

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

THIAGO SILVEIRA BAUER

**Desenvolvimento de um Curso Online - Desenvolvendo um Jogo usando  
Python para o Ensino Médio**

FLORIANÓPOLIS

2024

THIAGO SILVEIRA BAUER

**Desenvolvimento de um Curso Online - Desenvolvendo um Jogo usando  
Python para o Ensino Médio**

Trabalho de Conclusão do Curso de Graduação em Sistemas de Informação, do Departamento de Informática e Estatística, do Centro Tecnológico da Universidade Federal de Santa Catarina, requisito parcial à obtenção do título de Bacharel em Sistemas de Informação.

Orientadora: Prof.<sup>a</sup> Dr.<sup>a</sup> rer. nat. Christiane Gresse von Wangenheim, PMP.

FLORIANÓPOLIS

2024

## RESUMO

O entendimento sobre algoritmos e programação para resolução de problemas tem se tornado cada vez mais importante para as sociedades que almejam formar estudantes na Educação Básica com as competências necessárias para enfrentar os impactos das transformações digitais. Porém, ainda existem poucas iniciativas de ensino de *Python* voltadas para ensinar esse conteúdo já no ensino médio. Assim, o objetivo deste trabalho é desenvolver um curso online para ensinar conceitos básicos de algoritmos e programação com *Python* voltado ao desenvolvimento de um *shooter game* utilizando a biblioteca Pygame. O curso é desenvolvido baseado na análise do estado da arte, adotando o *design* instrucional para a criação do curso, incluindo o plano de ensino, o material didático e a definição da avaliação da aprendizagem do aluno. O resultado pode ser utilizado por educadores e professores do Ensino Médio para ampliar o ensino desse conhecimento em escolas brasileiras.

**Palavras-chave:** Computação, Programação, Python, Algoritmos, Curso, Ensino Médio.

## SUMÁRIO

1. INTRODUÇÃO
    - 1.1. Contextualização
    - 1.2. Objetivos
    - 1.3. Metodologia de pesquisa e trabalho
    - 1.4. Estrutura do documento
  2. FUNDAMENTAÇÃO TEÓRICA
    - 2.1. Conceitos de ensino de computação no ensino médio
    - 2.2. Conceitos de Python e a biblioteca PyGame
    - 2.3. PyCharm
  3. ESTADO DA ARTE
    - 3.1. Definição da revisão
    - 3.2. Execução da Busca
    - 3.3. Análise dos Resultados
      - 3.3.1. Quais UIs existem?
      - 3.3.2. Quais conceitos de algoritmos e programação são ensinados nessas UIs?
      - 3.3.3. Quais são as características instrucionais dessas UIs?
    - 3.4. Discussão
  4. DESENVOLVIMENTO DE UM CURSO PYTHON NO ENSINO MÉDIO
    - 4.1. Análise de Contexto
    - 4.2. Objetivos de Aprendizagem
    - 4.3. Design do Curso
    - 4.4. Material Didático
  5. Conclusão
- REFERÊNCIAS

# 1 INTRODUÇÃO

## 1.1 Contextualização

O Pensamento Computacional se refere à capacidade de sistematizar, representar, analisar e resolver problemas. Apesar de ser um termo recente, vem sendo considerado como um dos pilares fundamentais do intelecto humano, junto com leitura, escrita e aritmética, pois assim como estes, serve para descrever, explicar e modelar o universo e seus processos complexos (SBC, 2017). Segundo o Ministério da Educação, grande parte das futuras profissões exigirá, direta ou indiretamente, computação e tecnologias digitais (MEC, 2017), mostrando-se essencial a abordagem e desenvolvimento a respeito do tema na educação básica, com intuito de formar cidadãos com os valores e habilidades necessárias para enfrentar os impactos das transformações digitais.

Em 2022 o Ministério da Educação homologou o Parecer NE/CEB 2/2022, que define o ensino de computação na educação básica de todo o país. No que diz respeito ao Ensino Médio, uma das premissas se refere ao desenvolvimento de projetos para investigar desafios do mundo contemporâneo, construir soluções e tomar decisões éticas, democráticas e socialmente responsáveis, articulando conceitos, procedimentos e linguagens próprias de computação, preferencialmente de forma colaborativa (MEC, 2022). Principalmente no ensino médio em relação a algoritmos e programação, de acordo com a Sociedade Brasileira de Computação, espera-se que o aluno desenvolva conhecimentos nos eixos de Pensamento Computacional, Mundo Digital e Cultura Digital (SBC, 2017). A elaboração de projetos de modelagem computacional envolvendo esse eixos é de relevância para os alunos trabalharem colaborativamente na busca de soluções para problemas de diversas áreas, compreendendo como a Computação pode ser usada tanto no seu dia a dia quanto no mundo do trabalho, provendo ferramentas e habilidades essenciais (SBC, 2017).

Nesse estágio escolar, considera-se o Python como uma boa opção para introduzir a programação para iniciantes, devido às facilidades oferecidas pela linguagem, como o desenvolvimento rápido de jogos e sintaxe simples (SOL SBC, 2011). Python é uma linguagem de programação poderosa e dinâmica (Python,

2023), que permite a construção de soluções para análise de dados, Inteligência Artificial e Machine Learning, além do desenvolvimento de aplicativos móveis e sistemas web e *desktop*. Por causa de sua sintaxe simplificada, grande parte dos cursos de algoritmos e programação para o nível iniciante utilizam Python para introduzir os conceitos para seus alunos. O Python também representa um alto nível de praticidade, exigindo um código consideravelmente menor para a mesma funcionalidade escrita em linguagens compiladas como C++ e Java, tornando-o mais fácil de arrumar, retrabalhar e otimizar (Python, 2023). O Python também tem uma extensa coleção de bibliotecas, o que a torna uma linguagem extensível, sendo fácil de adicionar novas capacidades e funcionalidades ao projeto sem grandes esforços, quando comparado a outras linguagens de programação (Python, 2023). Por esses e outros motivos, a linguagem se consolidou no mercado, sendo utilizada por gigantes como Google, Dropbox, Reddit, Youtube, entre outros e atualmente é a linguagem mais utilizada (Spectrum, 2022).

Existem diversos cursos online que, utilizando Python (Miranda, 2022; Diosdado, 2022), ensinam conceitos básicos de algoritmos e programação, como variáveis, funções, condicionais, estruturas de dados e repetições. No entanto, apenas uma pequena parcela é direcionado para o aprendizado no Ensino Médio. Além disso, a maioria dos cursos adota um sequenciamento de conteúdo ensinando um conceito atrás do outro, muitos sem um projeto como base, não deixando claro a aplicação que esses conhecimentos possuem na prática. Assim, poucos cursos apresentam os conceitos de programação baseados em projetos, porém tipicamente apresentando somente projetos simples e sem relevância social. Existem também vários tutoriais e cursos que ensinam o desenvolvimento de jogos com Python (FreeCodeCamp, 2019; Tech With Tim, 2023), variando de jogos 2D simples como ping-pong, snake e tetris, chegando até jogos mais elaborados com funcionalidades multijogador. Levando em consideração o elevado número de visualizações nos tutoriais citados no YouTube, pode-se inferir que há uma grande busca pelo tema de desenvolvimento de jogos em Python.

Assim, observa-se atualmente uma carência de cursos para o ensino de *Python* no ensino médio em escolas brasileiras que ensinam os conceitos de algoritmos e programação com base em projetos e de forma interdisciplinar. Portanto, este trabalho visa desenvolver um curso para ensinar Python no ensino

médio de forma interdisciplinar e dirigido por um projeto, que é o desenvolvimento de um jogo alinhado às metas de desenvolvimento de sustentabilidade das Nações Unidas (ONU, 2023). Espera-se que a existência de um curso estruturado dessa forma possa ajudar a preencher a lacuna observada no ensino de programação no ensino médio, promovendo uma abordagem mais prática e interdisciplinar. A integração dos conceitos de programação com a criação de um projeto significativo, como o desenvolvimento de um jogo, não só facilita a compreensão dos conceitos básicos, mas também torna o aprendizado mais envolvente e relevante para os alunos. Além disso, ao alinhar o projeto com as metas de desenvolvimento sustentável das Nações Unidas, o curso proporciona uma oportunidade para que os estudantes compreendam e contribuam para questões globais, ao mesmo tempo em que desenvolvem habilidades técnicas e de pensamento crítico.

Este trabalho pretende, portanto, oferecer um modelo de curso que não só ensina programação de forma eficaz, mas também estimula a consciência social e a criatividade dos estudantes. Acredita-se que, ao abordar a programação através de um projeto interativo e significativo, os alunos poderão se engajar mais profundamente com o material, desenvolver habilidades valiosas e estar mais preparados para os desafios futuros no campo da tecnologia e além.

## **1.2 OBJETIVOS**

### **Objetivo geral**

O objetivo geral deste projeto é o desenvolvimento de um curso online para ensinar *Python* em escolas brasileiras em nível de Ensino Médio. O curso é alinhado aos currículos de referência do ensino de computação da ACM/CSTA (Seehorn e Clayborn, 2017) e SBC (2017). O curso é projetado de forma que permita a sua adoção de forma interdisciplinar inserida em conteúdo programado do Ensino Médio, desenvolvendo um jogo e alinhado às metas de desenvolvimento de sustentabilidade das Nações Unidas.

Este projeto visa o design instrucional do curso, incluindo a análise do contexto, definição dos objetivos de aprendizagem, definição da estratégia

instrucional e a preparação dos recursos didáticos a serem utilizados durante a sua execução. Como resultado, é criado um plano de ensino, o material didático em forma de um curso online.

### **Objetivos específicos**

Os objetivos específicos são:

- O1. Elaborar a fundamentação teórica em relação a ensino de *computação* no ensino médio e a linguagem Python;
- O2. Analisar o contexto referente aos alunos, ambiente e currículos de referência e definir os objetivos de aprendizagem;
- O3. Levantar o estado da arte em relação a cursos semelhantes;
- O4. Definir o *design* instrucional do curso;
- O5. Desenvolver o material didático para o curso online.

### **1.3 Metodologia de pesquisa e trabalho**

A fim de alcançar os resultados esperados com este trabalho, é adotada uma combinação de metodologias de pesquisa de acordo com o respectivo objetivo a ser buscado. Então, de acordo com os objetivos específicos do projeto, são adotadas etapas da seguinte forma:

**Etapa 1 – Fundamentação teórica:** análise e síntese de conceitos básicos envolvidos no tema. São abordados conceitos de ensino de *computação* no ensino médio e sobre a linguagem Python e a biblioteca PyGame (PyGame, 2023). Este estudo é feito por meio de uma análise e síntese da literatura.

Atividade 1.1: Sintetizar conceitos de ensino de *computação* no ensino médio;

Atividade 1.2: Sintetizar conceitos de Python e a biblioteca PyGame, IDE PyCharm.

**Etapa 2 - Análise do contexto:** análise do contexto da unidade instrucional a ser desenvolvida, identificando características e restrições em relação ao público alvo, à infraestrutura e ao contexto educacional. É seguido o modelo de design instrucional ADDIE (BRANCH, 2009).

Atividade 2.1: Analisar o contexto em termos de necessidades e objetivos de aprendizagem alinhado às diretrizes dos currículos;

Atividade 2.2: Analisar o contexto em termos de perfil dos aprendizes e instrutores;

Atividade 2.3: Analisar o contexto em termos de ambiente em escolas brasileiras.

**Etapa 3 - Levantamento do estado da arte:** levantamento de trabalhos existentes relacionados ao tema do projeto. É realizado um estudo de mapeamento seguindo um processo proposto por Petersen et al. (2008; 2015) para identificar e analisar unidades instrucionais ensinando Python, atualmente sendo utilizadas e voltadas ao ensino de computação em ensino médio.

Atividade 3.1: Definir o protocolo de revisão;

Atividade 3.2: Executar a busca;

Atividade 3.3: Extrair e analisar as informações.

**Etapa 4 - Design da unidade instrucional:** engloba a parte de planejamento e design da unidade instrucional a ser realizada. A definição do design da unidade segue a metodologia de design instrucional ADDIE (BRANCH, 2009).

Atividade 4.1: Definir os objetivos de aprendizagem e sequenciar o conteúdo da unidade instrucional e definir uma estratégia instrucional, criando o plano de ensino/

Atividade 4.2: Definir a avaliação da aprendizagem do aluno.

**Etapa 5 - Desenvolvimento da unidade instrucional:** nesta etapa é realizado o desenvolvimento de todo o material didático para a aplicação do curso.

Atividade 5.1: Desenvolver o material de atividades online utilizando slides interativos;

Atividade 5.2: Desenvolver vídeos/animações interativos.

#### **1.4 Estrutura do documento**

No capítulo 2 é apresentado a fundamentação teórica, envolvendo os conceitos de ensino de computação no ensino médio abordando as perspectivas e competências exigidas pelos currículos respectivamente do MEC, SBC e CSTA, além das estratégias instrucionais e metodologias de ensino. Em seguida são apresentados os conceitos de Python, e da biblioteca PyGame, seus possíveis comandos e funcionalidades. No capítulo 3 é apresentado o mapeamento sistemático para análise do estado da arte em relação a unidades instrucionais voltadas ao ensino de Python para criar jogos no Ensino Médio. O capítulo 4 introduz a análise do contexto das Unidades Instrucionais selecionadas, e traz a proposta de plano de ensino, objetivos de aprendizagem, design do curso e artefatos criados. O capítulo 5 apresenta as conclusões e sugestões para trabalhos futuros.

## **2 FUNDAMENTAÇÃO TEÓRICA**

### **2.1 Conceitos de ensino de computação no ensino médio**

A constante transformação ocasionada pelas tecnologias, bem como sua repercussão na forma como as pessoas se comunicam, impacta diretamente no funcionamento da sociedade e, portanto, no mundo do trabalho. É preciso garantir aos jovens aprendizagens para atuar em uma sociedade em constante mudança, prepará-los para profissões que ainda não existem, para usar tecnologias que ainda não foram inventadas e para resolver problemas que ainda não conhecemos. Certamente, grande parte das futuras profissões envolverá, direta ou indiretamente, computação e tecnologias digitais (BNCC, 2017). A Base Nacional Comum Curricular, expressa sua preocupação com os impactos das transformações digitais na sociedade, a partir das diretrizes de currículo escolar no Brasil, e se explicita já nas competências gerais para a Educação Básica.

Referente ao pensamento computacional, a Base Nacional Comum Curricular, diz respeito às capacidades de compreender, analisar, definir, modelar, resolver, comparar e automatizar problemas e suas soluções, de forma metódica e sistemática, por meio do desenvolvimento de algoritmos. Os conhecimentos a respeito do mundo digital, que refere-se às aprendizagens relativas às formas de processar, transmitir e distribuir a informação de maneira segura, além de envolver a compreensão da importância de codificar, armazenar e proteger a informação. A cultura digital envolve aprendizagens voltadas a uma participação mais consciente e democrática por meio das tecnologias digitais, o desenvolvimento de uma postura crítica, ética e responsável, além da fluência no uso da tecnologia digital para expressão de soluções e manifestações culturais de forma contextualizada e crítica.

Já no ensino médio, considera-se imprescindível ampliar e aprofundar as competências desenvolvidas nas etapas anteriores, considerando a relação direta entre a cultura juvenil e a cultura digital, sabendo que jovens estão cada vez mais conectados e envolvidos com tecnologias e não somente como consumidores, mas também como agentes protagonistas. Portanto, na etapa do ensino médio, o foco passa a ser o reconhecimento do potencial das tecnologias para a realização de

atividades relacionadas a todas as áreas do conhecimento, incluindo o mundo do trabalho e também práticas sociais.

Outro aspecto considerado pela Base Nacional Comum Curricular refere-se a aprendizagem de aritmética, álgebra, geometria, estatística e probabilidade, pois além desses processos de aprendizagem serem potencialmente ricos para o desenvolvimento de competências fundamentais e letramento matemático, eles também são essenciais para o desenvolvimento do pensamento computacional dos jovens. Isso se dá pela necessidade do desenvolvimento da capacidade de traduzir uma determinada situação em diferentes linguagens, transformando problemas apresentados em língua materna, em representações diversas, como fórmulas, tabelas e gráficos, e vice-versa. Ainda associado à matemática, considera-se que os temas de algoritmos e fluxogramas devem ser objetos de estudo nas aulas, considerando que algoritmos possuem pontos em comum com a álgebra, principalmente no que se diz respeito ao conceito de variável. Outra habilidade em comum da álgebra e do pensamento computacional é a identificação de padrões para estabelecimento de generalizações, propriedades, e novamente, algoritmos.

No Ensino Médio, o foco é a continuidade dessas aprendizagens, construindo uma visão integrada da matemática que seja aplicada a realidade em diferentes contextos, levando em consideração as vivências individuais dos estudantes, que são impactados diretamente e de diferentes maneiras pelos avanços tecnológicos, mídias sociais e mercado de trabalho. Ainda nesse contexto, destaca-se a importância da utilização de tecnologias digitais para o desenvolvimento matemático e continuidade do desenvolvimento do pensamento computacional que foi iniciado na etapa do Ensino Fundamental.

Em fevereiro de 2022 o Ministério da Educação e Cultura emitiu um parecer que complementa a Base Nacional Comum Curricular, com intuito de estabelecer a importância e objetivos do ensino de Computação nas escolas, desde a educação infantil até o Ensino Médio. O MEC compreende que na etapa do ensino médio, devem ser desenvolvidas as seguintes competências:

1. Compreender as possibilidades e os limites da Computação para resolver problemas, tanto em termos de viabilidade quanto de eficiência, propondo e

analisando soluções computacionais para diversos domínios do conhecimento, considerando diferentes aspectos.

2. Analisar criticamente artefatos computacionais, sendo capaz de identificar as vulnerabilidades dos ambientes e das soluções computacionais buscando garantir a integridade, privacidade, sigilo e segurança das informações.
3. Analisar situações do mundo contemporâneo, selecionando técnicas computacionais apropriadas para a solução de problemas.
4. Construir conhecimento usando técnicas e tecnologias computacionais, produzindo conteúdos e artefatos de forma criativa, com respeito às questões éticas e legais, que proporcionem experiências para si e os demais.
5. Desenvolver projetos para investigar desafios do mundo contemporâneo, construir soluções e tomar decisões éticas, democráticas e socialmente responsáveis, articulando conceitos, procedimentos e linguagens próprias da Computação preferencialmente de maneira colaborativa.
6. Expressar e partilhar informações, ideias, sentimentos e soluções computacionais utilizando diferentes plataformas, ferramentas, linguagens e tecnologias da Computação de forma fluente, criativa, crítica, significativa, reflexiva e ética.
7. Agir pessoal e coletivamente com respeito, autonomia, responsabilidade, flexibilidade, resiliência e determinação, identificando e reconhecendo seus direitos e deveres, recorrendo aos conhecimentos da Computação e suas tecnologias frente às questões de diferentes naturezas.

Cada uma das competências descritas também exige determinadas habilidades, mostradas na Tabela 1.

Tabela 1. Competências exigidas pelo MEC

| <b>Objetivos</b>   | <b>Habilidades</b>  |
|--|---|
| Compreender as possibilidades e os limites da Computação para resolver problemas, tanto em termos de viabilidade quanto de eficiência, propondo e analisando soluções computacionais para diversos domínios do conhecimento, considerando diferentes aspectos. | (EM13CO01) Explorar e construir a solução de problemas por meio da reutilização de partes de soluções existentes.   |
|  | (EM13CO02) Explorar e construir a solução de problemas por meio de refinamentos, utilizando diversos níveis de abstração desde a especificação até a implementação. |
|  | (EM13CO03) Identificar o comportamento dos algoritmos no que diz respeito ao consumo de recursos como tempo de execução, espaço de memória e energia, entre outros  |

|  |   |
|--|---|
|  | (EM13CO04) Reconhecer o conceito de metaprogramação como uma forma de generalização na construção de programas, permitindo que algoritmos sejam entrada ou saída para outros algoritmos.  |
|  | (EM13CO05) Identificar os limites da Computação para diferenciar o que pode ou não ser automatizado, buscando uma compreensão mais ampla dos limites dos processos mentais envolvidos na resolução de problemas.                    |
|  | (EM13CO06) Avaliar software levando em consideração diferentes características e métricas associadas.   |
| Analisar criticamente artefatos computacionais, sendo capaz de identificar as vulnerabilidades dos ambientes e das soluções computacionais buscando garantir a integridade, privacidade, sigilo e segurança das informações.                                 | (EM13CO07) Compreender as diferentes tecnologias, bem como equipamentos, protocolos e serviços envolvidos no funcionamento de redes de computadores, identificando suas possibilidades de escala e confiabilidade.                  |
|  | (EM13CO08) Entender como mudanças na tecnologia afetam a segurança, incluindo novas maneiras de preservar sua privacidade e dados pessoais online, reportando suspeitas e buscando ajuda em situações de risco.                     |
| Analisar situações do mundo contemporâneo, selecionando técnicas computacionais apropriadas para a solução de problemas.   | (EM13CO09) Identificar tecnologias digitais, sua presença e formas de uso, nas diferentes atividades no mundo do trabalho.  |
|  | (EM13CO10) Conhecer os fundamentos da Inteligência Artificial, comparando-a com a inteligência humana, analisando suas potencialidades, riscos e limites.   |
|  | (EM13CO11) Criar e explorar modelos computacionais simples para simular e fazer previsões, identificando sua importância no desenvolvimento científico.   |
| Construir conhecimento usando técnicas e tecnologias computacionais, produzindo informação e/ou artefatos de forma criativa, com respeito às questões legais, que proporcionem experiências para si e os demais.   | (EM13CO12) Produzir, analisar, gerir e compartilhar informações a partir de dados, utilizando princípios de ciência de dados.   |
|  | (EM13CO13) Analisar e utilizar as diferentes formas de representação e consulta a dados em formato digital para pesquisas científicas.  |
|  | (EM13CO14) Avaliar a confiabilidade das informações encontradas em meio digital, investigando seus modos de construção e considerando a autoria, a estrutura e o propósito da mensagem.   |
|  | (EM13CO15) Analisar a interação entre usuários e artefatos computacionais, abordando aspectos da experiência do usuário e promovendo reflexão sobre a qualidade do uso dos artefatos nas esferas do trabalho, do lazer e do estudo. |
|  | (EM13CO16) Desenvolver projetos com robótica, utilizando artefatos físicos ou simuladores.  |
| Desenvolver projetos para investigar desafios do mundo contemporâneo, construir soluções e tomar decisões éticas, democráticas e socialmente responsáveis, articulando conceitos, procedimentos e linguagens próprias da Computação de maneira colaborativa. | (EM13CO17) Construir redes virtuais de interação e colaboração, favorecendo o desenvolvimento de projetos de forma segura, legal e ética.   |
|  | (EM13CO18) Planejar e gerenciar projetos integrados às áreas de conhecimento de forma colaborativa, solucionando problemas, usando diversos artefatos computacionais.   |
| Expressar e partilhar informações, ideias, sentimentos e soluções  | (EM13CO19) Expor, argumentar e negociar propostas, produtos e serviços, utilizando diferentes mídias e ferramentas digitais.  |

|  |   |
|--|---|
| computacionais utilizando diferentes plataformas, ferramentas, linguagens e tecnologias da Computação de forma fluente, criativa, crítica, significativa, reflexiva e ética.   | (EM13CO20) Criar conteúdos, disponibilizando-os em ambientes virtuais para publicação e compartilhamento, avaliando a confiabilidade e as consequências da disseminação dessas informações.                   |
|  | (EM13CO21) Comunicar ideias complexas de forma clara por meio de objetos digitais como mapas conceituais, infográficos, hipertextos e outros.   |
|  | (EM13CO22) Produzir e publicar conteúdo como textos, imagens, áudios, vídeos e suas associações, bem como ferramentas para sua integração, organização e apresentação, utilizando diferentes mídias digitais. |
| Agir pessoal e coletivamente com respeito, autonomia, responsabilidade, flexibilidade, resiliência e determinação, identificando e reconhecendo seus direitos e deveres, recorrendo aos conhecimentos da Computação e suas tecnologias para tomar decisões frente às questões de diferentes naturezas. | (EM13CO23) Analisar criticamente as experiências em comunidades virtuais e as relações advindas da interação e comunicação com outras pessoas, bem como seus impactos na sociedade.                           |
|  | (EM13CO24) Identificar e reconhecer como as redes sociais e artefatos computacionais em geral interferem na saúde física e mental de seus usuários.   |
|  | (EM13CO25) Dialogar em ambientes virtuais com segurança e respeito às diferenças culturais e pessoais, reconhecendo e denunciando atitudes abusivas.  |
|  | (EM13CO26) Aplicar os conceitos e pressupostos do direito digital em sua conduta e experiências com o cotidiano da cultura digital, bem como na produção e uso de artefatos computacionais.                   |

Os currículos do Ensino Médio são compostos pela formação geral básica em conjunto com os itinerários formativos (MEC, 2017). Esses itinerários envolvem 4 áreas do conhecimento:

**Linguagens e suas Tecnologias.** Compreende disciplinas como Língua Portuguesa, Artes, Língua Estrangeira Moderna, Educação Física, Tecnologias da Informação e Comunicação, entre outras. Foco nas linguagens, comunicação, expressão, produção artística e cultural.

**Matemática e suas Tecnologias.** Envolve disciplinas como Matemática, Estatística, Probabilidade e seus desdobramentos. Ênfase em raciocínio lógico, resolução de problemas, modelagem matemática e aplicações tecnológicas.

**Ciências da Natureza e suas Tecnologias.** Inclui disciplinas como Física, Química, Biologia e outras áreas relacionadas. Enfoque nas ciências naturais, experimentação, investigação científica, sustentabilidade e tecnologia.

**Ciências Humanas e Sociais Aplicadas.** Engloba disciplinas como História, Geografia, Filosofia, Sociologia, Psicologia, entre outras. Ênfase nas ciências humanas, sociais, políticas, culturais, econômicas e questões contemporâneas.

Além desses itinerários, é importante ressaltar que os estudantes também têm a possibilidade de realizar uma formação técnica e profissionalizante, por meio da oferta de cursos técnicos integrados ao Ensino Médio. Essa flexibilização curricular permite que os estudantes escolham os itinerários formativos de acordo com suas afinidades e interesses, permitindo uma formação mais personalizada e voltada para suas vocações. Os itinerários são oferecidos pelas escolas de forma a garantir a formação básica comum e aprofundamentos específicos. Dessa forma, o novo Ensino Médio permite a adoção interdisciplinar de um curso de algoritmos e programação como proposto por este trabalho.

Já a Sociedade Brasileira de Computação define os principais conceitos a serem trabalhados no Ensino Médio nos eixos Pensamento Computacional, Mundo Digital e Cultura Digital (Figura 1).

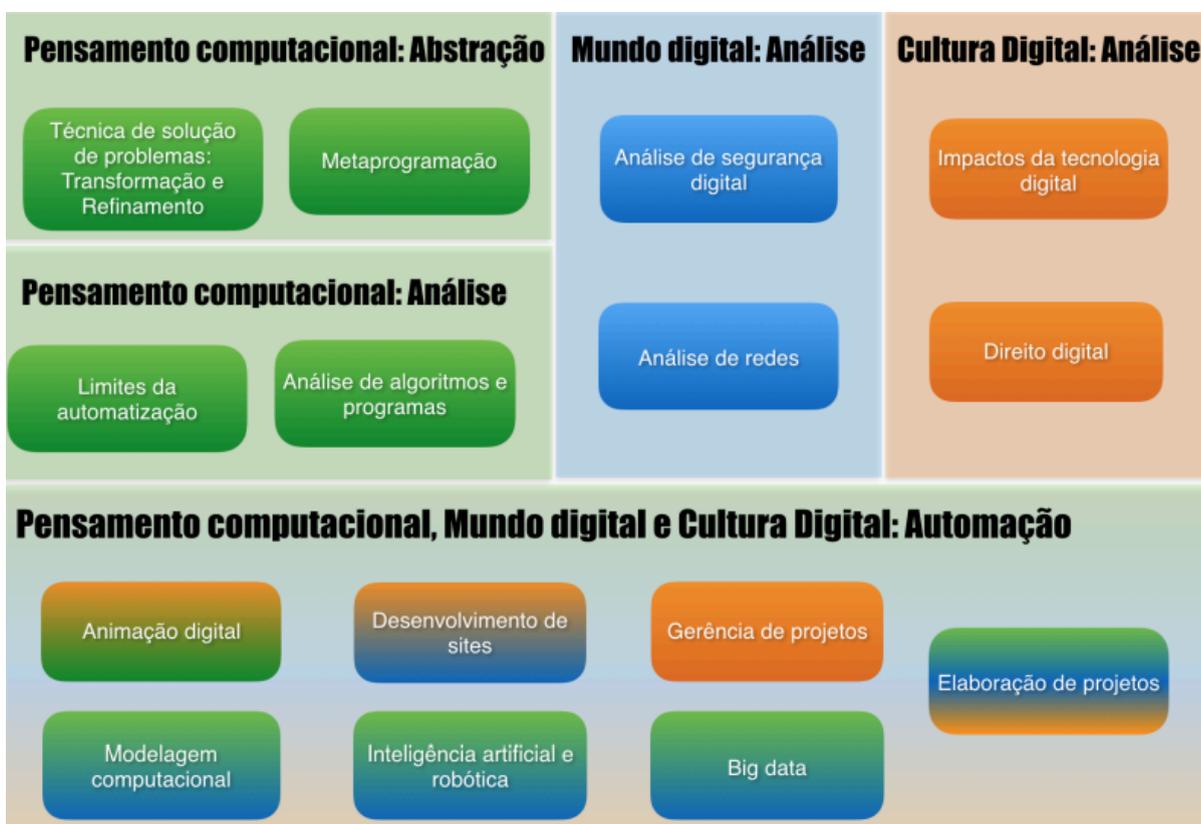


Figura 1. Conceitos a serem trabalhados no ensino médio, nos eixos de Pensamento Computacional, Mundo Digital e Cultura Digital (SBC, 2017)

No Ensino Médio a ênfase é na elaboração de projetos aplicando os conhecimentos e habilidades adquiridos na etapa do Ensino Fundamental, e no desenvolvimento de habilidades relacionadas à análise crítica e argumentação, sob diferentes aspectos (SBC, 2017). Na Tabela 2, a descrição dos objetos de conhecimento e habilidades da Computação no Ensino Médio (SBC, 2017) são apresentados.

Tabela 2. Descrição dos objetos de conhecimento de computação no Ensino Médio (SBC, 2017)

| <b>Objeto de conhecimento</b>                         | <b>Habilidades</b>   |
|---|--|
| <b>Técnica de solução de problemas: Transformação</b> | Compreender a técnica de solução de problemas através de transformações: comparar problemas para reusar soluções.  |
| <b>Técnica de solução de problemas: Refinamento</b>   | Compreender a técnica de solução de problemas através de refinamentos: utiliza diversos níveis de abstração no processo de construção de soluções.   |
| <b>Avaliação de algoritmos e programas</b>            | Analisar algoritmos quanto ao seu custo (tempo, espaço, energia, ...) para justificar a adequação das soluções a requisitos e escolhas entre diferentes soluções.  |
|   | Argumentar sobre a correção de algoritmos, permitindo justificar que uma solução de fato resolve o problema proposto   |
|   | Avaliar programas e projetos feitos por outras equipes com relação a qualidade, usabilidade, facilidade de leitura, questões éticas, etc.  |
| <b>Metaprogramação</b>                                | Reconhecer o conceito de metaprogramação como uma forma de generalização, que permite que algoritmos tenham como entrada (ou saída) outros algoritmos.   |
| <b>Limites da computação</b>                          | Entender os limites da Computação para diferenciar o que pode ou não ser mecanizado, buscando uma compreensão mais ampla dos processos mentais envolvidos na resolução de problemas.   |
| <b>Modelagem computacional</b>                        | Criar modelos computacionais para simular e fazer previsões sobre diferentes fenômenos e processos.  |
| <b>Inteligência artificial e robótica</b>             | Compreender os fundamentos da inteligência artificial e da robótica  |
| <b>Análise de redes</b>                               | Avaliar a escalabilidade e confiabilidade de redes, compreendendo as noções dos diferentes equipamentos envolvidos (como roteadores, <i>switchs</i> , etc) bem como de topologia, endereçamento, latência, banda, carga, delay |
| <b>Análise de segurança digital</b>                   | Comparar medidas de segurança digital, considerando o equilíbrio entre usabilidade e segurança   |
| <b>Big data</b>                                       | Entender o conceito de Big Data e utilizar ferramentas para representar, manipular e visualizar dados massivos   |
| <b>Desenvolvimento de sites</b>                       | Criar e manter sites e blogs com conteúdo individual e/ou coletivo   |
| <b>Animação digital</b>                               | Produzir animações digitais  |
| <b>Impactos da tecnologia digital</b>                 | Analisar e refletir sobre o tempo de vivência online, em jogos, em redes sociais, dentre outros  |

|                               |   |
|-------------------------------|---|
|                               | Reconhecer a influência dos avanços tecnológicos no surgimento de novas atividades profissionais  |
| <b>Direito digital</b>        | Compreender o direito digital e suas relações com o cotidiano do universo digital   |
| <b>Gerência de projetos</b>   | Gerenciar projetos digitais colaborativos usando computação em nuvem  |
| <b>Elaboração de projetos</b> | Elaborar e executar projetos integrados às áreas de conhecimento curriculares, em equipes, solucionando problemas, usando computadores, celulares, e outras máquinas processadoras de instruções. |

Dessa forma, o curso de Python proposto neste trabalho oferece a oportunidade de desenvolver diversos conhecimentos recomendados pela SBC. Entre eles, destacam-se a compreensão dos impactos da tecnologia digital e o desenvolvimento do pensamento computacional, especialmente por meio da habilidade de solucionar problemas utilizando a reutilização e o refinamento de soluções. Isso é alcançado pela elaboração de um projeto de jogo completo, que aplica técnicas de resolução de problemas e aprimoramento de soluções existentes.

Por outro lado, a *Computer Science Teachers Association* (CSTA) desenvolveu diretrizes conceituais para o ensino de ciências da computação. O *Framework K-12 para Ciências da Computação* foi desenvolvido por especialistas em educação e computação, servindo como um guia de alto nível que pode ser usado para o desenvolvimento de currículos personalizados (CSTA, 2017). A Tabela 3 apresenta os conceitos fundamentais e conceitos de computação definidos pela CSTA.

Tabela 3. Conceitos fundamentais CSTA (2017)

| <b>Conceitos fundamentais</b> | <b>Subconceitos</b>                                      | <b>Visão geral</b>   |
|-------------------------------|--|--|
| Sistemas de Computação        | Dispositivos; Hardware e Software; Solução de problemas. | As pessoas interagem com uma grande variedade de dispositivos de computação que coletam, armazenam, analisam e agem de acordo com as informações de forma que possam afetar positivamente e negativamente as capacidades humanas. Os componentes físicos (hardware) e instruções (software) que compõem um sistema de computação comunicam e processam informações em formato digital. A compreensão de hardware e software é útil para a solução de problemas de um sistema de computação que não funciona como pretendido. |
| Redes e Internet              | Comunicação e Organização da Rede; Cibersegurança        | Os dispositivos de computação normalmente não operam isoladamente. As redes conectam dispositivos de computação para compartilhar informações e recursos e são uma parte cada vez mais integral da computação. As redes e os sistemas de comunicação proporcionam maior conectividade no mundo da computação,  |

|                          |  |   |
|--------------------------|--|---|
|                          |  | fornecendo comunicação rápida e segura e facilitando a inovação.  |
| Dados e Análise          | Coleta, Visualização e Transformação; Armazenamento; Inferência e Modelos.   | Existem sistemas de computação para processar dados. A quantidade de dados digitais gerados no mundo está se expandindo rapidamente, de modo que a necessidade de processar dados de forma eficiente é cada vez mais importante. Os dados são coletados, armazenados e analisados para entender melhor o mundo e fazer previsões mais precisas.   |
| Algoritmos e Programação | Algoritmos; Variáveis; Controle; Modularidade; Desenvolvimento de Programas. | Um algoritmo é uma sequência de etapas projetada para realizar uma tarefa específica. Algoritmos são traduzidos em programas, ou código, para fornecer instruções para dispositivos de computação. Algoritmos e programação controlam todos os sistemas de computação, capacitando as pessoas a se comunicar com o mundo de novas maneiras e resolver problemas convincentes. O processo de desenvolvimento para criar programas significativos e eficientes envolve escolher quais informações usar e como processá-las e armazená-las, separando grandes problemas em pequenos, recombinação de soluções existentes e analisando diferentes soluções. |
| Impactos da computação   | Cultura; Interações sociais; Leis de segurança e ética.                      | A computação afeta muitos aspectos do mundo de maneiras positivas e negativas nos níveis local, nacional e global. Indivíduos e comunidades influenciam a computação através de seus comportamentos e interações culturais e sociais e, por sua vez, a computação influencia novas práticas culturais. Uma pessoa informada e responsável deve compreender as implicações sociais do mundo digital, incluindo a equidade e o acesso à computação.   |

As práticas fundamentais da ciência da computação descrevem os comportamentos e modos de pensar que os alunos alfabetizados computacionalmente usam para participar plenamente no mundo, rico em dados e interconectado de hoje (CSTA, 2017). A Tabela 4 representa as práticas fundamentais e suas respectivas descrições, indicadas pela CSTA.

Tabela 4. Descrição das práticas fundamentais CSTA (2017)

| Práticas fundamentais                            | Descrição  |
|--|--|
| 1. Promover uma cultura de computação inclusiva  | Incluir perspectivas dos outros; Atender as necessidades de diversos usuários; Apoio pessoal e defesa dos colegas.   |
| 2. Colaborar em torno da Computação              | Cultivar relações de trabalho; Criar normas de equipe, expectativas e cargas de trabalho equitativas; Solicitar e incorporar comentários e fornecer feedback; Avaliar e selecionar as ferramentas tecnológicas |
| 3. Reconhecer e definir problemas computacionais | Decompor problemas complexos em subproblemas; avaliar a viabilidade de resolver um problema computacionalmente.  |
| 4. Desenvolver e usar abstrações                 | Extrair características comuns de processos ou fenômenos; Avaliar e incorporar funcionalidades tecnológicas; Modelar e simular fenômenos.  |
| 5. Criar artefatos computacionais                | Planejar, criar e modificar um artefato computacional;   |
| 6. Testar e refinar artefatos                    | Testar artefatos; Identificar e corrigir erros; Avaliar e refinar um   |

|                                 |   |
|---------------------------------|---|
| Computacionais                  | artefato.   |
| 7. Comunicar sobre a computação | Selecionar, organizar e interpretar conjuntos de dados; Descrever, justificar e documentar soluções computacionais; Articular ideias. |

Também é detalhada pela CSTA os objetivos de aprendizagem de computação para o Ensino Médio (Tabela 5). Levando em consideração o foco do presente trabalho na parte de algoritmos e programação são apresentados na Tabela 5 somente os objetivos ligados ao conceito de algoritmos e programação.

Tabela 5. Objetivos de aprendizagem de computação no Ensino Médio CSTA, 2017)

| ID       | Objetivo de aprendizagem   |
|----------|--|
| 3A-DA-13 | Criar protótipos que utilizem algoritmos para resolver problemas computacionais alavancando conhecimentos e interesses pessoais anteriores   |
| 3A-AP-14 | Usar listas para simplificar soluções, generalizando problemas computacionais ao invés de usar repetidamente variáveis simples.  |
| 3A-AP-15 | Justificar a seleção de estruturas de controle específicas quando as compensações envolvem implementação, legibilidade e desempenho do programa e explicam os benefícios e desvantagens das escolhas feitas. |
| 3A-AP-16 | Projetar e desenvolver iterativamente artefatos computacionais para intenção prática, expressão pessoal ou para resolver uma questão social, usando eventos para iniciar instruções                          |
| 3A-AP-17 | Decompor problemas em componentes menores através de análises sistemáticas, usando construções como procedimentos, módulos e / ou objetos.   |
| 3A-AP-18 | Criar artefatos usando procedimentos dentro de um programa, combinações de dados e procedimentos, ou programas independentes, mas inter relacionados   |
| 3A-AP-19 | Conceber e desenvolver programas sistematicamente para grandes públicos, incorporando feedback dos usuários  |
| 3A-AP-20 | Avaliar as licenças que limitam ou restringem o uso de artefatos computacionais ao usar recursos como bibliotecas.   |
| 3A-AP-21 | Avaliar e aperfeiçoar artefatos computacionais para torná-los mais utilizáveis e acessíveis.   |
| 3A-AP-22 | Projetar e desenvolver artefatos computacionais trabalhando em equipe, usando ferramentas colaborativas  |
| 3A-AP-23 | Documentar decisões de projeto usando texto, gráficos, apresentações e / ou demonstrações no desenvolvimento de programas complexos.   |

Sendo assim, a partir do desenvolvimento do projeto proposto por este trabalho, será possível atingir alguns dos objetivos propostos pela CSTA, como

avaliar e aperfeiçoar artefatos computacionais para torná-los mais utilizáveis e acessíveis, além de criar protótipos que utilizam algoritmos para resolver problemas computacionais alavancando conhecimentos e interesses pessoais dos estudantes.

## 2.2 Conceitos de Python e a biblioteca PyGame

Python é uma linguagem de programação desenvolvida sob uma licença de código aberto aprovada pela OSI, que a torna livremente utilizável e distribuível mesmo para uso comercial (Python, 2023). É uma linguagem interpretada de alto-nível e que foi lançada pela primeira vez em 1991, que ganhou popularidade devido à sua simplicidade, facilidade de uso e flexibilidade, além de ser considerada uma linguagem poderosa e rápida, amigável e de fácil aprendizado tanto para programadores experientes em outras linguagens quanto para programadores iniciantes. O Python suporta diversos paradigmas de programação, entre eles programação orientada a objetos, funcional e procedural.

O Python atualmente é a linguagem de programação mais utilizada, incluindo em grandes e críticos sistemas (TIOBE, 2024) (Tabela 6).

Tabela 6. Linguagens de programação mais utilizadas em Setembro de 2024 (TIOBE, 2024)

| Posição em Setembro 2024 | Linguagem | Porcentagem de uso |
|--------------------------|-----------|--------------------|
| 1                        | Python    | 20.17%             |
| 2                        | C++       | 10.75%             |
| 3                        | Java      | 9.45%              |
| 4                        | C         | 8.89%              |
| 5                        | C#        | 6.08%              |

O Python é uma linguagem de tipagem dinâmica, o que significa que as variáveis não precisam ser declaradas com um tipo específico de dados, pois esse tipo é inferido em tempo de execução, o que permite maior flexibilidade e facilidade, porém exige uma atenção maior aos tipos de dados ao se escrever um código. Python também inclui uma grande quantidade de comandos:

**Comandos de entrada e saída (E/S):** O Python fornece uma variedade de comandos que podem ser usados para leitura de entrada do usuário, impressão de

saída no console, além de leitura e gravação de dados em arquivos. Os comandos de E/S mais comuns são `input()`, `print()`, `open()`, `close()`, `read()` e `write()`.

**Comandos de declarações condicionais:** Muito comum ser utilizado em blocos de *if/else*, o uso de declarações condicionais permite que diferentes blocos de código sejam executados com base em determinadas condições. Esses blocos condicionais geralmente são usados com operadores de comparação como “==” (igual a), “>” (maior que) e “<” (menor que) (Tabela 7).

Tabela 7. Exemplos de comandos de declarações condicionais em Python.

|   |  |
|---|--|
| <pre>#exemplo de código Python  valor1 = 10 valor2 = 20  if valor1 &gt; valor2:     print('O valor1 é maior do que valor2') else:     print('O valor2 é maior do que valor1')</pre> <pre>O valor2 é maior do que valor1</pre> | <p>Neste caso, considerando que o valor 2 é maior que o valor 1, o bloco 'else' será executado</p> |
|---|--|

**Funções:** São blocos de código que são definidos para realizar uma tarefa específica. Elas são usadas para dividir o código em partes menores e mais gerenciáveis, e também para reutilizar o código em diferentes partes do programa. O uso de funções em Python traz diversas vantagens para o desenvolvimento de software, como modularidade, reutilização de código e legibilidade. Para definir uma função em Python, utiliza-se a palavra-chave *'def'* seguida pelo nome da função e seus argumentos, se houver (Tabela 8):

Tabela 8. Exemplo de função em Python.

```
def soma(a, b):  
    return a + b  
  
resultado = soma(2, 3)  
print(resultado)
```

Neste exemplo, a função soma recebe dois argumentos, a e b, e retorna a soma desses valores. A função soma é chamada com os argumentos 2 e 3, e o resultado é armazenado na variável resultado. O resultado é então impresso na tela usando a função print(). Neste caso, a saída da função seria o número 5, considerando que 2 + 3 é igual a 5.

**Estruturas de repetição:** As estruturas de repetição *'while'* e *'for'* são estruturas de controle de fluxo em Python que permitem que um conjunto de instruções seja executado várias vezes. A repetição *'while'* é usada quando não se sabe quantas vezes um bloco de instruções deve ser executado. Ela continua executando o bloco de instruções enquanto a condição especificada for verdadeira (Tabela 9).

Tabela 9. Exemplo de estrutura de repetição *'while'* em Python.

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

Neste exemplo, o bloco de instruções é executado enquanto 'i' for menor que 5. A cada iteração, o valor de 'i' é impresso na tela e incrementado em 1

A repetição *'for'* é usada quando se sabe quantas vezes um bloco de instruções deve ser executado. Ela é mais útil quando se trabalha com estruturas de dados como listas, tuplas e dicionários (Tabela 10).

Tabela 10. Exemplo de estrutura de repetição *'for'* em Python.

```
frutas = ['maçã', 'banana', 'laranja']  
for fruta in frutas:  
    print(fruta)
```

Neste exemplo, a variável 'fruta' assume cada um dos valores da lista 'frutas' a cada iteração do loop. O valor de 'fruta' é então impresso na tela.

**Comandos de estruturas de dados:** O Python fornece uma variedade de comandos de estruturas de dados para trabalhar com diferentes tipos de dados. Esses comandos incluem listas, tuplas, dicionários e conjuntos, que podem ser utilizados para armazenar e manipular dados, sendo muito útil para criação de programas complexos.

**Comandos matemáticos:** O Python fornece uma variedade de comandos matemáticos que podem ser usados para executar operações matemáticas simples e complexas, incluindo operações aritméticas, funções trigonométricas, funções logarítmicas e muito mais.

**Manipulação de *strings*:** Com esses comandos, é possível manipular e formatar strings para atender às suas necessidades específicas. Esses comandos incluem concatenação, divisão, formatação e muito mais.

**Comandos orientados a objetos:** Para a programação orientada a objetos, o Python também fornece comandos específicos para trabalhar com objetos e classes. Esses comandos incluem criar objetos, definir classes e acessar propriedades e métodos de objetos.

O Python se torna ainda mais poderoso devido ao amplo conjunto de bibliotecas existentes (Python, 2023). As bibliotecas são coleções de códigos pré-escritos que podem ser usados para executar tarefas específicas, como manipulação e visualização de dados e aprendizado de máquina. Algumas bibliotecas populares em Python incluem NumPy, Plotly, Pydot e Matplotlib. Além das bibliotecas o Python existem também *frameworks* como Flask, um micro *framework* web escrito em Python que é usado para construção de sites, aplicativos web e APIs.

Python também está sendo amplamente utilizado para criação de jogos 2D e 3D, utilizando diferentes bibliotecas específicas para o desenvolvimento de *videogames*, como por exemplo a biblioteca Pygame (PyGame, 2023). O PyGame permite a criação de jogos completos e programas multimídia, sendo altamente portátil e compatível com quase todas as plataformas e sistemas operacionais, já tendo sido baixado milhões de vezes (PyGame, 2023). PyGame é gratuito, e possui mais de 660 projetos de jogos já publicados, incluindo finalistas de festivais e

eventos de jogos (PyGame, 2023). Principalmente, PyGame é fácil de usar e muito utilizado por crianças e adultos para criação de jogos de tiro, e tem sido ensinado em cursos para crianças e estudantes universitários (PyGame, 2023). A biblioteca fornece uma vasta opção de comandos e funcionalidades para criação de jogos simples e complexos.

### **Inicialização:**

`pygame.init()`: Inicializa todos os módulos do PyGame.

`pygame.display.set_mode()`: Cria uma janela de exibição para o jogo.

### **Desenho e cores:**

`pygame.draw.rect()`: Desenha um retângulo.

`pygame.draw.circle()`: Desenha um círculo.

`pygame.draw.line()`: Desenha uma linha.

`pygame.draw.polygon()`: Desenha um polígono.

`pygame.Color()`: Cria cores personalizadas.

### **Imagens e Sprites:**

`pygame.image.load()`: Carrega uma imagem.

`pygame.image.get_rect()`: Obtém um retângulo para uma imagem.

`pygame.sprite.Sprite()`: Cria um objeto sprite.

### **Eventos e entradas:**

`pygame.event.get()`: Obtém a lista de eventos.

`pygame.KEYDOWN` e `pygame.KEYUP`: Eventos de teclado.

`pygame.MOUSEBUTTONDOWN` e `pygame.MOUSEBUTTONUP`: Eventos de mouse.

### **Sons e música:**

`pygame.mixer.Sound()`: Carrega um arquivo de som.

`pygame.mixer.music.load()`: Carrega um arquivo de música.

`pygame.mixer.music.play()`: Reproduz a música.

### **Temporização:**

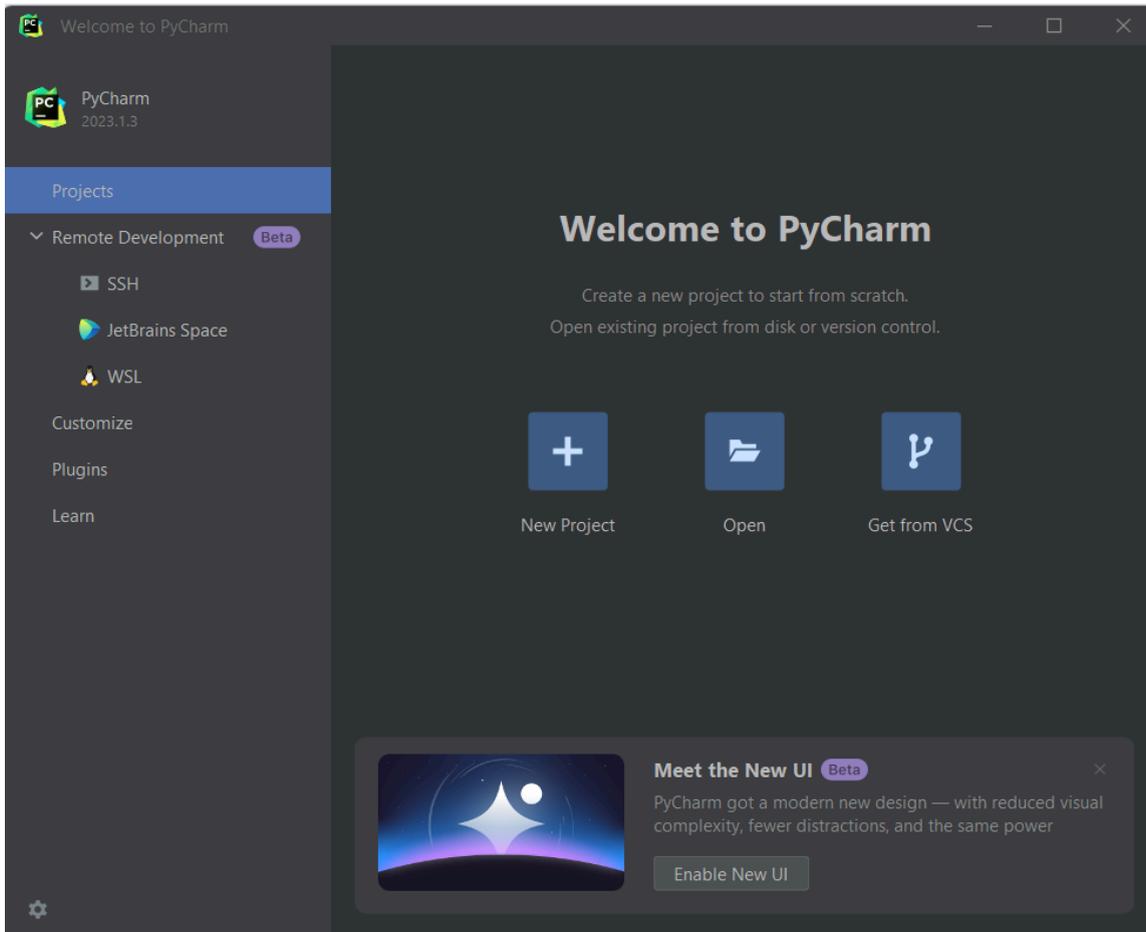
`pygame.time.Clock()`: Controla a taxa de atualização do jogo.

`pygame.time.get_ticks()`: Retorna o tempo decorrido em milissegundos.

## **2.3 PyCharm**

Para o desenvolvimento de sistemas com Python, existem diversas IDEs que suportam a linguagem. Porém, existem IDEs desenvolvidas com o objetivo de facilitar o uso do Python e suas bibliotecas, como é o caso do PyCharm (Jetbrains, 2023). O PyCharm oferece preenchimento de código inteligente, verificação dinâmica de erros, correções rápidas, assistência para testes, facilita a navegação pelo projeto e muito mais (Jetbrains, 2023). O PyCharm também possui uma versão gratuita, e oferece suporte específico para os frameworks de desenvolvimento web mais modernos, como Django, Flask, Google App Engine, Pyramid, entre outros.

Figura 2. Screenshot do PyCharm 2023.



Observa-se que o PyCharm facilita a instalação e utilização tanto do Python quanto do PyGame, além de permitir que seja gerado um executável do jogo, que poderá ser compartilhado pelos alunos, servindo como motivador adicional para o curso.

### 3. ESTADO DA ARTE

Visa-se a análise do estado da arte em relação a unidades instrucionais voltadas ao ensino de Python para criar jogos no Ensino Médio nos últimos 5 anos. Para levantar o estado da arte foi conduzido um mapeamento sistemático seguindo os procedimentos propostos por Petersen et al. (2008, 2015).

#### 3.1 Definição da revisão

A pergunta de pesquisa é: Quais unidades instrucionais existem para ensinar conceitos de Python por meio de criação de jogos no contexto do Ensino Médio publicados entre 2018 e 2023? Essa pergunta de pesquisa é refinada nas seguintes questões de análise:

AQ1. Quais unidades instrucionais existem?

AQ2. Quais conceitos de Python são ensinados nessas UIs?

AQ3. Quais são as características instrucionais dessas UIs?

AQ4. Como cada UI foi desenvolvida e como a sua qualidade foi avaliada?

**Critérios de inclusão/exclusão.** Foram consideradas quaisquer unidades instrucionais (curso, atividade, tutorial, etc.) que tivessem foco no ensino de Python por meio de jogos do Ensino Médio, publicadas entre 2018 e 2023. Unidades instrucionais que focam no ensino de Python para alunos do ensino superior, ensino infantil ou fundamental, e/ou que abordassem o ensino de computação sem aprofundar-se em conceitos de Python foram excluídas. São excluídas unidades instrucionais não voltadas à criação de jogos. Também foram excluídas publicações em forma de resumos, blogs, vídeos, ferramentas que não apresentam uma unidade instrucional ou não disponíveis gratuitamente.

**Critérios de qualidade.** Foram considerados apenas artigos ou materiais com informações suficientes relacionadas a conceitos de Python, indicando, por exemplo, conteúdo de aulas, materiais de apoio, etc.

**Fontes dos dados.** Foram examinados todos os materiais e artigos publicados em português e inglês, e disponíveis nas mais importantes bibliotecas

digitais acessíveis por meio da busca no Scopus no portal Capes. O Google também foi incluído, por indexar um grande conjunto de dados de diferentes fontes já que nessa área emergente muitas UIs não foram publicadas em forma de artigo científico.

**Definição da *string de busca*.** A string de busca foi composta de conceitos relacionados à questão de pesquisa, incluindo sinônimos (Tabela 11).

Tabela 11. Termos e sinônimos

| Termo chave    | Sinônimo          | Tradução                            |
|----------------|-------------------|-------------------------------------|
| Python         |                   | Python                              |
| Jogo           |                   | Game                                |
| Curso          | Tutorial, oficina | Course, tutorial, workshop          |
| “Ensino Médio” | Jovem, Criança    | “High School”, teen , kid, children |

A partir disso foram montadas as strings de busca, uma versão em inglês para buscas no Scopus e Google e uma versão em português para uma busca no Google. A string de busca foi adaptada para cada fonte de dados apresentada na Tabela 12.

Tabela 12. Search Strings.

| Fonte  | Search String   |
|--------|---|
| Scopus | ( TITLE-ABS-KEY ( python ) AND TITLE-ABS-KEY ( game ) AND TITLE-ABS-KEY ( course OR tutorial OR workshop ) AND TITLE-ABS-KEY ( "High School" OR teen OR kid OR children ) ) AND PUBYEAR > 2018 AND PUBYEAR > 2018 |
| Google | Python Jogo Curso “Ensino Médio”  |
|        | Python Game Course “High School”  |

### 3.2 Execução da Busca

A pesquisa foi realizada em maio de 2023 pelo autor e revisada pela orientadora (Tabela 13). Algumas pesquisas retornaram muitos resultados mesmo depois da adaptação das strings de busca, devido ao fato de que artigos que

descrevem a utilização de jogos no ensino utilizam os mesmos termos dos artigos que se referem ao desenvolvimento de jogos para o ensino.

Tabela 13. Resultados da busca

|              | Resultados da busca | Resultados analisados | Resultados potencialmente relevantes | Resultados relevantes |
|--------------|---------------------|-----------------------|--------------------------------------|-----------------------|
| Scopus       | 6                   | 6                     | 4                                    | 0                     |
| Google 1     | 1.100               | 200                   | 15                                   | 1                     |
| Google 2     | 10.400              | 200                   | 12                                   | 1                     |
| <b>Total</b> |                     |                       |                                      | <b>2</b>              |

Na fase inicial da análise, títulos e resumos foram verificados. Nas plataformas de pesquisa científica, diversos artigos foram descartados por focar na aplicação de jogos no contexto da educação e não focar no desenvolvimento de um jogo a partir do desenvolvimento e aplicação de material instrucional. O foco na utilização de Python para o ensino de outras matérias, como matemática e física, e não no contexto de algoritmos e programação, é outra razão pela qual muitos trabalhos foram descartados. Também foram excluídos, a partir da pesquisa no Google, artigos superficiais, assim como cursos não disponíveis gratuitamente. Na análise inicial, foram identificados 31 artefatos potencialmente relevantes. Posteriormente, todos os documentos foram lidos integralmente de maneira que fossem assegurados os critérios de inclusão e exclusão. Unidades Instrucionais claramente limitadas a estágios educacionais que não incluem o Ensino Médio, ou desenvolvimento de jogos em Python, foram excluídas. Algumas UIs foram descartadas por não conter detalhes da estratégia instrucional.

### 3.3 Análise dos Resultados

#### 3.3.1 Quais UIs existem?

Apesar do número expressivo nos resultados de pesquisa, foram identificadas apenas duas UIs consideradas relevantes para o ensino de Python, dentro do contexto proposto por este trabalho, para alunos do Ensino Médio publicados no período de 2018-2023 (Tabela 9). Isso se deve a especificidade dos filtros propostos, levando em consideração diversos critérios de exclusão. Grande parte das UIs

analisadas utilizam gamificação e jogos no contexto educacional, mas não envolvem o desenvolvimento de um jogo como parte do projeto de ensino. Outras UIs analisadas focaram no desenvolvimento de jogos utilizando outras plataformas e linguagens, como *sCool*, *AppInventor* e *Scratch*, não envolvendo o ensino de Python. Em casos mais raros, as UIs analisadas que se propõem a ensinar Python por meio do desenvolvimento de jogos não possuíam material instrucional detalhado ou não tinham como foco o Ensino Médio. Dessa forma, apenas as Unidades Instrucionais que cumprissem todos os requisitos foram selecionadas (Tabela 10).

Tabela 14. Unidades Instrucionais de ensino de Python

| Referência             | Título   |
|------------------------|--|
| (Miranda et al., 2022) | Metodologias ativas: ressignificando o ensino de programação para estudantes do ensino médio |
| (Diosdado, Alex, 2022) | Introducing Students to Computer Science: Using Python to Build a Basic Quiz Game            |

### 3.3.2 Quais conceitos de algoritmos e programação são ensinados nessas UIs?

O trabalho “Metodologias ativas: ressignificando o ensino de programação para estudantes do ensino médio” (Miranda et al., 2022) divide o conteúdo programático em 5 diferentes módulos, agrupando em cada etapa assuntos que seguem uma progressão de dificuldade entre si. O módulo 1 é composto por aspectos gerais e modo de instalação do Python, algoritmos, funções de entrada e saída de dados, chamadas de métodos, argumentos, comentários, variáveis, depuração e biblioteca matemática. O módulo 2 por sua vez trata dos operadores lógicos e relacionais, estruturas condicionais e estruturas de repetição. O módulo 3 aborda listas, strings e arquivos. No módulo 4 funções e a biblioteca PyGame, finalizando com um projeto no quinto e último módulo (Tabela 11).

Tabela 15. Conteúdo programático (Miranda, 2022)

|          |  |
|----------|--|
| Módulo 1 | História, Aspectos Gerais e Instalação do Python |
|          | Algoritmos                                       |
|          | Função de Entrada e Saída de dados               |
|          | Chamadas de Métodos                              |
|          | Argumentos                                       |
|          | Comentários                                      |
|          | Variáveis  |
|          | Depuração  |

|          |                                   |
|----------|-----------------------------------|
|          | Biblioteca Matemática             |
| Módulo 2 | Operadores Lógicos e Relacionais  |
|          | Estruturas Condicionais Repetição |
|          | Estruturas de Repetição           |
| Módulo 3 | Listas                            |
|          | Strings                           |
|          | Arquivos                          |
| Módulo 4 | Funções                           |
|          | PyGame                            |
| Módulo 5 | Módulo 5 Projeto                  |

O conteúdo proposto por Miranda (2022) foi construído com intuito de abranger todos os conhecimentos necessários para que o aluno seja capaz de desenvolver o projeto do Módulo 5, cujos temas foram escolhidos individualmente por cada estudante do curso. Os temas escolhidos para os projetos envolveram jogos com questões matemáticas, quiz sobre filmes, jogo da forca, entre outros.

O trabalho “Using Python to Build a Basic Quiz Game” (Diosdado et al., 2022) não possui um conteúdo programático tão bem organizado. O conteúdo é ministrado durante a realização de pequenos projetos, ensinando os conceitos necessários para que uma determinada atividade seja realizada. Os conceitos ensinados neste trabalho envolvem funções de entrada e saída de dados, argumentos, variáveis, operadores lógicos e relacionais, estruturas condicionais, strings, inteiros, booleanos e funções (Tabela 12).

Tabela 16. Conceitos ensinados por cada UI

|   | (Miranda et al., 2022) | (Diosdado et al., 2022) |
|---|------------------------|-------------------------|
| <b>História, Aspectos Gerais e Instalação do Python</b> | x                      |                         |
| <b>Algoritmos</b>                                       | x                      |                         |
| <b>Função de Entrada e Saída de dados</b>               | x                      | x                       |
| <b>Chamadas de Métodos</b>                              | x                      |                         |
| <b>Argumentos</b>                                       | x                      | x                       |
| <b>Comentários</b>                                      | x                      |                         |
| <b>Variáveis</b>  | x                      | x                       |
| <b>Depuração</b>  | x                      |                         |
| <b>Biblioteca Matemática</b>                            | x                      |                         |
| <b>Operadores Lógicos e Relacionais</b>                 | x                      | x                       |
| <b>Estruturas Condicionais</b>                          | x                      | x                       |
| <b>Estruturas de Repetição</b>                          | x                      |                         |
| <b>Listas</b>   | x                      |                         |
| <b>Strings</b>  | x                      | x                       |
| <b>Inteiros</b>   | x                      | x                       |
| <b>Booleanos</b>  | x                      | x                       |

|          |   |   |
|----------|---|---|
| Arquivos | x |   |
| Funções  | x | x |
| PyGame   | x |   |

O conteúdo coberto por Diosdado (2022) também é ensinado no curso proposto por Miranda (2022), que se mostrou muito mais completo e abrangente, além de possuir um grande diferencial por ministrar aulas sobre a biblioteca para criação de jogos PyGame.

### 3.3.3 Quais são as características instrucionais dessas UIs?

Ambos os trabalhos relevantes disponibilizam um plano de ensino, porém Miranda (2022) modularizou o conteúdo programático de forma que se cumprisse uma progressão no nível de dificuldade. Neste curso optou-se pela metodologia da Sala de Aula Invertida para estruturar o seu curso, por considerar que caso o estudante não tenha computador em casa para acompanhar os conteúdos enviados pela internet, o mesmo poderia utilizar o celular para este fim. Foram levadas em consideração as pesquisas mais recentes daquela época, que mostravam que a maioria das casas brasileiras (mais de 80%) que possuíam acesso à internet se conectavam por smartphones (VILLELA, 2018). Os aplicativos recomendados pela equipe foram o “*Dcoder, compiler IDE: Code & Programming on mobile*” como ferramenta de desenvolvimento para escrever os códigos, e o “*QPython3 — Python3 for Android*”, que contém alguns recursos como interpretador de códigos Python, ambiente de tempo de execução, editor e bibliotecas. Isso facilita o uso do Python no Android, além de ser gratuito.

A carga horária é de 62 horas/aula, sendo 36 horas correspondentes aos encontros presenciais e 26 horas/aula referente às videoaulas e atividades enviadas. Duas vezes na semana enviava-se uma videoaula, de duração média de 8 minutos, com atividades online na plataforma Google Sala de Aula. As atividades postadas sempre tinham três níveis de dificuldade: fácil, médio e difícil. O nível fácil era obrigatório para todos os participantes, os demais eram desafios opcionais para quem desejasse aprofundar-se no conteúdo. Já os encontros presenciais aconteciam nos sábados pela manhã, com duração de 3 horas, no decorrer de 3 meses. As videoaulas apresentavam o conteúdo programado dispondo-se de slides e também de jogos.

Considerando que grande parte dos primeiros vídeos se tratavam de conteúdos muito teóricos, e com receio de tornar o encontro presencial entediante, a equipe de professores envolvidos no curso utilizou a Ludicidade e a Aprendizagem baseada em Jogos Digitais nas primeiras atividades presenciais (Miranda, 2022). Foi utilizada uma atividade guiada por um Jogo Digital na plataforma online Kahoot! (Kahoot!, 2023), que se trata de um quiz com perguntas sobre o conteúdo das videoaulas da semana, e os estudantes competiam para ver quem respondia corretamente em menos tempo.

Para colaborar no aprendizado, as aulas seguintes continuaram a utilizar jogos digitais. O *LightBot* foi utilizado como uma atividade na aula sobre algoritmos, jogo em que um robô obedece a comandos ordenados pelo jogador com a finalidade de chegar aos quadros azuis presentes no caminho para iluminá-lo (Miranda, 2022) (Figura 3). Também adotou-se do jogo *CodeCombat* (CodeCombat, 2023) (Figura 4) que tem como propósito o ensino de *Javascript* e *Python*, enquanto o aluno comanda um guerreiro por meio de códigos escritos em uma das duas linguagens citadas. O *CodeCombat* foi empregado quando precisava-se ensinar sobre comentários no código, variáveis, depuração, chamada de métodos, argumentos, métodos com vários parâmetros, strings, estrutura de repetição de tipo `while true` (Miranda et al., 2022).

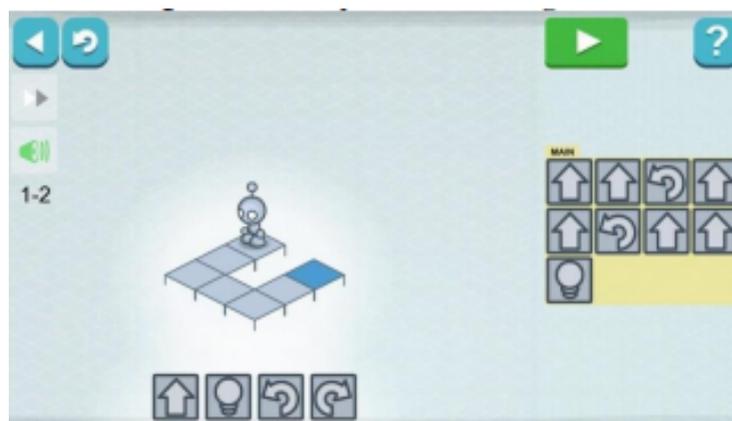


Figura 3. Exemplo de fase do *LightBot*. Fonte: Miranda et al., 2022.



Figura 4. Exemplo de fase do *CodeCombat*. Fonte: Miranda et al., 2022.

Para que no módulo 5 cada estudante fosse capaz de desenvolver um jogo, e assim colocar em prática tudo o que foi aprendido durante a capacitação, iniciou-se a apresentação do PyGame na parte final do módulo 4. Então os instrutores apresentaram exemplos de jogos desenvolvidos por PyGame com e sem interface gráfica para a turma. Durante o último módulo foi iniciada a metodologia de Aprendizagem Baseada em Projetos (Miranda, 2022), sendo todo o tempo disponibilizado para que os participantes desenvolvessem seus projetos de jogos. Dessa forma, os estudantes que antes aprendiam por meio de jogos, passavam a aprender enquanto criavam o seu próprio jogo, sob acompanhamento dos instrutores (Miranda, 2022). Cada participante teve a liberdade de escolher o tipo de jogo a ser desenvolvido. No último dia do curso, cada discente apresentou seu projeto para toda a turma e seus professores. Após a conclusão do curso, os alunos foram avaliados mediante o desempenho das atividades e a entrega do projeto e, em seguida, os certificados foram entregues aos alunos concluintes. Porém, não são reportados detalhes referente aos critérios adotados para avaliação do projeto.

Foram matriculados 16 estudantes para a realização do curso, sendo 7 meninos e 9 meninas, respectivamente representando 44% e 56%. Houveram 3 desistências, duas justificadas pela impossibilidade de comparecer às aulas presenciais por motivos trabalhistas, e uma por fatores familiares. Ao todo, 13 estudantes concluíram o curso, 6 meninos e 7 meninas.

No trabalho de Diosdado (2022) não define sua UI em módulos, mas determina uma sequência de instruções que devem ser cumpridas pelos alunos, com intuito de completar uma determinada tarefa ou atividade. Para ajudar a aumentar a exposição de um aluno ao campo da ciência da computação, a ideia foi criar uma atividade de divulgação que introduz alguns conceitos básicos de programação para alunos que tiveram pouca ou nenhuma experiência anterior. O primeiro plano de aula foi estruturado em cinco passos utilizando Madlib (Diosdado, 2022), um jogo de *Python* onde o usuário deve inserir textos em espaços em branco na história sem conhecer a mesma. Primeiro os alunos são apresentados aos conceitos de *strings* e números inteiros, em seguida, são apresentados à sintaxe da função de impressão e são solicitados a praticar a implementação dessa função por conta própria. No terceiro passo os alunos recebem uma explicação e alguns exemplos de variáveis em python e solicitados para salvar uma mensagem em uma variável e imprimir o conteúdo da mesma. Posteriormente há um breve exemplo de concatenação básica de strings, e por fim são mostrados como a função de entrada funciona e solicitados a inserir seu nome, salvar em uma variável e imprimir em uma frase usando concatenação.

A primeira atividade foi aplicada com dois estudantes do primeiro ano do ensino médio, que se voluntariaram para realizar a atividade em um sábado de manhã. Ambos estudantes possuíam breve experiência com programação anteriormente, porém estavam vendo pela primeira vez grande parte dos conceitos. A partir da análise dos resultados da primeira atividade, foi implementada uma segunda atividade usando o MadLib como um “esqueleto” e com base em alguns dos feedbacks recebidos. O novo conceito da atividade era o desenvolvimento de um jogo de “quiz” que foi projetado para ensinar conceitos como strings, variáveis, impressão e entrada. O quiz adicionou alguns novos conceitos que não foram inicialmente cobertos, como condicionais e booleanos. Como IDE principal foi utilizado o *PythonTutor* (PythonTutor, 2023), e a atividade foi aplicada com os mesmos dois estudantes da primeira atividade, a segunda levando em torno de 50 minutos para ser completada. Nada é especificado sobre o método de avaliação de aprendizagem utilizado.

Posteriormente a atividade do Quiz foi aplicada novamente, dessa vez em uma aula de álgebra na escola ‘*Gabrielino High School*’ em San Gabriel, Califórnia.

A atividade foi realizada por 25 estudantes, entre eles, 16 (64%) afirmaram não possuir nenhuma experiência prévia com programação, 8 afirmaram possuir pouca experiência de conceitos básicos, e 1 único estudante respondeu já possuir experiência prévia relevante. A atividade foi completada durante o período de uma aula, aproximadamente 50 minutos. Baseado nos feedbacks recebidos pelos alunos, que apontaram o ritmo da aula como muito acelerado e instruções muito longas e difíceis de acompanhar, a atividade recebeu melhorias e foi novamente aplicada em uma oficina na biblioteca *'Cal Poly's Kennedy Library*. Dessa vez 10 estudantes se inscreveram, porém 1 único realmente compareceu no dia. Dessa forma, a oficina se tornou uma espécie de mentoria, onde a aluna realizava por conta própria as instruções entregues pelo professor, e solicitava ajuda e explicações quando necessário. A atividade foi realizada em 90 minutos, e não foram reportados detalhes referente aos critérios adotados para avaliação do projeto.

Alguns termos relacionados aos objetivos de aprendizagem das UIs podem indicar o tipo de características que as UIs buscam para o ensino de algoritmos e programação em Python. É possível notar que estas têm investido em experiências de aprendizagem baseadas em projetos, além de aprendizagem baseada em jogos digitais, ludicidade e gamificação.

Tabela 17. Resumo das características instrucionais das UIs

| Ref            | Tipo de UI          | Duração       | Modo instrucional                              | Ambientes/ferramentas utilizadas   | Métodos instrucionais  | Método de avaliação                            | idioma         |
|----------------|---------------------|---------------|--|--|--|--|----------------|
| Miranda, 2022  | Curso               | 62h           | Vídeo aulas à distância, encontros presenciais | PyGame; CodeCombat; Lightbot; Kahoot!; DCoder; QPython3; Google Classroom; | Aulas expositivas; Atividades práticas Aprendizagem Baseada em Projetos; Sala de Aula Invertida, | Desempenho das atividades e entrega do projeto | Português - BR |
| Diosdado, 2022 | Atividades /oficina | 50min / 90min | Aulas presenciais                              | PythonTutor  | Aula expositiva; Atividades práticas   | Não informado                                  | Inglês         |

### 3.4 Discussão

As buscas realizadas mostraram que as UIs para o ensino de Python no Ensino Médio seguem uma tendência de utilizar uma aprendizagem sequencial, não comumente baseada em projetos, para introduzir os conceitos de algoritmos e programação. Ao buscar por UIs que ensinassem por meio do desenvolvimento de jogos, observou-se uma carência de trabalhos que se encaixassem nos filtros propostos por este trabalho. Há uma relevante parcela de UIs que utilizam aprendizagem baseada em jogos digitais e gamificação, através de plataformas como *CodeCombat* e *LightBot*, que foram previamente citados, porém sem foco no desenvolvimento de um novo jogo. Também se observa uma falta de cursos no nível de Ensino Médio que já usem ambientes mais alinhados ao desenvolvimento profissional, necessitando sempre de uma introdução por meio de outras disciplinas, como a matemática, por exemplo.

Das UIs que foram selecionadas, destaca-se o trabalho de Miranda (2022), que apresenta um curso completo, com plano de ensino estruturado, sequencial, e que se encerra com o desenvolvimento de um jogo. Ainda assim, a aprendizagem baseada em projetos não é utilizada no começo do curso, onde os conceitos são ensinados de forma isolada, e o projeto do jogo é apresentado somente no módulo final. O trabalho de Diosdado (2022) apresenta uma atividade aplicada com alunos do Ensino Médio, atividade esta que possui uma sequência de passos que devem ser reproduzidos com foco no desenvolvimento em um artefato final que não necessariamente representa um projeto. A segunda atividade aplicada por Diosdado representa uma metodologia de aprendizagem baseada em projetos, considerando que a sequência de instruções tem como finalidade o desenvolvimento de um jogo de Quiz, se mostrando assim relevante para o foco deste trabalho. Ambos trabalhos não demonstram preocupação com os métodos de avaliação, sendo que o trabalho de Miranda (2022) somente se baseou no desempenho dos alunos durante as aulas, e na entrega do projeto final. Diosdado (2022) por sua vez, não apresenta informações sobre os métodos avaliativos utilizados.

Dessa forma, pode-se inferir que existe uma carência de cursos para ensino de algoritmos e programação em Python, com metodologia de aprendizagem

baseada em projetos para desenvolvimento de jogos, e que possuam material instrucional detalhado e metodologias de avaliação explícitas.

**Ameaças à validade da revisão da literatura.** Algumas ameaças podem afetar a validade do estudo de mapeamento. Foram identificadas estas potenciais ameaças e estratégias foram organizadas para minimizar seus impactos. Mapeamentos sistemáticos podem se tornar tendenciosos, e resultados positivos podem ter mais chances de serem publicados do que os negativos. Porém, como o presente estudo não leva em consideração os resultados, sendo eles positivos ou negativos, já que tem como foco levantar e analisar as características das UIs, não é considerada uma ameaça neste caso. A possível omissão de estudos relevantes é outra ameaça que foi mitigada com a construção de strings de busca amplos, com uma variedade de termos e seus sinônimos pensados para serem o mais concisos possível. Ameaças relacionadas à seleção de Unidades instrucionais relevantes assim como a extração de dados foi mitigada com o desenvolvimento de regras de definição de inclusão e exclusão e critérios de qualidade. Todos os estudos foram analisados, por meio de um protocolo rígido, pelo autor e revisados pela orientadora.

## **4. DESENVOLVIMENTO DO CURSO “DESENVOLVENDO SEU PRIMEIRO JOGO COM PYTHON NO ENSINO MÉDIO - SHOOTER MOSQUITO”**

Neste capítulo, é apresentado o curso “DESENVOLVENDO SEU PRIMEIRO JOGO COM PYTHON NO ENSINO MÉDIO - SHOOTER MOSQUITO”, desenvolvido para o público do Ensino Médio, com o objetivo de introduzir conceitos de programação e desenvolvimento de jogos utilizando a linguagem Python. O curso é estruturado com base em uma análise cuidadosa do contexto do público-alvo, identificando suas necessidades e características, e é orientado por objetivos de aprendizagem bem definidos. A definição do plano de ensino e o desenvolvimento do material didático foram feitos de forma a garantir uma abordagem eficaz e envolvente para os estudantes. Para isso, foi adotada a abordagem ADDIE (Análise, Desenho, Desenvolvimento, Implementação e Avaliação), conforme proposto por Branch (2009), que proporciona uma metodologia sistemática e iterativa para o design instrucional. Essa abordagem assegura que o curso seja desenvolvido de maneira estruturada, considerando não apenas as necessidades dos alunos, mas também a avaliação contínua de sua eficácia ao longo do processo de aprendizagem.

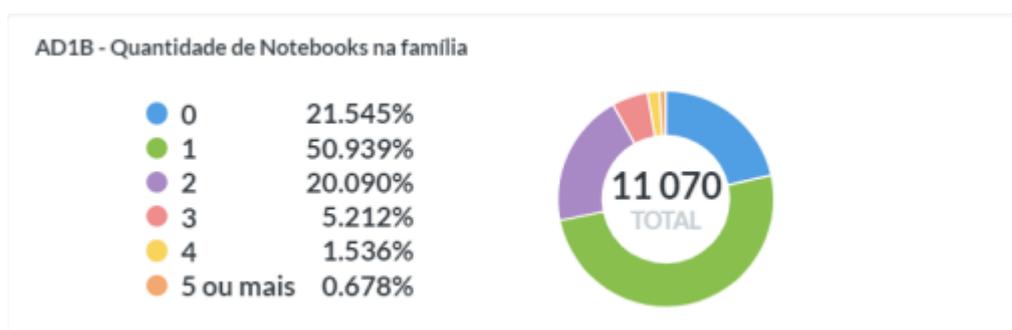
### **4.1 Análise de Contexto**

Inicialmente, analisa-se o domínio do cenário em que pretende-se aplicar o curso, buscando assegurar que será coerente com a realidade. A análise do contexto em relação às diretrizes dos currículos foi abordada na seção 2.1 da Fundamentação Teórica, correlacionando com os objetivos deste trabalho.

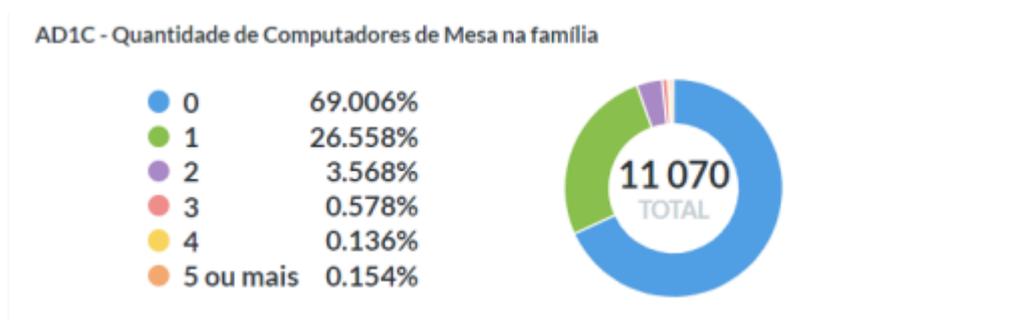
**Público alvo e Instituições de Ensino.** O Ensino Médio brasileiro corresponde ao período de três anos que finaliza a Educação Básica, sendo a faixa etária dos estudantes tipicamente entre 15 a 18 anos. De acordo com as diretrizes curriculares nacionais, os alunos já adquiriram competências básicas em matemática, línguas, ciências, etc. De acordo com o currículo do MEC, espera-se que os estudantes do Ensino Médio consolidem e continuem desenvolvendo os conhecimentos adquiridos no ensino fundamental, obtendo preparo para o mercado de trabalho e vida pessoal (MEC, 2017). Entretanto, o conhecimento em línguas estrangeiras pode ser limitado, o que exige o desenvolvimento de material

instrucional na língua nativa. Levando em consideração que somente há alguns anos a computação vem fazendo parte da BNCC, pode-se inferir que grande parte dos alunos não possui conhecimento prévio em algoritmos e programação.

Os jovens brasileiros entre 15 e 19 anos fazem parte do grupo etário com maior acesso a computadores e energia elétrica em relação a outros grupos etários. É possível inferir que, de modo geral, os alunos do ensino médio possuem acesso a computadores para desenvolver as atividades de ensino a distância (Figura 5). De acordo com o Censo Escolar Covid-19 (IFSC, 2021) realizado entre alunos de cursos técnicos integrados ao Ensino Médio e graduação do Instituto Federal de Santa Catarina, 75% dos entrevistados indicam ter ao menos um *notebook* disponível na família e 30% possuem ao menos um computador de mesa (Figura 6). A pesquisa também aponta que quase 90% dos alunos participaram das atividades a distância de forma total ou parcial no período da pandemia (Figura 7).

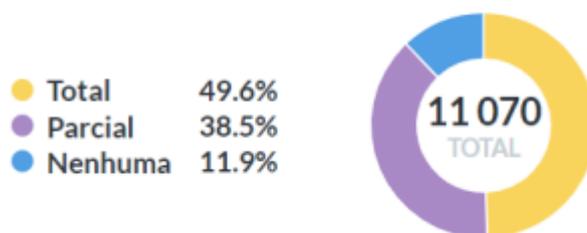


**Figura 5.** Quantidade de *notebooks* na família de acordo com o Censo Escolar Covid-19 (IFSC, 2021).



**Figura 6.** Quantidade de computadores de mesa na família (IFSC, 2021).

AD5 - O seu nível de participação em ANPs



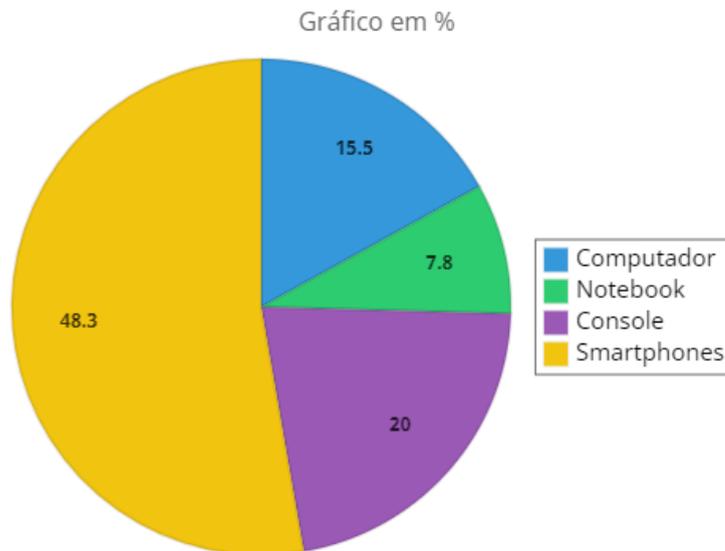
**Figura 7.** Participação dos alunos em atividades não presenciais (ANPs) (IFSC, 2021).

Dos alunos que não participaram das aulas a distância, 2,7% justificaram a ausência por não ter computador disponível e 2,4% por possuírem internet de baixa qualidade, com limitação de banda.

Diante deste recorte no contexto do IFSC, é possível supor que uma grande parte dos alunos do Ensino Médio possuam as condições e infraestrutura necessária para realizar cursos online de curta duração. Em 2018, o Brasil contava com 28.673 escolas que ofertavam o Ensino Médio, entre elas, 95,1% possuíam acesso à internet. Aproximadamente 60% dessas escolas possuem disponibilidade de infraestrutura e recursos tecnológicos como laboratório de informática e internet (INEP, 2019).

**Jogos digitais.** Hoje em dia, os jogos digitais podem ser acessados a partir de dispositivos celulares, consoles, computadores e laptops. A pesquisa Game Brasil, realizada em 2022, apresenta dados relevantes a respeito das plataformas utilizadas para jogos digitais (Figura 8). Estima-se que 74,5% da população brasileira, incluindo todas as faixas etárias, joga algum tipo de videogame (Pesquisa Game Brasil, 2022). Com tamanha popularidade, em especial entre os jovens, fica evidente o poder de influência dos games na sociedade atual, e é crescente o número de pesquisas que evidenciam seu potencial educativo (Passos e Novo Jr. 2021). Dessa forma, pode-se inferir que a inserção de jogos digitais no ambiente educacional, utilizando metodologias apropriadas, pode ser um grande aliado do aprendizado.

Figura 8. Gráfico das plataformas mais utilizadas para jogos (Pesquisa Game Brasil, 2022)



Em termos de grupos etários, 56,8% dos usuários de jogos digitais pertencem à faixa etária de idades entre 16 e 29 anos. Em relação às plataformas que jogam, os smartphones figuram com 48,3% da preferência dos jogadores. Videogames e consoles próprios para jogos representam 20%, os computadores 15,5% e os notebooks 7,8% (PESQUISA GAME BRASIL, 2022). Somados, computadores e notebooks representam uma parcela de 23,3% dos usuários, superior até mesmo aos consoles tradicionais. Dessa forma, infere-se que a proposta de um curso voltado para o desenvolvimento de um jogo de computador, que possa ser compartilhado com os amigos, seja um motivador para os estudantes.

Analisando os jogos mais populares da Twitch TV em 2023 (Twitch TV, 2023), de acordo com dados divulgados pela plataforma TwitchMetrics, pode-se concluir que os usuários tendem a preferir jogos baseados em esportes, guerras e jogos de tiro, carros e simuladores de mundo aberto (TwitchMetrics, 2023).

## 4.2 Objetivos de Aprendizagem

Os objetivos de aprendizagem do curso Python foram definidos com base nas competências e conceitos presentes nos currículos do MEC (2017 e 2022), SBC (2017) e CSTA (2017). Os objetivos foram selecionados considerando o contexto educacional e a base necessária para conhecimento de conceitos e práticas básicas de algoritmos e programação, além da linguagem Python, introduzindo conceitos para a realização do projeto de desenvolvimento de um jogo.

Tabela 13. Objetivos de aprendizagem do curso Python

| ID  | Objetivo de aprendizagem  | Referência(s)                                 |
|-----|---|---|
| OA1 | Identificar, decompor e resolver problemas a partir de soluções algorítmicas, reutilizando partes de soluções já existentes quando possível   | CSTA: 3A-AP-17;<br>3A-DA-13;<br>MEC:EM13CO01; |
| OA2 | Compreender e justificar a utilização de diferentes conceitos de programação, como eventos, listas, condicionais, estruturas de repetição, funções de entrada e saída, variáveis.     | CSTA<br>:3A-AP-14;3A-AP-15;<br>3A-AP-16;      |
| OA3 | Compreender as funcionalidades e aplicações da biblioteca PyGame para o desenvolvimento de jogos em Python  | CSTA:3A-AP-20;                                |
| OA4 | Avaliar e aperfeiçoar os artefatos e componentes desenvolvidos durante as aulas do curso, adicionando imagens, alterando cores, tipografia e ícones, a fim de aprimorar a usabilidade | CSTA: 3A-AP-21;<br>MEC:EM13CO02;              |
| OA5 | Identificar a presença e formas de uso das tecnologias utilizadas no mundo do trabalho.   | MEC:EM13CO09;                                 |
| OA6 | Compartilhar o jogo desenvolvido  | MEC:EM13CO20;EM13<br>CO22;                    |
| OA7 | Lembrar a importância do combate ao mosquito da dengue para segurança e saúde pública   | (ONU, 2023)                                   |

### 4.3 Design do Curso

Alinhado ao contexto e aos objetivos de aprendizagem, o curso foi projetado para iniciantes, com intuito de ensinar a desenvolver um jogo 2D em Python utilizando a biblioteca PyGame. Escolheu-se um jogo no estilo *shooter game* com a temática que faz alusão ao combate ao mosquito da dengue, alinhado com os objetivos de desenvolvimento sustentável, que inclui a cobertura universal de saúde (ONU, 2023). Prevê-se uma aplicação interdisciplinar deste curso, assim, o curso poderá ser aplicado nos itinerários do ensino médio, dentro de ciências da natureza e suas tecnologias, que envolve temas da área da Biologia, Física e Química (MEC, 2021), ou como uma atividade extracurricular. Visa-se ensinar os principais conceitos de algoritmos e programação em Python, e estimular os alunos a exercitarem o entendimento sobre as diferentes ferramentas abordadas, além da biblioteca de desenvolvimento de jogos PyGame, com foco no desenvolvimento do projeto do jogo. Espera-se que os estudantes consigam desenvolver um jogo pré definido dentro do curso, e que consigam compartilhar os resultados com seus colegas.

Para facilitar a aplicação do curso no Ensino Médio, projetou-se um curso de curta duração. O curso fornecerá todo o conteúdo necessário para o entendimento do tema e avaliação de aprendizagem dos alunos, dessa forma poderá ser aplicado

tanto de forma presencial em sala de aula, sem a necessidade da assistência de professores graduados na área de computação, quanto de forma online.

O plano de ensino foi desenvolvido levando em consideração principalmente as diretrizes da BNCC (MEC 2017) e as competências expostas pelo MEC (2017 e 2022), SBC (2017) e CSTA (2017) para levantar os objetivos do curso, levando em consideração o estágio escolar dos alunos do Ensino Médio. Também foram adicionadas atividades extras com o objetivo de estimular os estudantes a modificarem os códigos por trás do projeto de desenvolvimento do jogo *shooter mosquito*, assim como a introdução do conteúdo necessário para entender o básico de algoritmos, linguagem Python e uso da biblioteca *PyGame*. Todas as informações relevantes coletadas no mapeamento sistemático foram consideradas, como conceitos e métodos abordados, e até mesmo a biblioteca *PyGame*.

Tabela 14. Plano de Ensino

| Aula (duração)                          | Conteúdo   | Objetivos de Aprendizagem | Método Instrucional                | Material Instrucional | Avaliação |
|---|--|---------------------------|------------------------------------|-----------------------|-----------|
| <b>Introdução</b>                       |  |                           |                                    |                       |           |
| 1 (15 min)                              | Aplicações e oportunidades com Python, Jogos com Python  | OA5;                      | Aula interativa                    | Slides, vídeo         | Quiz      |
| 2 (10 min)                              | Instalação do Python, Pycharm, visão geral sobre a biblioteca PyGame e configurações iniciais  |                           | Aula prática                       | Slides, vídeo         | Quiz      |
| <b>Projeto do jogo shooter mosquito</b> |  |                           |                                    |                       |           |
| 3 (35 min)                              | Criação da primeira tela: comentários, variáveis, tipos primitivos, operadores lógicos, importações, estruturas de repetição                 | OA1; OA2; OA3;            | Aula interativa, atividade prática | Slides, vídeo         | Quiz      |
| 4 (30 min)                              | Criando o personagem jogável: ícones, imagens, música, estilização   | OA1; OA2; OA3; OA4        | Aula interativa, atividade prática | Slides, vídeo         | Quiz      |
| 5 (30 min)                              | Criando mecânica de movimentação: funções, argumentos, chamada de função, operadores lógicos e relacionais, estruturas de repetição, eventos | OA1; OA2                  | Aula interativa, atividade prática | Slides, vídeo         | Quiz      |
| 6 (10min)                               | Movimentação de inimigos   | OA1; OA2                  | Aula interativa, atividade prática | Slides, vídeo         | Quiz      |
| 7 (22 min)                              | Comando de ação do personagem, estruturas condicionais, retorno de função, escopo de variáveis.  | OA1; OA2                  | Aula interativa, atividade prática | Slides, vídeo         | Quiz      |
| 8 (15 min)                              | Deteção de colisão utilizando métodos matemáticos. Criação do placar da partida.   | OA1; OA2; OA4             | Aula interativa, atividade prática | Slides, vídeo         | Quiz      |
| 9 (25 min)                              | Múltiplos inimigos simultâneos: funções com listas   | OA1; OA2; OA4             | Aula interativa, atividade prática | Slides, vídeo         | Quiz      |
| 10 (15 min)                             | Compartilhando o projeto desenvolvido: criando executável e adicionando jogo em plataforma online  | OA6; OA7                  | Aula interativa, atividade prática | Slides, vídeo         |           |

#### 4.4 Material Didático

As aulas foram estruturadas como apresentações interativas, que incluem vídeos, atividades interativas e links úteis, além do código do jogo e outros materiais como imagens e áudio. Os vídeos apresentam conteúdos teóricos com o suporte de slides, intercalando o desenvolvimento prático do jogo proposto, utilizando os

conceitos abordados de forma imediata. Cada funcionalidade do jogo é inicialmente explicada de forma expositiva e, em seguida, desenvolvida na prática com base nos conceitos apresentados.

Figura 9. Exemplo de vídeo 1

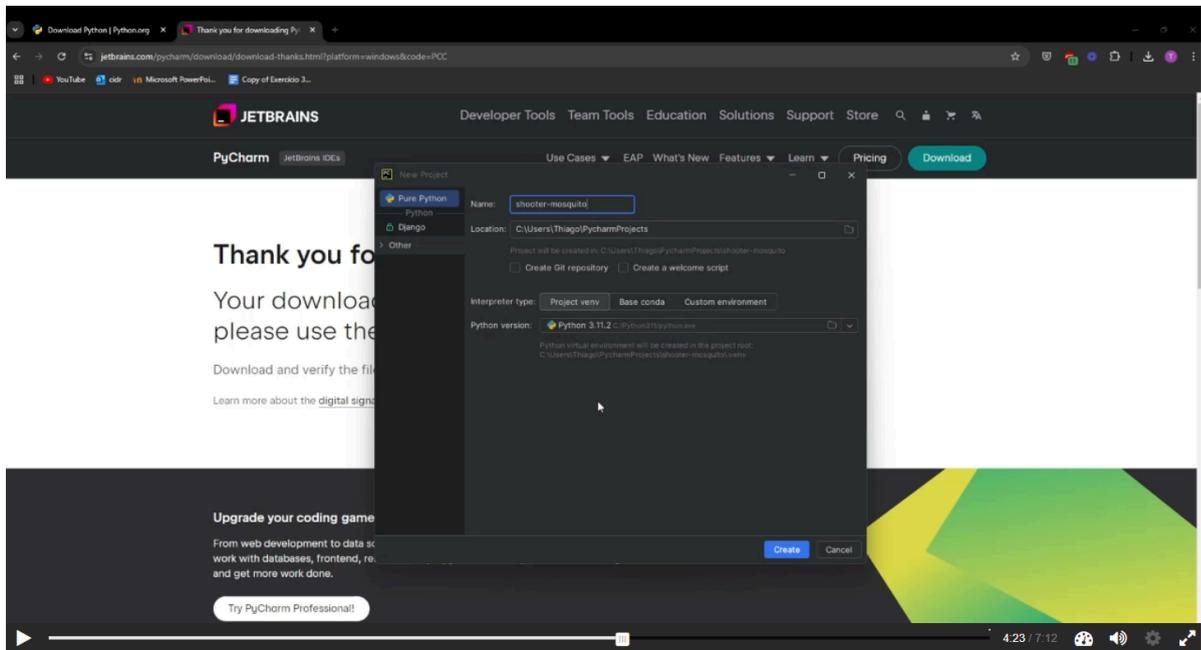
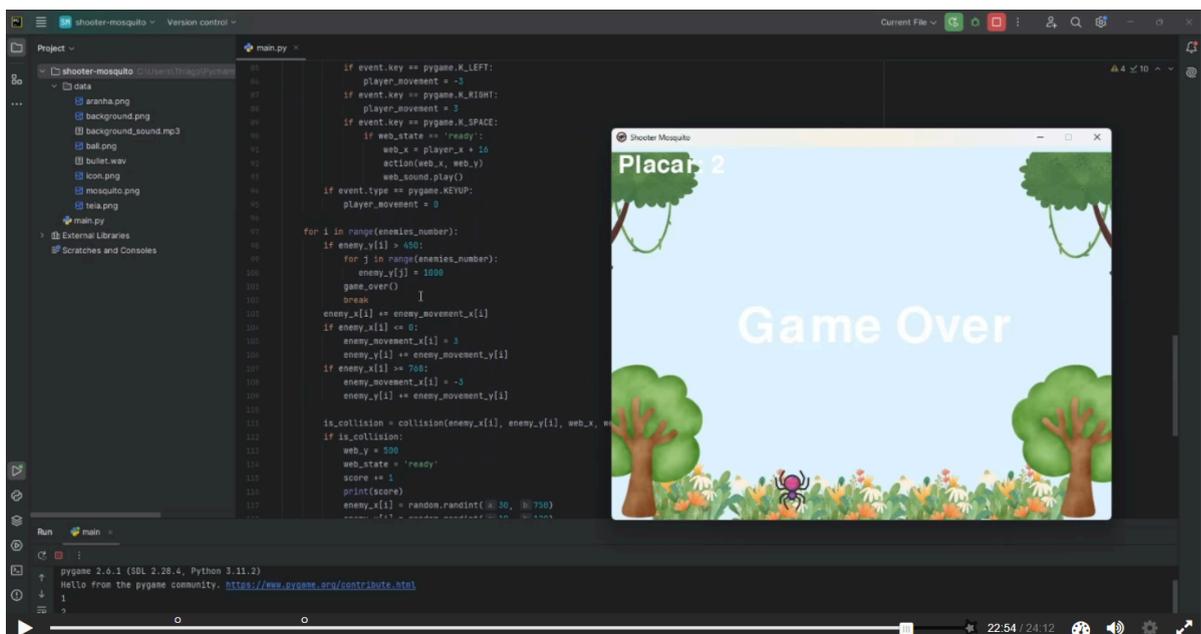


Figura 10. Exemplo de vídeo 2



Na Figura 9, apresentamos um exemplo da primeira aula prática, na qual foi realizado o passo a passo da instalação das ferramentas, facilitando o preparo do

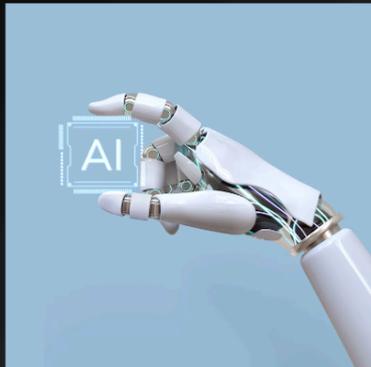
ambiente de desenvolvimento pelos alunos. Já na Figura 10, temos um exemplo onde a funcionalidade a ser desenvolvida é apresentada no jogo pronto, servindo como referência antes de passarmos para a implementação no código. O conteúdo dos vídeos é dividido em partes e apresentado por meio de slides, que também foram disponibilizados separadamente (Tabela 15).

Tabela 15. Exemplos de slides do curso

| Slides   |
|--|
|  <p>The slide features a dark background with the main title 'Programação com Python' in large, light blue font. Below it, a light blue rounded rectangle contains the subtitle 'Desenvolvendo Jogos'. To the right is a blue game controller icon. At the bottom, there are logos for 'COMPUTAÇÃO NA ESCOLA', 'GQS', 'INCoD', 'INE', and 'UNIVERSIDADE FEDERAL DE SANTA CATARINA'. A small copyright notice is at the very bottom.</p> |
| <b>Aula 1</b>  |

# Inteligência Artificial

Amplamente aplicado na área de Aprendizagem de Máquina



Por meio de algoritmos, dá aos computadores a capacidade de identificar padrões em grandes quantidades de dados e fazer previsões

- Sistemas de recomendação
- Sistemas de reconhecimento de imagem

Grandes empresas como YouTube, Spotify, Facebook, Netflix, Dropbox e até o Google, tem parte de suas aplicações desenvolvidas em Python

Copyright © Computação na Escola/INCoD/INE/UFSC. Todos os Direitos Reservados. Proibida a distribuição e reprodução sem autorização prévia. Fonte de imagem: freepik.com

## Aula 2

# Requisitos para instalação

Para uma instalação bem sucedida de todas as ferramentas utilizadas no curso, espera-se que os seguintes requisitos sejam cumpridos:

**Sistema Operacional:** Windows 8 ou posterior, macOS 10.14 ou posterior, ou distribuições Linux modernas

**Processador:** Dual-core de 2 GHz ou superior

**Memória RAM:** 2 GB ou mais

**Espaço em Disco:** Aproximadamente 4 GB

Copyright © Computação na Escola/INCoD/INE/UFSC. Todos os Direitos Reservados. Proibida a distribuição e reprodução sem autorização prévia.

## Aula 3

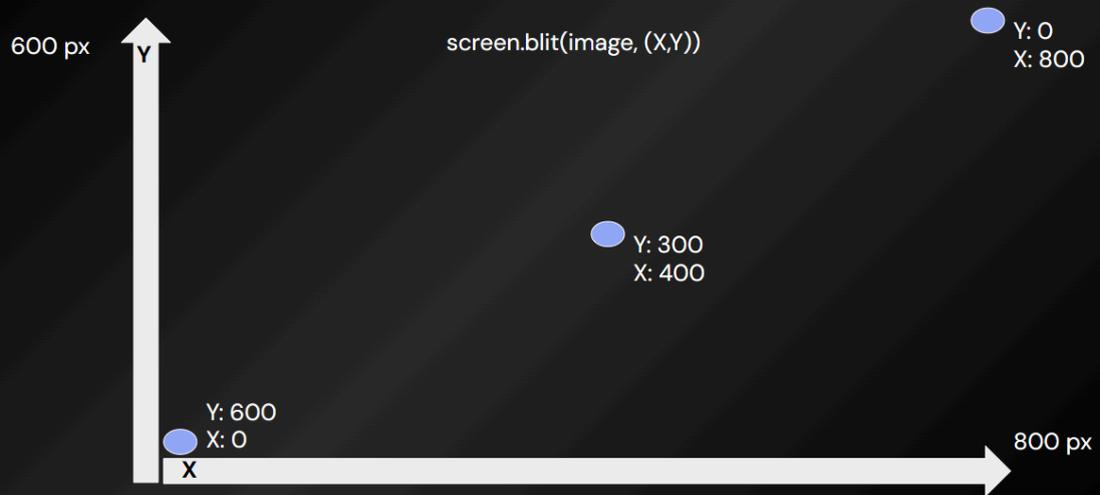
## Tamanho de Tela



Copyright © Computação na Escola/INCoD/INE/UFSC. Todos os Direitos Reservados. Proibida a distribuição e reprodução sem autorização prévia. Fonte de imagem: freepik.com

## Aula 4

## Desenhando na Tela



Copyright © Computação na Escola/INCoD/INE/UFSC. Todos os Direitos Reservados. Proibida a distribuição e reprodução sem autorização prévia.

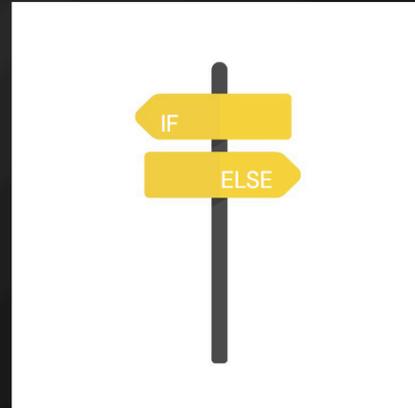
## Aula 5

# Estruturas condicionais

Estruturas condicionais permitem que o programa tome decisões diferentes, de acordo com um valor fornecido. Por exemplo:

Se  $x$  for menor que 10, multiplique  $x$  por 2

Se  $x$  for maior que 10, divida  $x$  por 2



Copyright © Computação na Escola/INCoD/INE/UFSC. Todos os Direitos Reservados. Proibida a distribuição e reprodução sem autorização prévia.

## Aula 6

# Movimentação do inimigo

## Aula 6

Copyright © Computação na Escola/INCoD/INE/UFSC. Todos os Direitos Reservados. Proibida a distribuição e reprodução sem autorização prévia.

## Aula 7

# Escopo de variáveis

Define a parte do programa onde ela é visível e acessível

Onde a variável pode ser usada

Dentro de uma função, bloco de código, ou em todo o programa

Qual variável é usada quando há ambiguidade

Se duas variáveis com o mesmo nome existem em diferentes escopos, a variável no escopo mais local é usada

Tipos de escopo

Local, Global, Built-in



Copyright © Computação na Escola/INCoD/INE/UFSC. Todos os Direitos Reservados. Proibida a distribuição e reprodução sem autorização prévia. Fonte de imagem: freepik.com

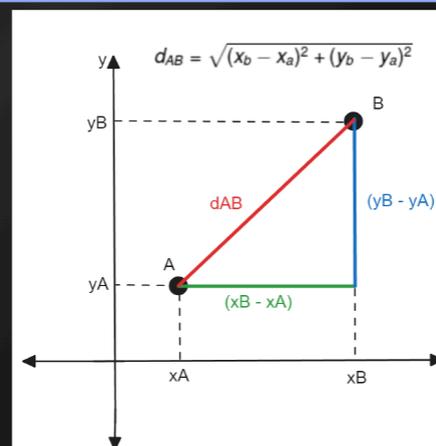
## Aula 8

# Distância entre dois pontos

Teorema de Pitágoras

Para calcular a distância entre dois pontos no plano cartesiano, utilizamos o teorema de Pitágoras.

Dados os pontos  $A(x_A, y_A)$  e  $B(x_B, y_B)$ , é possível construir um triângulo retângulo cuja hipotenusa seja exatamente o segmento AB



Copyright © Computação na Escola/INCoD/INE/UFSC. Todos os Direitos Reservados. Proibida a distribuição e reprodução sem autorização prévia. Fonte de imagem: freepik.com

## Aula 9

# Funções com Listas

Alterar o valor da lista usando índice (posição da lista começando do 0)

- o `bebidas = ['agua', 'suco', 'refrigerante']` # Cria a lista
- o `bebidas[2] = 'agua de coco'` # [ 'agua', 'suco', 'agua de coco' ]

Adicionar elementos

- o `bebidas = ['agua', 'suco']` # Cria a lista
- o `bebidas.append('energetico')` # [ 'agua', 'suco', 'energetico' ]

Remover elementos

- o `bebidas.remove('cha')` # Remove a primeira ocorrência de 'cha'
- o `item_removido = bebidas.pop(2)` # Remove e retorna o elemento no índice 2

Copyright © Computação na Escola/INCoD/INE/UFSC. Todos os Direitos Reservados. Proibida a distribuição e reprodução sem autorização prévia.

## Aula 10

# Código / Comandos

```
import os, sys
dirpath = os.getcwd()
sys.path.append(dirpath)
if getattr(sys, "frozen", False):
    os.chdir(sys._MEIPASS)
```

```
pip install pyinstaller
pyinstaller -F main.py
pyinstaller main.spec
```

Copyright © Computação na Escola/INCoD/INE/UFSC. Todos os Direitos Reservados. Proibida a distribuição e reprodução sem autorização prévia.

Cada aula começa com uma revisão do conteúdo abordado na aula anterior e finaliza com uma introdução ao conteúdo da próxima aula. Para reforçar o aprendizado, foram incluídas algumas atividades interativas com diferentes formatos, incluindo múltipla escolha, múltipla escolha com seleção múltipla, verdadeiro ou falso, e preenchimento de lacunas (Tabela 16).

Na primeira aula, foram apresentados os objetivos e o conteúdo do curso, que consiste no desenvolvimento de um jogo no estilo "shooter mosquito" utilizando Python e a biblioteca Pygame. Além disso, foi destacada a popularidade e versatilidade do Python, utilizado em diversas áreas. Também foram introduzidas as ferramentas PyCharm, como IDE para facilitar o desenvolvimento, e Pygame, para simplificar a criação de jogos. Por fim, foi explicado o impacto do curso na construção de habilidades de resolução de problemas e no aprendizado de conceitos essenciais de programação.

Na segunda aula, o foco foi na instalação e configuração das ferramentas necessárias para o curso, incluindo o Python, o PyCharm Community Edition e a biblioteca Pygame. Foram discutidos os requisitos mínimos de sistema, e o processo de instalação foi detalhado. Após isso, foi criado o projeto no PyCharm, com configurações específicas, como a definição do interpretador Python e a instalação do Pygame. A aula terminou com a importação inicial da biblioteca no arquivo principal do projeto, preparando o ambiente para o desenvolvimento do jogo.

A terceira aula envolveu a criação da janela do jogo utilizando o Pygame, além da configuração do loop principal para manter o jogo em execução. Foram explicados o uso de variáveis e tipos primitivos em Python, como inteiros e booleanos, para controlar elementos do jogo, como posição e velocidade. Implementou-se o loop while para garantir que o jogo ficasse ativo enquanto a variável running fosse verdadeira, e foi demonstrado como tratar eventos para fechar a janela ao clicar no ícone de fechar aba "X".

Na quarta aula, trabalhou-se na personalização do jogo, incluindo a adição de elementos visuais como título, ícone, imagens de fundo e personagens, e a implementação de funcionalidades como a exibição contínua dos objetos na tela e a atualização da interface. A biblioteca Pygame foi utilizada para criar essas interações, e também foi introduzido o uso de áudio, com a configuração de um som de fundo tocando continuamente durante o jogo.

A quinta aula abordou a implementação da movimentação do personagem utilizando o teclado. Foram usados eventos do Pygame para capturar as teclas pressionadas e controlar a posição do personagem, alterando suas coordenadas no

eixo X. Também foram aplicados limites para impedir que o personagem ultrapassasse os limites da tela, e foi criada uma função para renderizar elementos na tela, permitindo a reutilização do código tanto para o personagem quanto para os inimigos. A movimentação foi controlada pelas setas esquerda e direita do teclado, e a lógica foi ajustada para parar o movimento quando a tecla fosse solta.

Na sexta aula, foi implementada a movimentação dos inimigos. Utilizando a biblioteca random, geraram-se posições aleatórias para os inimigos no início do jogo, garantindo que aparecessem em locais variados na tela. A movimentação foi configurada para que os inimigos se deslocassem para a direita até atingirem o limite da tela, momento em que desciam e começavam a se mover para a esquerda. Aplicou-se uma lógica de limites para evitar que os inimigos ultrapassassem os limites horizontais da tela, tornando o jogo mais dinâmico.

Na sétima aula, foi implementado o disparo de projéteis. Criaram-se variáveis para controlar a posição e o estado do projétil, que pode estar "ready" (pronto para ser disparado) ou "triggered" (disparado). Quando a barra de espaço é pressionada, o projétil é disparado da posição do personagem e se move para cima, sendo reiniciado quando sai da tela. Ajustou-se a lógica para que o projétil não seguisse o movimento do personagem, e foi adicionado um efeito sonoro ao disparo.

Na oitava aula, foi trabalhada a detecção de colisões. Utilizou-se o teorema de Pitágoras para calcular a distância entre o projétil e o inimigo, empregando a biblioteca math para calcular a raiz quadrada e as diferenças nos eixos X e Y. Se a distância fosse menor ou igual a X pixels, considerava-se que houve colisão. A aula também abordou a manipulação de variáveis globais e a lógica para aumentar a pontuação e reiniciar as posições dos elementos após a colisão.

Na nona aula, foi ensinado a renderizar múltiplos inimigos no jogo. Para isso, utilizou-se listas para armazenar as informações de cada inimigo, como posição e imagem, e estruturas de repetição para atualizar a posição dos inimigos durante o jogo. Também foi criada uma função para exibir o placar na tela, além de uma lógica para encerrar o jogo quando os inimigos se aproximam demais do personagem. Com isso, o jogo tornou-se mais dinâmico, com múltiplos inimigos e uma interface para o placar.

Por fim, na décima aula, o curso foi encerrado com a criação do arquivo executável do jogo e a publicação na plataforma Game Jolt, permitindo que o jogo fosse compartilhado de forma online.

Tabela 16. Exemplos de questionários do curso

| <b>Múltipla Escolha com múltipla seleção</b>  |
|---|
| <p>Qual das seguintes áreas é uma aplicação típica do Python?</p> <ul style="list-style-type: none"><li><input type="checkbox"/> Análise de dados e Inteligência Artificial</li><li><input type="checkbox"/> Desenvolvimento Web</li><li><input type="checkbox"/> Desenvolvimento de Jogos</li><li><input type="checkbox"/> Automação de tarefas</li><li><input type="checkbox"/> Desenvolvimento de sistemas operacionais</li></ul> <p><a href="#">Verificar</a></p> |
| <b>Verdadeiro ou Falso</b>  |
| <p>IDE é um ambiente de desenvolvimento integrado que oferece ferramentas para escrever, depurar e executar códigos</p> <p><input type="radio"/> Verdadeiro    <input type="radio"/> Falso</p> <p><a href="#">Verificar</a></p>   |
| <b>Múltipla Escolha</b>   |

Assinale a afirmativa INCORRETA sobre tipos primitivos em Python



O tipo 'bool' em Python é usado para representar valores lógicos True (verdadeiro) e False (falso)

O tipo de dado 'int' em Python é usado para representar números inteiros, como -5, 0 e 100

Em Python, os tipos primitivos incluem inteiros (int), números de ponto flutuante (float), strings (str) e booleanos (bool)

Python não suporta o tipo de dado 'float', sendo restrito apenas a números inteiros

### Preencha as lacunas

Preencha corretamente os espaços em branco

Uma variável definida dentro de uma função ou bloco de código, e que só pode ser acessada dentro daquele bloco é uma variável de escopo

Uma variável definida fora de qualquer função ou bloco de código é  e pode ser acessada em qualquer lugar do programa

O escopo  é o mais restrito e protege o código de interferências externas

O escopo  deve ser utilizado com cuidado para evitar conflitos de nomes

Verificar

A tabela 17 apresenta as ferramentas utilizadas no desenvolvimento do jogo, destacando a tentativa inicial de utilizar a IDE online gratuita chamada *'Replit'*. No entanto, devido à impossibilidade de criar o arquivo executável do jogo por meio dessa plataforma, o que compromete o cumprimento de um dos objetivos de aprendizagem do curso, optou-se por utilizar a IDE PyCharm Community Edition. A escolha do PyCharm se deu por sua robustez e compatibilidade com as bibliotecas necessárias para o desenvolvimento do jogo, proporcionando um ambiente mais adequado para a criação do arquivo executável e o alcance dos objetivos pedagógicos propostos.

Tabela 17. Ferramentas Utilizadas para criar o Jogo

| <b>Ferramenta</b>                | <b>Função</b>                               |
|----------------------------------|---|
| <b>Python</b>                    | Linguagem de programação                    |
| <b>Pygame</b>                    | Biblioteca Python para criação de jogos     |
| <b>PyCharm Community Edition</b> | Ambiente de Desenvolvimento Integrado - IDE |

A fim de facilitar o desenvolvimento das aulas, foram utilizadas ferramentas online gratuitas, apresentadas na Tabela 18.

Tabela 18. Ferramentas Extras Utilizadas para criar o Curso

| <b>Ferramenta</b>          | <b>Função</b>  |
|----------------------------|--|
| <b>Google Drive</b>        | Upload dos vídeos para avaliação e correção            |
| <b>Google Docs</b>         | Criação do roteiro das aulas                           |
| <b>Google Presentation</b> | Criação dos slides                                     |
| <b>H5P</b>                 | Montagem das aulas, adicionando vídeos e questionários |
| <b>OBS Studio</b>          | Gravação dos vídeos das aulas                          |
| <b>Microsoft Clipchamp</b> | Edição dos vídeos                                      |

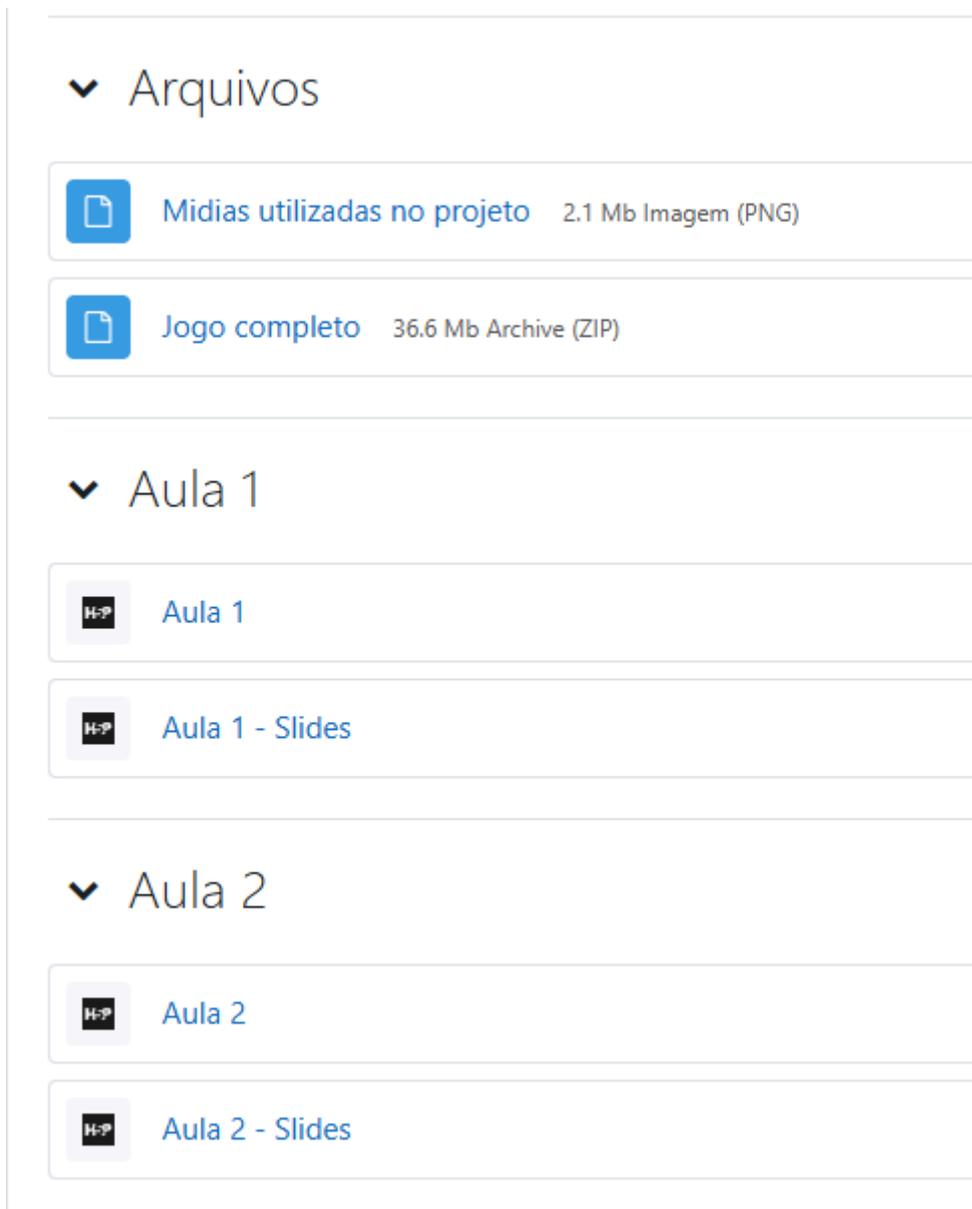
Para o desenvolvimento do jogo, foram utilizados diversos arquivos de mídia, conforme apresentado na tabela 19. Todos os arquivos foram obtidos de sites que oferecem recursos sem direitos autorais, como o Freepik.com, ou criados por meio de ferramentas de edição online. Essa abordagem garantiu o cumprimento das normas de licenciamento e contribuiu para a criação de elementos visuais personalizados para o jogo, mantendo a originalidade e evitando o uso indevido de conteúdos protegidos por direitos autorais.

Tabela 19. Imagens e recursos utilizados para criar o jogo

| Utilidade                 | Nome do Arquivo      | Arquivo   |
|---------------------------|----------------------|---|
| Personagem jogável        | aranha.png           |                            |
| Imagem de fundo           | background.png       |                            |
| Ícone do jogo             | icon.png             |                            |
| Inimigos                  | mosquito.png         |                          |
| Projétil disparado        | teia.png             |                          |
| Som do projétil disparado | bullet.mp3           | <br>bullet               |
| Som de fundo              | background_sound.mp3 | <br>background_sou<br>nd |

O curso é disponibilizado via Moodle utilizando h5p (<http://h5p.org>), e organizado em tópicos por aula. Todos os recursos de mídia utilizados na criação do jogo, assim como o jogo completo, estão disponíveis na seção “Arquivos”.

Figura 11. Exemplo de Tópicos



The image shows a Moodle course interface with three main sections, each with a dropdown arrow:

- Arquivos**
  - Midias utilizadas no projeto 2.1 Mb Imagem (PNG)
  - Jogo completo 36.6 Mb Archive (ZIP)
- Aula 1**
  - Aula 1
  - Aula 1 - Slides
- Aula 2**
  - Aula 2
  - Aula 2 - Slides

## 5. CONCLUSÃO

Este trabalho apresenta o desenvolvimento do curso de Programação com Python, focado no ensino de fundamentos de programação para alunos do ensino médio. Inicialmente, foi realizada uma síntese da fundamentação teórica, abordando o ensino de programação com Python utilizando jogos no ensino médio e conceitos fundamentais de programação. Em seguida, um mapeamento sistemático foi conduzido para identificar o estado da arte em unidades instrucionais com objetivos semelhantes ao do presente estudo, que identificou que atualmente existem muito poucos cursos com estas características.

Seguindo um processo de design instrucional, foi feita uma análise dos perfis de estudantes, professores e ambientes escolares brasileiros, com o objetivo de definir objetivos de aprendizagem e elaborar o plano de ensino do curso. A partir desse plano, foi desenvolvido o material didático, composto por slides interativos, vídeos, questionários e atividades práticas, visando oferecer o conteúdo necessário para que os alunos possam criar um *Shooter Game* utilizando Python, a biblioteca Pygame e o ambiente de desenvolvimento integrado Pycharm.

Dessa maneira, espera-se contribuir com o ensino de programação na educação brasileira, apresentando conceitos fundamentais de programação e desenvolvimento de jogos com Python, estimulando o interesse dos jovens nas áreas de tecnologia e desenvolvendo o pensamento algorítmico. O curso desenvolvido agrega ao atual estado da arte, considerando que existem poucas unidades instrucionais que focam no ensino de programação para jovens do ensino médio, a partir do desenvolvimento de jogos utilizando o Python, principalmente no idioma português.

Com relação a trabalhos futuros, sugere-se a aplicação e avaliação do curso com alunos do público-alvo, visando a obtenção de *feedback* que ajudem a aprimorar e refinar o material didático desenvolvido. Para a contribuição contínua do tema, sugere-se a criação de novos cursos, que desenvolvam jogos mais complexos aplicando e aprofundando os conceitos apresentados neste curso.

## REFERÊNCIAS

Atlas das Juventudes. Evidências Para A Transformação Das Juventudes (2021). Disponível em:

<https://atlasdasjuventudes.com.br/wp-content/uploads/2021/06/ATLAS-DAS-JUVE>  
Acesso em: 17 de Junho de 2023.

Basili, V., Caldiera, G. e Rombach, H. D. The Goal Question Metric Approach. John Wiley & Sons, Encyclopedia of Software Engineering, 1994.

BNCC. Base Nacional Comum Curricular. Brasília: MEC, 2017. Disponível em:  
[http://basenacionalcomum.mec.gov.br/images/BNCC\\_EI\\_EF\\_110518\\_versaofinal\\_site.pdf](http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf). Acesso em: 30 de novembro de 2022.

Branch, R. Instructional design: The ADDIE approach. Springer, 2009.

Carol Willing, and Michael Zingale. Teaching and Learning with Jupyter, 2019.  
Disponível em: <https://jupyter4edu.github.io/jupyter-edu-book/>. Acesso em: 30 de Novembro de 2022

CodeCombat. 2023. Disponível em: <https://codecombat.com/> Acesso em: 28 de Junho de 2023

FreeCodeCamp. Learn Python by Building Five Games - Full Course, 2019.  
Disponível em: <https://www.youtube.com/watch?v=XGf2GcyHPPhc> Acesso em: 28 de Junho de 2023

Google Colab. Olá, este é o Colaboratory, 2019. Disponível em:  
<https://colab.research.google.com/>. Acesso em: 30 de Novembro de 2022

IFSC. IFSC Analytics: Censo Escolar Covid-19. 2021. Disponível em:  
[https://censo.ifsc.edu.br/files/formulario\\_censo\\_servidores.pdf](https://censo.ifsc.edu.br/files/formulario_censo_servidores.pdf) Acesso em: 17 de Junho de 2023.

INEP. Dados Do Censo Escolar. 2019. Disponível em: <https://www.gov.br/inep/pt-br>

Acesso em: 19 de Junho de 2023

Jetbrains, PyCharm, 2023. Disponível em: <https://www.jetbrains.com/pt-br/pycharm/>  
Acesso em: 30 de Junho de 2023

MEC. MEC aprova parecer que define normas sobre o ensino de computação na educação básica, 2022. Disponível em:  
<https://www.gov.br/mec/pt-br/assuntos/noticias/mec-aprova-parecer-que-define-normas-sobre-o-ensino-de-computacao-na-educacao-basica>. Acesso em: 30 de novembro de 2022.

MEC. Novo Ensino Médio, 2021. Disponível em:  
<https://www.gov.br/mec/pt-br/novo-ensino-medio/itinerarios-formativos-do-novo-ensino-medio/ciencias-da-natureza-e-suas-tecnologias> Acesso em: 19 de Junho de 2023

MEC. Computação, Complemento à BNCC. Disponível em:  
<http://portal.mec.gov.br/docman/fevereiro-2022-pdf/236791-anexo-ao-parecer-cnec-eb-n-2-2022-bncc-computacao/file> Acesso em: 27 de Junho de 2023

Newzoo. Brazilian games market to hit \$2.3bn in 2021. Nov. 20, 2020. Disponível em:  
<https://newzoo.com/insights/articles/brazilian-games-marketconsumer-insights-brazils-mobile-players-are-likely-to-play-competitive-midcore-games>. Acesso em: 19 de Junho de 2023.

ONU. Objetivos de Desenvolvimento Sustentável. 2023. Disponível em:  
<https://brasil.un.org/pt-br/sdgs> Acesso em: 19 de Junho de 2023

Passos, L. J. P. e Novo Jr., J. E. F. Proposta de um role-playing audiogame acusmático para educação musical. Encontro de Educação Musical da Unicamp, online, 2021.

Pesquisa Game Brasil. 2022. Disponível em:  
<http://pesquisagamebrasil.rds.land/2022-painel-gratuitopgb22>. Acesso em: 19 de Junho de 2023

Petersen, K., Vakkalanka, S., Kuzniarz, L. Guidelines for conducting systematic mapping studies in software engineering : An update', Information and Software Technology, vol. 64, 2015.

PyGame. PyGame about. Disponível em: <https://www.pygame.org/wiki/about>  
Acesso em: 28 de Junho de 2023

Python. Python org. Disponível em: <https://www.python.org/> Acesso em: 27 de Junho de 2023

PythonTutor, 2023. Disponível em: <https://pythontutor.com/> Acesso em: 28 de Junho de 2023

SBC. Diretrizes para ensino de Computação na Educação Básica. (2017).  
Disponível em:  
<https://www.sbc.org.br/documentos-da-sbc/send/203-educacao-basica/1220-bncc-e-m-itinerario-informativo-computacao-2>. Acesso em: out. 2022.

Seehorn, D. & Clayborn, L. CSTA K-12 CS Standards for All. Proc. of the ACM SIGCSE Technical Symposium on Computer Science Education. Association for Computing Machinery, New York, NY, USA, 2017.

SOL, SBC. Atraindo Alunos do Ensino Médio para a Computação: Uma Experiência Prática de Introdução à Programação utilizando Jogos e Python, WORKSHOP DE INFORMÁTICA NA ESCOLA (WIE), Aracajú, 2011.

Spectrum, Stephen Cass. Top Programming Languages 2022, 2022. Disponível em: <https://spectrum.ieee.org/top-programming-languages-2022>. Acesso em: 30/11/2022

SuperData 2020 Year in Review. Disponível em:  
<https://www.digitalmusicnews.com/wp-content/uploads/2021/01/SuperData2020YearinReview.pdf> Acesso em: 19 de Junho de 2023

Tech With Tim. How to Make a Game in Python. Disponível em:  
<https://www.youtube.com/watch?v=waY3LfJhQLY> Acesso em: 28 de Junho de 2023

TIOBE. Tiobe index. Disponível em: <https://www.tiobe.com/tiobe-index/> Acesso em: 27 de Março de 2024

TwitchMetrics, The Most Popular Games on Twitch, June 2023. Disponível em: <https://www.twitchmetrics.net/games/popularity> Acesso em: 30 de Junho de 2023

Twitch TV. Disponível em: <https://www.twitch.tv/> Acesso em: 30 de Junho de 2023

Villela, Flávia. Celular é principal meio de acesso à internet no Brasil, mostra IBGE: Mais da metade dos 67 milhões de domicílios brasileiros passaram a ter. Agência Brasil, 6 abr. 2016. Disponível em:

<https://agenciabrasil.ebc.com.br/economia/noticia/2016-04/celular-e-principal-meio-d-e-acesso-internet-na-maioria-dos-lares> Acesso em: 09 jun. 2024

**Apêndice: Artigo do TCC**

# **Desenvolvimento de um Curso Online para Ensino de Python: Criação de Jogos no Ensino Médio**

**Thiago Silveira Bauer, Christiane Gresse von Wangenheim**

Departamento de Informática e Estatística, Universidade Federal de Santa Catarina,

Florianópolis, SC, Brasil

thiago.s.bauer@hotmail.com, c.wangenheim@ufsc.br

***Abstract.** Understanding algorithms and programming is essential to prepare students for the digital transformations of society. However, few initiatives in Brazil address programming education with Python in high school in a practical and interdisciplinary manner. This article presents the development of an online course for high school students, teaching basic programming concepts using Python and the PyGame library for game creation. The course follows the ADDIE methodology, integrating national curriculum guidelines and aligning with the UN's sustainability goals. As a result, the developed material supports educators and students, fostering a more contextualized, practical, and interdisciplinary learning experience.*

***Resumo.** A compreensão de algoritmos e programação é fundamental para preparar estudantes para as transformações digitais da sociedade. Contudo, poucas iniciativas no Brasil abordam o ensino de programação com Python no Ensino Médio de maneira prática e interdisciplinar. Este artigo apresenta o desenvolvimento de um curso online voltado para o Ensino Médio, que ensina conceitos básicos de programação utilizando Python e a biblioteca PyGame para criação de jogos. O curso segue a metodologia ADDIE, integrando as diretrizes curriculares nacionais e alinhando-se às metas de sustentabilidade da ONU. Como resultado, o material desenvolvido fornece suporte a educadores e estudantes, promovendo uma aprendizagem mais contextualizada, prática e interdisciplinar.*

## **1. Introdução**

A revolução digital transformou profundamente a sociedade, trazendo consigo a necessidade de formar indivíduos capazes de lidar com tecnologias emergentes. Nesse contexto, o ensino de computação, especialmente de algoritmos e programação, tornou-se essencial para preparar as novas gerações para os desafios e oportunidades do século XXI. O Ministério da

Educação (MEC) reconhece essa demanda e, por meio da Base Nacional Comum Curricular (BNCC), estabeleceu a inclusão do pensamento computacional como uma das competências fundamentais a serem desenvolvidas no Ensino Médio.

Python tem se destacado como uma das linguagens de programação mais utilizadas no mundo, especialmente por sua simplicidade, versatilidade e ampla gama de bibliotecas que tornam o aprendizado acessível para iniciantes. Entre essas bibliotecas, PyGame se sobressai como uma ferramenta poderosa para o desenvolvimento de jogos, permitindo que os alunos experimentem a aplicação prática da programação de forma interativa e envolvente. Apesar dessas vantagens, no Brasil, ainda são escassas as iniciativas voltadas ao ensino de Python para estudantes do Ensino Médio que combinam teoria e prática por meio de projetos concretos, como a criação de jogos.

Além disso, o ensino baseado em projetos alinhados a temas socialmente relevantes, como as metas de desenvolvimento sustentável das Nações Unidas (ONU), pode promover uma experiência de aprendizado interdisciplinar e significativa. Projetos dessa natureza não apenas facilitam a assimilação dos conceitos técnicos, mas também incentivam a reflexão crítica e a criatividade dos estudantes ao abordar problemas reais.

Diante desse cenário, este trabalho propõe o desenvolvimento de um curso online que utiliza a criação de jogos com Python como eixo central para o ensino de programação. O curso foi concebido para atender às necessidades dos alunos do Ensino Médio, considerando os desafios pedagógicos e estruturais enfrentados pelas escolas brasileiras. A abordagem adotada busca não apenas introduzir os conceitos fundamentais da programação, mas também promover o engajamento dos estudantes por meio de práticas interativas e contextualizadas.

## **2. Metodologia**

O desenvolvimento do curso foi realizado seguindo a abordagem ADDIE (Análise, Desenho, Desenvolvimento, Implementação e Avaliação), uma metodologia sistemática amplamente utilizada para design instrucional. Cada etapa foi detalhadamente planejada para garantir a efetividade do curso e sua adequação ao público-alvo.

Na etapa de análise, foram realizados estudos sobre os currículos de referência do MEC, ACM/CSTA e SBC para compreender as diretrizes pedagógicas relacionadas ao

ensino de computação no Ensino Médio. Também foi analisado o perfil dos estudantes e as condições de infraestrutura das escolas, identificando a necessidade de materiais acessíveis e ferramentas gratuitas. Essa etapa destacou o Python como uma linguagem ideal devido à sua simplicidade e praticidade, bem como a biblioteca PyGame, que possibilita o desenvolvimento de jogos de forma didática.

### 3. O Curso

O curso "Desenvolvendo seu Primeiro Jogo com Python no Ensino Médio – Shooter Mosquito" foi projetado especificamente para estudantes do Ensino Médio, considerando a análise do contexto do público-alvo e a elaboração do plano de ensino com base na revisão bibliográfica. Com esses resultados, o contexto dos estudantes foi avaliado e o material didático do curso foi desenvolvido.

#### 3.1 Objetivos de Aprendizagem

O curso foi projetado com objetivos claros, alinhados às competências gerais e específicas previstas pela BNCC e aos objetivos de aprendizagem da CSTA e MEC.

Tabela 1. Objetivos de Aprendizagem

| ID  | Objetivo de aprendizagem  | Referência(s)                                 |
|-----|---|---|
| OA1 | Identificar, decompor e resolver problemas a partir de soluções algorítmicas, reutilizando partes de soluções já existentes quando possível   | CSTA: 3A-AP-17;<br>3A-DA-13;<br>MEC:EM13CO01; |
| OA2 | Compreender e justificar a utilização de diferentes conceitos de programação, como eventos, listas, condicionais, estruturas de repetição, funções de entrada e saída, variáveis.     | CSTA<br>:3A-AP-14;3A-AP-15;<br>3A-AP-16;      |
| OA3 | Compreender as funcionalidades e aplicações da biblioteca PyGame para o desenvolvimento de jogos em Python  | CSTA:3A-AP-20;                                |
| OA4 | Avaliar e aperfeiçoar os artefatos e componentes desenvolvidos durante as aulas do curso, adicionando imagens, alterando cores, tipografia e ícones, a fim de aprimorar a usabilidade | CSTA: 3A-AP-21;<br>MEC:EM13CO02;              |
| OA5 | Identificar a presença e formas de uso das tecnologias utilizadas no mundo do trabalho.   | MEC:EM13CO09;                                 |
| OA6 | Compartilhar o jogo desenvolvido  | MEC:EM13CO20;EM13CO22;                        |
| OA7 | Lembrar a importância do combate ao mosquito da dengue para segurança e saúde pública   | (ONU, 2023)                                   |

#### 3.2 Design do Curso

O curso foi desenvolvido para iniciantes, com o objetivo de ensinar a criação de um jogo 2D em Python utilizando a biblioteca PyGame. O projeto escolhido é um *shooter game* com temática voltada para o combate ao mosquito da dengue, alinhando-se aos Objetivos de Desenvolvimento Sustentável (ODS) da ONU, especialmente no que tange à cobertura universal de saúde (ONU, 2023). Além disso, o curso possui caráter interdisciplinar, permitindo sua aplicação em itinerários do Ensino Médio, especificamente em Ciências da Natureza e suas Tecnologias, que englobam temas de Biologia, Física e Química (MEC, 2021), ou como uma atividade extracurricular.

### 3.3 Plano de Ensino

O plano de ensino foi elaborado com base nas diretrizes da BNCC (MEC, 2017) e nas competências destacadas pelo MEC (2017, 2022), SBC (2017) e CSTA (2017). O curso foi estruturado em 10 aulas sequenciais, que progridem em complexidade. Cada aula foi planejada com uma combinação de conteúdos teóricos e atividades práticas, incluindo exemplos guiados e desafios independentes. A integração entre as aulas foi projetada para garantir uma progressão lógica, permitindo que os alunos consolidem os conhecimentos e os apliquem em novas situações. Durante todo do curso, os estudantes desenvolvem um jogo completo no estilo shooter game.

Tabela 2. Plano de Ensino

| Aula (duração)                          | Conteúdo   | Objetivos de Aprendizagem | Método Instrucional                | Material Instrucional | Avaliação |
|---|--|---------------------------|------------------------------------|-----------------------|-----------|
| <b>Introdução</b>                       |  |                           |                                    |                       |           |
| 1 (15 min)                              | Aplicações e oportunidades com Python, Jogos com Python  | OA5;                      | Aula interativa                    | Slides, vídeo         | Quiz      |
| 2 (10 min)                              | Instalação do Python, Pycharm, visão geral sobre a biblioteca PyGame e configurações iniciais                                |                           | Aula prática                       | Slides, vídeo         | Quiz      |
| <b>Projeto do jogo shooter mosquito</b> |  |                           |                                    |                       |           |
| 3 (35 min)                              | Criação da primeira tela: comentários, variáveis, tipos primitivos, operadores lógicos, importações, estruturas de repetição | OA1; OA2; OA3;            | Aula interativa, atividade prática | Slides, vídeo         | Quiz      |
| 4 (30 min)                              | Criando o personagem jogável: ícones, imagens, música, estilização   | OA1; OA2; OA3; OA4        | Aula interativa, atividade prática | Slides, vídeo         | Quiz      |
| 5 (30 min)                              | Criando mecânica de movimentação: funções, argumentos, chamada de função, operadores lógicos e                               | OA1; OA2                  | Aula interativa, atividade prática | Slides, vídeo         | Quiz      |

|                    |   |               |                                    |               |      |
|--------------------|---|---------------|------------------------------------|---------------|------|
|                    | relacionais, estruturas de repetição, eventos   |               |                                    |               |      |
| <b>6 (10min)</b>   | Movimentação de inimigos  | OA1; OA2      | Aula interativa, atividade prática | Slides, vídeo | Quiz |
| <b>7 (22 min)</b>  | Comando de ação do personagem, estruturas condicionais, retorno de função, escopo de variáveis.   | OA1; OA2      | Aula interativa, atividade prática | Slides, vídeo | Quiz |
| <b>8 (15 min)</b>  | Detecção de colisão utilizando métodos matemáticos. Criação do placar da partida.                 | OA1; OA2; OA4 | Aula interativa, atividade prática | Slides, vídeo | Quiz |
| <b>9 (25 min)</b>  | Múltiplos inimigos simultâneos: funções com listas  | OA1; OA2; OA4 | Aula interativa, atividade prática | Slides, vídeo | Quiz |
| <b>10 (15 min)</b> | Compartilhando o projeto desenvolvido: criando executável e adicionando jogo em plataforma online | OA6; OA7      | Aula interativa, atividade prática | Slides, vídeo |      |

### 3.4 Material Didático

O material didático foi concebido para ser acessível, claro e prático, atendendo às necessidades específicas do público-alvo. As aulas do curso foram organizadas em um formato interativo, composto por apresentações que incluem vídeos (Figura 1), slides (Figura 2), atividades práticas (Figura 3), links úteis e materiais complementares, como código-fonte, imagens e áudios (Figura 4). Os vídeos combinam conteúdos teóricos apresentados por meio de slides com o desenvolvimento prático do jogo proposto, aplicando imediatamente os conceitos ensinados. Cada funcionalidade do jogo é inicialmente explicada de forma expositiva e, em seguida, implementada na prática, permitindo que os alunos consolidem os conceitos enquanto constroem o jogo passo a passo. Essa abordagem busca integrar teoria e prática de maneira dinâmica e eficiente.

Figura 1. Vídeos

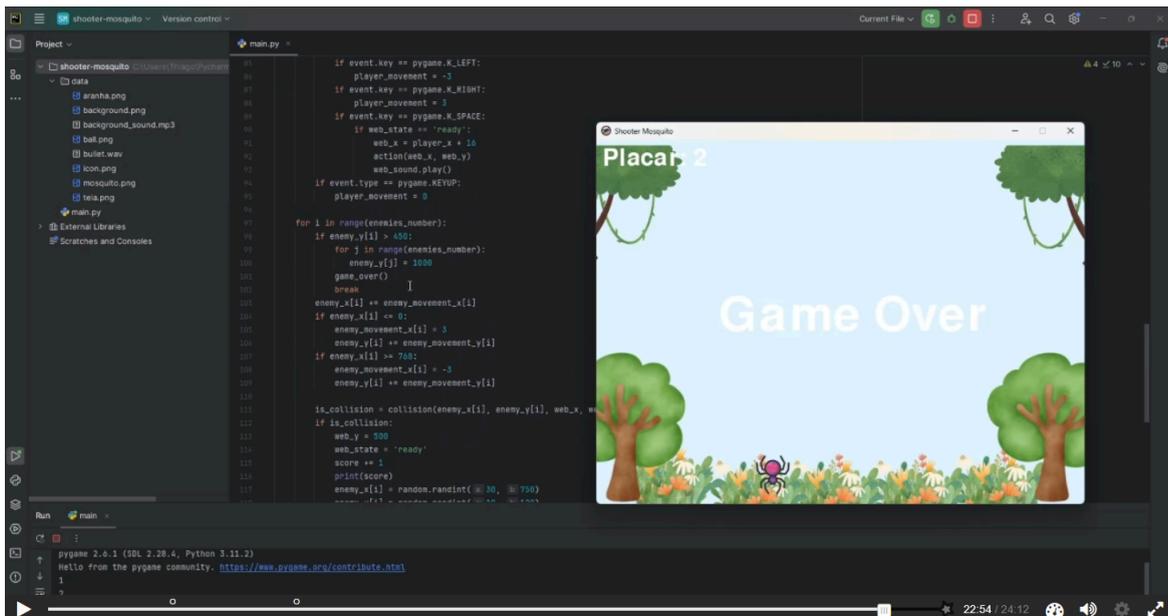


Figura 2. Slides

## Estruturas condicionais

Estruturas condicionais permitem que o programa tome decisões diferentes, de acordo com um valor fornecido. Por exemplo:

Se  $x$  for menor que 10, multiplique  $x$  por 2

Se  $x$  for maior que 10, divida  $x$  por 2

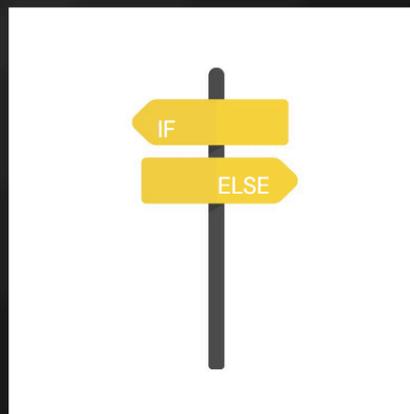


Figura 3. Atividades práticas

Qual das seguintes áreas é uma aplicação típica do Python?

- Análise de dados e Inteligência Artificial
- Desenvolvimento Web
- Desenvolvimento de Jogos
- Automação de tarefas
- Desenvolvimento de sistemas operacionais

 Verificar

Figura 4. Arquivos

- ▼ Arquivos
  -  [Mídias utilizadas no projeto](#) 2.1 Mb Imagem (PNG)
  -  [Jogo completo](#) 36.6 Mb Archive (ZIP)
- ▼ Aula 1
  -  [Aula 1](#)
  -  [Aula 1 - Slides](#)
- ▼ Aula 2
  -  [Aula 2](#)
  -  [Aula 2 - Slides](#)

## **5. Conclusão**

O curso desenvolvido demonstrou potencial para suprir lacunas no ensino de programação no Ensino Médio, integrando a aprendizagem técnica com projetos práticos e relevantes. Ao utilizar Python e PyGame como ferramentas, o curso ofereceu uma abordagem acessível e motivadora para introduzir conceitos de programação.

A aplicação do curso pode ampliar as oportunidades de ensino de computação em escolas brasileiras, promovendo o pensamento computacional e o engajamento dos alunos. Como trabalhos futuros, sugere-se a implementação do curso em diferentes contextos escolares e a avaliação do impacto no aprendizado dos estudantes.

## **Referências**

MEC. Base Nacional Comum Curricular. 2017. Disponível em: <https://www.gov.br/mec>.

Python. Documentação Oficial. 2023. Disponível em: <https://www.python.org>.

PyGame. Documentação Oficial. 2023. Disponível em: <https://www.pygame.org>.

BRANCH, R. M. Instructional Design: The ADDIE Approach. Springer, 2009

## **Apêndice 2. Repositório do código fonte**

<https://codigos.ufsc.br/thiago.s.bauer/shooter-mosquito/-/blob/main/shooter-mosquito.zip>