireciona à pesquisa de satisfação de *software*.

3.6 DESCRIÇÃO DO FENÔMENO FÍSICO ATRAVÉS DO SOFTWARE

Para simular os fenômenos físicos estudados no programa, deve-se fazer algumas alterações nas fórmulas que descrevem os mesmos, uma vez que a simulação funciona em *steps*, ou passos de simulação, controlados pela taxa de atualização da mesma (nesse caso o *framerate* de animação). Isso significa que a simulação utiliza uma representação de tempo discretizada, contra a variável contínua das fórmulas físicas.

Se a velocidade em um instante de tempo 't' é encontrada através da equação (RESNICK e HALLIDAY, 1983):

$$v = v_0 + a \cdot t$$

Pode-se descrever variação da velocidade entre *steps* de simulação com *framerate* 'f' como:

$$v = v_{\text{anterior}} + a/f$$

Onde v_{anterior} representa a velocidade no *step* anterior. Passando a dedução acima para a linguagem JavaScript:

```
speed += acc * (1 / framerate);
```

Outro ponto importante surge ao exibir os elementos em tela. A posição de um corpo no mundo real é representada por sua distância ao ponto de origem do sistema de coordenadas. No sistema internacional de unidades, essa distância é representada em metros (m), porém em uma tela, os objetos são representados por *pixels*, e é necessária uma conversão. Para isso,

definiu-se uma escala que descreve a quantidade de metros representada por cada *pixel* da tela de animação:

```
let worldScale = MAX_FALL_HEIGHT_METERS / MAX_FALL_HEIGHT_PX;
```

Portanto, pode-se representar a posição 'y' de um objeto em um determinado *step* de simulação como:

```
ypos += speed / (framerate * worldScale);
```

O que está sendo feito, efetivamente, é converter a velocidade de m/s para *pixels/frame*. Como a posição é atualizada a cada *frame*, basta somar a velocidade ao valor da posição anterior para obter a posição atual em *pixels*.

4 RESULTADOS E DISCUSSÕES

Neste capítulo serão apresentadas as opiniões dos usuários do *software* obtidas através de um questionário de satisfação e também algumas ideias para melhorias futuras.

4.1 PESQUISA DE SATISFAÇÃO DO SOFTWARE

A pesquisa de satisfação foi desenvolvida utilizando a ferramenta Google Forms, que fornece uma série de funcionalidades para a criação de formulários online. O link para a pesquisa foi anexado às telas de simulação e de login do simulador.

4.1.1 Metodologia da pesquisa

Com o auxílio de um professor de física, o simulador foi aplicado a uma turma do primeiro ano do ensino médio, com um total de 30 alunos. Destes, 24 responderam a pesquisa de satisfação.

A aplicação foi realizada em uma das aulas da turma com o professor durante um período de classe (45 minutos). O simulador foi apresentado aos alunos após a contextualização teórica e solicitou-se que os educandos utilizassem o sistema como ferramenta de auxílio na resolução das atividades de fixação e que, após a resolução, respondessem a pesquisa de satisfação. Foi permitido que os alunos que não portavam uma ferramenta de acesso à internet realizassem a atividade em casa.

Para avaliar a qualidade do *software*, foram criadas 13 perguntas, sendo 4 de conceito aberto e 9 de múltipla escolha. Para cada pergunta foi acrescentada uma pequena descrição com os principais pontos a serem avaliados ao responder a mesma. O objetivo da pesquisa não foi avaliar a eficácia do método de ensino, mas sim coletar as opiniões dos usuários acerca do simulador, avaliando sua facilidade de utilização, interface e precisão.

A seguir apresentam-se os resultados das 12 primeiras perguntas, sendo a última um espaço para comentários e sugestões.

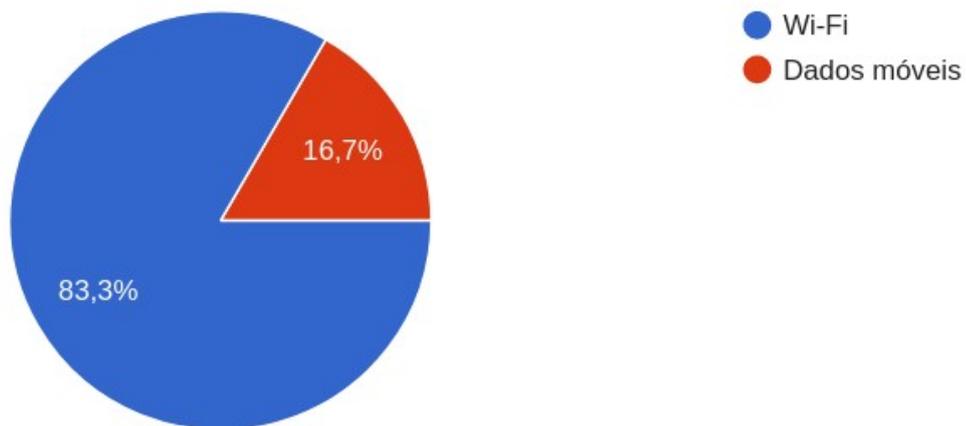
4.1.2 Sobre os métodos de acesso

Para avaliar os métodos utilizados para acessar o simulador, foram realizadas três perguntas:

1. Qual dispositivo você usou para acessar o simulador?
2. Caso tenha utilizado um dispositivo móvel (smartphone ou tablet), você acessou o sistema utilizando o wi-fi ou dados móveis (3g/4g)?
3. Qual navegador você utilizou para acessar o simulador?

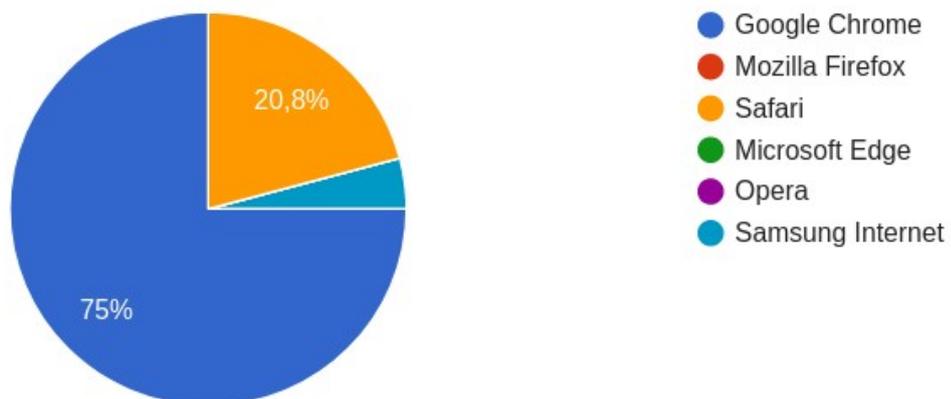
100% dos respondentes informaram que acessaram o simulador através de *smartphones*, sendo que 16.7% utilizaram dados móveis e 83.3% wi-fi para conexão à internet. Destes, a maioria (75%) informou utilizar o navegador Google Chrome, seguidos de Safari (20.8%) e Samsung Internet (4.2%).

Figura 20 – Gráfico de utilização de tecnologia para acesso à internet.



Fonte: Google Forms

Figura 21 – Gráfico de utilização de navegadores.



Fonte: Google Forms

Não se percebeu nenhuma correlação entre os métodos utilizados para acessar o simulador e as respostas qualitativas acerca do *software*, o que indica que a preocupação com portabilidade se demonstrou pertinente, uma vez que a experiência de usuário não foi afetada pelo dispositivo utilizado.

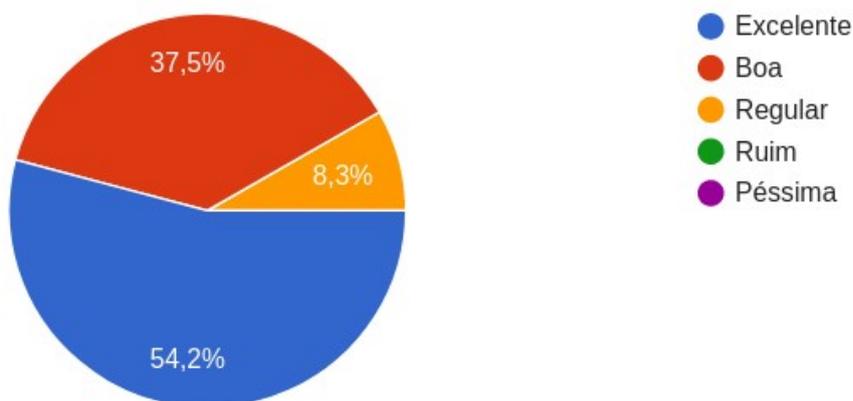
4.1.3 Sobre a usabilidade e desempenho

Foi feita uma pergunta acerca da facilidade de utilização do sistema e duas acerca do desempenho:

4. Como você avalia a facilidade de utilização do sistema?
5. Como você avalia o desempenho do sistema?
6. Caso tenha encontrado algum bug ou travamento, poderia detalhar brevemente o ocorrido?

As duas primeiras perguntas foram criadas com respostas de múltipla escolha e a terceira com resposta aberta. Para as perguntas 4 e 5, os resultados estão nos gráficos abaixo:

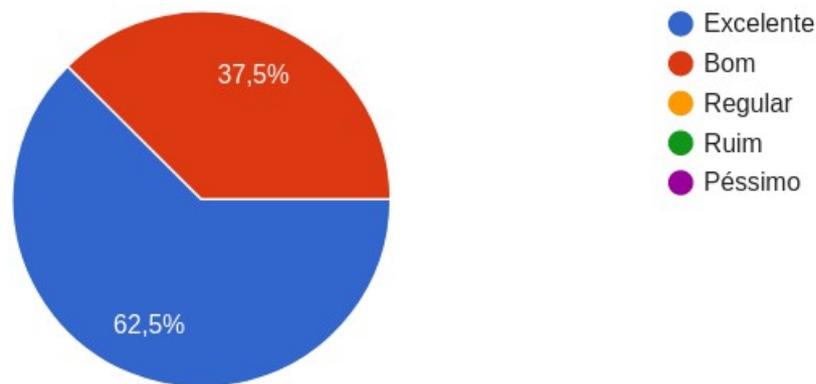
Figura 22 – Avaliação da facilidade de utilização do sistema.



Fonte: Google Forms

Durante a aplicação, foi solicitado que o professor interferisse o mínimo possível, para garantir que as únicas instruções de utilização fossem as disponibilizadas pelo sistema, de modo a possibilitar a avaliação da facilidade de utilização do mesmo.

Figura 23 – Avaliação do desempenho do sistema.



Fonte: Google Forms

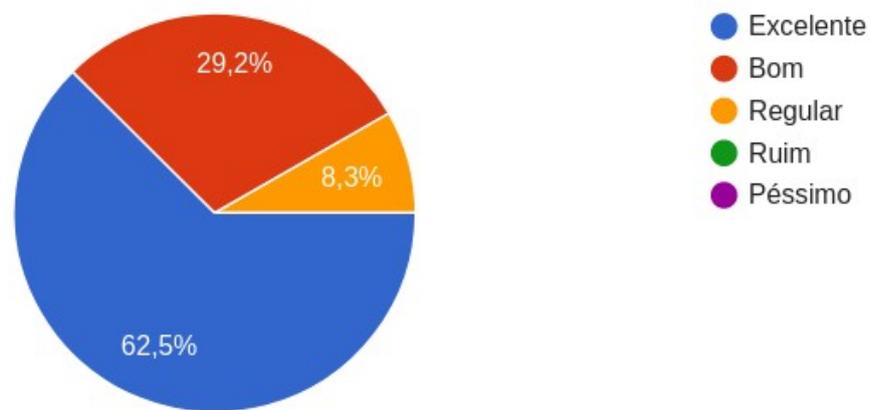
Para a pergunta 6, deixou-se aberto para que os usuários descrevessem dificuldades encontradas acerca do desempenho do sistema, como *bugs* ou travamentos. Apenas um dos respondentes informou que algumas das simulações demoravam um pouco pra carregar. Isso não foi observado durante o desenvolvimento e provavelmente é causado pelo servidor de hospedagem do sistema, que fica nos Estados Unidos, uma vez que o comportamento não se repete em servidor local.

4.1.4 Sobre a precisão das simulações

Para avaliar a precisão das simulações foram feitas duas perguntas, a primeira de múltipla escolha e a segunda de resposta aberta:

7. Como você avalia a precisão das simulações apresentadas?
8. Caso as simulações tenham apresentado algum resultado incorreto, poderia descrever brevemente o acontecido?

Figura 24 – Avaliação da precisão das simulações



Fonte: Google Forms

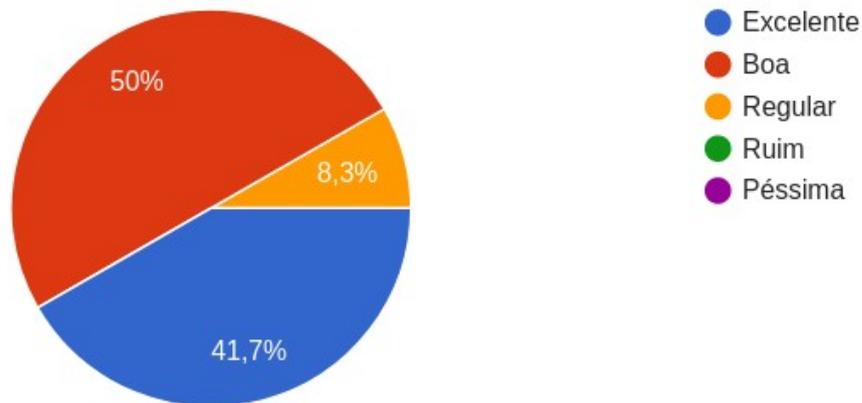
A figura 31 mostra a resposta dos usuários à pergunta 7. Percebe-se que a grande maioria considerou a precisão das simulações boa ou excelente. Apenas um respondente indicou, na pergunta 8, que algumas das respostas das questões não eram exatas, o que pode ser causado pelos arredondamentos necessários para trabalhar com números reais em JavaScript.

4.1.5 Sobre a interface

Para avaliar a interface e qualidade gráfica do simulador, duas perguntas de múltipla escolha foram feitas:

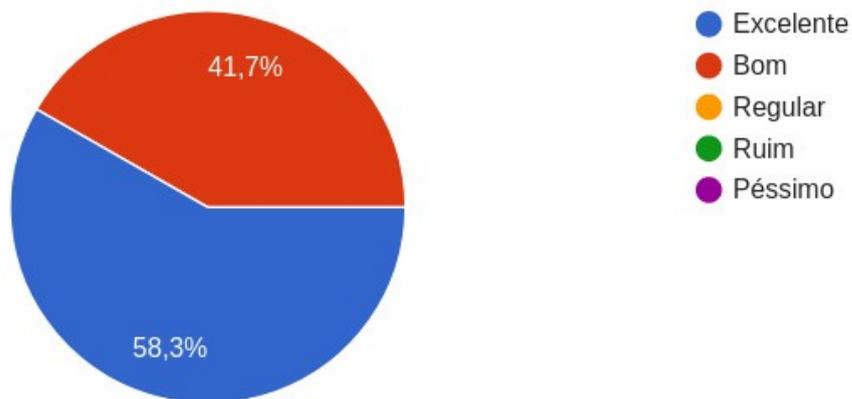
9. Como você avalia a interface do simulador?
10. Como você avalia a qualidade das animações do simulador?

Figura 25 – Avaliação da interface do simulador



Fonte: Google Forms

Figura 26 – Avaliação da qualidade das simulações



Fonte: Google Forms

Segundo as respostas dos usuários, a qualidade da interface e das simulações está de acordo com as expectativas, com grande maioria apresentando um retorno positivo aos questionamentos.

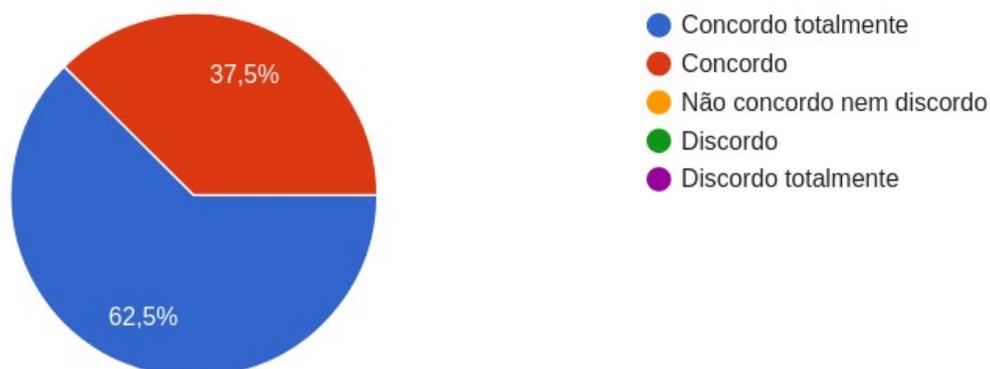
4.1.6 Sobre o auxílio no aprendizado

Os seguintes questionamentos foram apresentados para avaliar a opinião dos alunos acerca da utilização do simulador em sala de aula:

11. Você concorda com a frase abaixo? “Professores deveriam utilizar mais ferramentas de ensino como simuladores e experimentos em salas de aula”.

12. Na sua opinião, o simulador ajudou a fixar as informações sobre o conteúdo estudado?

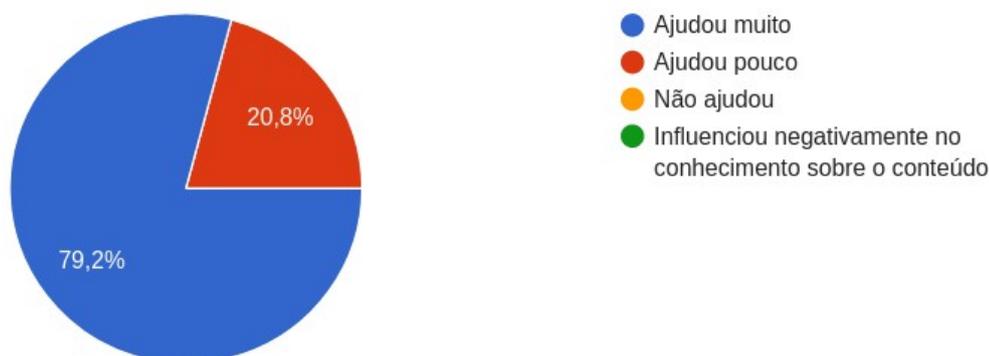
Figura 27 – Gráfico sobre utilização de simuladores em salas de aula.



Fonte: Google Forms

Percebe-se que todos os usuários concordam que professores deveriam utilizar mais ferramentas como a apresentada em salas de aula. Isso condiz com o relato do professor responsável pela turma, que informou que a maioria dos alunos demonstrou um interesse acentuado acerca do conteúdo apresentado.

Figura 28 – Gráfico sobre fixação do conteúdo



Fonte: Google Forms

Mesmo não sendo possível avaliar a eficácia do método utilizado através de uma pesquisa controlada, percebe-se que, na percepção dos alunos, o simulador se mostrou efetivo no auxílio da fixação dos conceitos estudados. O resultado condiz com alguns dos estudos encontrados, como o apresentado em Gerlič e S. Ülen (GERLIČ e ÜLEN, 2011), que demonstra um aumento no aproveitamento de alunos introduzidos a métodos conceituais, utilizando o computador como ferramenta de ensino.

4.2 MELHORIAS FUTURAS

Diversas melhorias podem ser implementadas para aprimorar a qualidade do *software* desenvolvido. Destas, algumas foram identificadas por necessidades dos usuários e outras foram encontradas já durante o processo de construção do protótipo, porém por questões de prazo e esforço de desenvolvimento não foram implementadas.

As principais melhorias identificadas durante o processo de desenvolvimento foram as seguintes:

1. Adição de novas simulações, representando diferentes fenômenos físicos;
2. Cadastro de turmas;
3. Interface de professor para cadastro de perguntas e controle de turmas;

4.2.1 Novas simulações

Para atender aos requisitos de projeto iniciais, somente a simulação de corpos em queda livre foi implementada porém, conforme apresentado na seção 2.2, um número muito grande de fenômenos físicos pode ser representado de maneira adequada por simuladores.

Talvez o próximo passo seja adicionar diferentes simulações de tipos de movimento em duas dimensões, como movimento retilíneo uniforme e uniformemente variado, lançamento horizontal e vertical, entre outros. Também podem ser introduzidos conceitos de outras forças externas, como atrito estático e dinâmico e resistência do ar.

Outro ponto a se considerar é a introdução de fenômenos diferentes da dinâmica de corpos, como eletricidade, magnetismo, gravitação, ótica, entre outros.

4.2.2 Cadastro de turmas e interface do professor

Atualmente, não é possível associar o usuário aluno a uma turma e, uma vez que o usuário deixa o sistema, suas informações só ficam armazenadas em memória até que o 'coletor de lixo' feche a conexão.

Uma solução interessante seria permitir que professores realizem o cadastro de turmas no sistema. Isso permitiria que alunos acessassem uma determinada turma (através de um código identificador único, por exemplo), e o professor teria acesso às respostas dos usuários e a seus resultados finais.

Também seria possível resolver o problema da inserção de novas perguntas no sistema. Hoje, o sistema gera as perguntas para cada aluno com base em um banco de

questões armazenado no servidor. O banco de questões atual é extremamente simples, armazenado em formato JSON, contendo alguns dados sobre a pergunta e sobre a simulação associada à mesma.

Figura 29 – Formato da pergunta no sistema

```
{
  "number": 1,
  "title": "Pergunta Um",
  "text": "Qual a velocidade final de um corpo de 80kg ao cair (do repouso) de uma altura de 2m na superfície terrestre? Considere  $g = 10\text{m/s}^2$ ?",
  "experimentData": {
    "gravity": 10,
    "planet": "earth",
    "initialVelocity": 0,
    "initialVelocityUn": "ms",
    "bodyMass": 80,
    "bodyMassUn": "kg",
    "fallHeight": 2
  },
  "options": ["6.33 m/s", "60.3 m/s", "603 m/s", "72 m/s", "2.1 m/s"]
}
```

Fonte: Autor

Com uma interface de professor, seria possível permitir que o mesmo criasse as perguntas e pré-configurasse as simulações, para atender as necessidades dos alunos e para melhor representar a dinâmica de cada sala de aula. Ao inserir uma nova questão, a nova interface geraria a pergunta no formato JSON (que já é lido pela Tela de Prova), não necessitando de alterações para integração.

4.2.3 Melhorias identificadas por usuários

Alguns usuários apresentaram sugestões de melhoria durante a utilização do protótipo. Os principais pontos levantados se destacam abaixo:

1. Melhoria na precisão das simulações;
2. Adicionar gráficos para visualizar as mudanças de posição e velocidade ao longo do tempo;
3. Adicionar uma 'linha do tempo', onde seja possível avançar e voltar no tempo para ver o estado do corpo (posição, velocidade e energia cinética) em determinado momento;
4. Permitir a 'ativação e desativação' da resistência do ar na simulação, de modo a avaliar a diferença na velocidade final de queda;
5. Adicionar diferentes tipos de colisão (elástica, inelástica e parcialmente elástica), para verificar como o corpo se comporta ao cair.

4.3 CONSIDERAÇÕES FINAIS

De modo geral, identificou-se que a interação dos usuários com o sistema foi positiva. O professor que aplicou o simulador relatou que o interesse dos alunos aumentou através da utilização do *software* e que alguns relataram inclusive utilizá-lo em casa como ferramenta de estudo extraclasse.

Por limitações de tempo e público não foi possível realizar uma pesquisa que avaliasse com precisão a eficácia do método de ensino auxiliado pelo simulador, porém por avaliação própria dos usuários, foi possível identificar que os mesmos consideraram que a interação foi positiva, com 100% dos respondentes indicando que o simulador ajudou pelo menos um pouco na fixação dos conceitos estudados.

Por fim, foi possível desenvolver um simulador utilizável, aplicando conceitos de engenharia de *software*, que atendeu aos requisitos de seus usuários. O diferencial da solução desenvolvida é que, além de ser de fácil utilização, a mesma proporciona tanto um recurso para auxílio no ensino do conteúdo estudado quanto uma ferramenta de avaliação, que pode também ser utilizada como método de fixação do conteúdo por meio de exercícios.

Para iterações futuras, foram levantadas melhorias que, se implementadas, tornariam o sistema ainda mais proveitoso.

REFERÊNCIAS

ABDULAYEVA, Aigerim. Using Smartphones in Home Education to Perform Physics Lab. **IEEE International Conference on Smart Information Systems and Technologies (SIST)**, [s. l.], p. 1-4, 2021.

BEN OUAHI, Mhamed et al. Opinions of Moroccan teachers towards the use of PhET simulations in teaching and learning physics – chemistry. **IEEE Congress on Information Science and Technology (CiSt)**, [s. l.], ano 2020, ed. 6, p. 274-278, 2020.

CETIC.BR; CGI.BR; NIC.BR. **Pesquisa sobre o uso das tecnologias de informação e comunicação nas escolas brasileiras**. [S. l.], 2018. Disponível em: <https://cetic.br/pt/tics/educacao/2018/escolas-urbanas-alunos>. Acesso em 30 jun 2022.

GERLIČ, Ivan; ÜLEN, Simon. The computer as a key component in the conceptual learning of physics. **Proceedings of the 22nd EAEEIE Annual Conference**, [s. l.], ano 2011, p. 1-5, 2011.

MDN WEBDOCS. **Referência JavaScript**: JavaScript | MDN. [S. l.], 2020. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference>. Acesso em: 4 jul. 2022.

NPM INC. **About npm**. [S. l.], [entre 2014 e 2022]. Disponível em: <https://docs.npmjs.com/about-npm>. Acesso em: 11 jul. 2022.

NPM INC. **Npm**: By the numbers. [S. l.], 2022c. Disponível em: <https://www.npmjs.com/>. Acesso em: 11 jul. 2022.

OPENJS FOUNDATION. **About**: Node.js. [S. l.], [entre 2009 e 2022]. Disponível em: <https://nodejs.org/en/about/>. Acesso em: 4 jul. 2022.

OPENJS FOUNDATION. **Express**: Node.js web application framework. [S. l.], 2017c. Disponível em: <https://expressjs.com/>. Acesso em: 11 jul. 2022.

PALANDRANI, Pedro. **A Decade of Change**: How Tech Evolved in the 2010s and What's In Store for the 2020s. [S. l.], 10 fev. 2020. Disponível em: <https://www.globalxetfs.com/a-decade-of-change-how-tech-evolved-in-the-2010s-and-whats-in-store-for-the-2020s/>. Acesso em: 30 jun. 2022.

RESNICK, Robert; HALLIDAY, David. **Física**. 4. ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos S.A., 1983.

SOCKET.IO. **Introduction**: Socket.IO. [S. l.], 2022c. Disponível em: <https://socket.io/docs/v4/>. Acesso em: 11 jul. 2022.

STACK OVERFLOW. **Stack Overflow Developer Survey**. [S. l.], 2022. Disponível em: <https://survey.stackoverflow.co/2022/>. Acesso em: 4 jul. 2022.

STATISTA. **Daily time spent with the internet per capita worldwide from 2011 to 2021, by device**. [S. l.], junho 2019. Disponível em: <https://www.statista.com/statistics/319732/daily-time-spent-online-device/>. Acesso em: 30 jun. 2022.

SUN, Jianhong. Solving Strategies Research for the Negative Impact of Computer Technology on Education. **International Workshop on Education Technology and Computer Science**, [s. l.], ano 2010, ed. 2, p. 671-674, 2010.

WAZLAWICK, Raul Sidnei. **Análise e projeto de sistemas de informação orientados a objetos**. 2. ed. rev. e atual. Rio de Janeiro: Elsevier, 2011. ISBN 978-85-352-3916-4.

WAZLAWICK, Raul Sidnei. **Engenharia de Software: Conceitos e Práticas**. 1. ed. Rio de Janeiro: Elsevier, 2013. ISBN 978-85-352-6084-7.

YE, Quianquian; MCCARTHY, Lauren Lee. **Home: p5.js**. [S. l.], [entre 2020 e 2022]. Disponível em: <https://p5js.org/>. Acesso em: 11 jul. 2022.

APÊNDICE A – DIAGRAMA DE CLASSES COMPLETO

