

SCRATCH E PENSAMENTO COMPUTACIONAL NO ENSINO SUPERIOR: RELATO DE PRÁTICA COM ESTUDANTES DE BIOLOGIA E COMPUTAÇÃO

 <https://doi.org/10.56238/sevened2025.018-023>

Vanessa Stangherlin Machado Paixão-Cortes

Universidade Federal do Rio Grande

Doutora em Ciência da Computação

vanessapaixaocortes@gmail.com

Cristiane Ellwanger

Univ. Fed. do Rio Grande do Sul - UFRGS - Porto Alegre - RS

Doutora em Design

Regina Barwaldt

Universidade Federal do Rio Grande

Doutora em Informática na Educação

RESUMO

A formação de profissionais em Ciências Biológicas requer o uso de métodos computacionais para a resolução de problemas complexos, o que demanda uma abordagem interdisciplinar. Nesse contexto, o pensamento computacional pode contribuir significativamente para o desenvolvimento de competências e habilidades, embora ainda se observe uma carência de sua abordagem na graduação. Este estudo relata a experiência de um Projeto de Ação na Escola (PAE), realizado no curso (ocultado para submissão), com o objetivo de investigar se a programação com jogos e animações digitais voltados à lógica de programação pode atuar como fator de engajamento para estudantes das Ciências Biológicas ou se permanece como privilégio das áreas exatas e tecnológicas. Participaram 12 estudantes, sendo sete da área biológica e cinco das exatas, com idades entre 18 e 30 anos e experiências variadas em programação. As atividades envolveram o desenvolvimento de jogos e animações na linguagem Scratch, com base no modelo motivacional ARCS, que orientou a análise a partir das categorias de Atenção, Relevância, Confiança e Satisfação. Os resultados indicaram que o pensamento computacional pode atuar como estratégia motivacional e representar uma alternativa viável para o engajamento no ensino-aprendizagem no ensino superior, tanto para iniciantes em computação quanto para estudantes das Ciências Biológicas.

Palavras-chave: Pensamento computacional. Linguagem de programação Scratch. Ciências Biológicas. Ensino Superior.

1 INTRODUÇÃO

O ensino de programação não é uma prática exclusiva aplicada a cursos da Computação e Engenharias (Zanetti, Borges, & Ricarte, 2016). Neste mesmo contexto, Denning (2018) corrobora que a computação provou ser produtiva para o avanço científico em praticamente todos os campos do conhecimento. Por exemplo, os avanços mais recentes nas Ciências Biológicas envolveram a modelagem, o sequenciamento e a edição de DNA¹ utilizando-se de métodos computacionais e programação.

Atualmente, observa-se a necessidade que alunos no ensino superior em Ciências Biológicas aprendam também a programar e desenvolver seu próprio *software* para a resolução de problemas biológicos e em saúde (Salazar, 2018), sendo essencial a capacidade de empregar métodos científicos em computação e aplicar metodologias em Engenharia de *Software* para projetar, implementar e manter com sucesso *softwares* em ambientes científicos (Welch, Schwartz, & Lewitter, 2011).

A fim de preparar os alunos para enfrentarem os desafios biológicos do século XXI, cultivarem seus interesses em carreiras científicas e melhorarem suas habilidades de pensamento quantitativo e crítico é essencial estimular o pensamento computacional (Qin, 2009). Entretanto, estudos apontam que apesar do interesse em trabalhar o pensamento computacional em nível fundamental e médio, em áreas distintas à Ciência da Computação, há menos empenho na pesquisa para verificar seu potencial no ensino superior (Czerkawski & Lyman, 2015). Ainda, o treinamento em computação em nível superior em Ciências Biológicas não supre a demanda por conhecimentos e habilidades computacionais necessárias para a prática efetiva da pesquisa científica em suas carreiras (Salazar, 2018). O ensino focado somente na alfabetização digital não é mais suficiente, pois se faz necessário ensinar a proficiência digital, incluindo a programação (Wangenheim, Nunes, & Santos, 2014).

É neste contexto que esse artigo apresenta e discute as experiências práticas de um Projeto Ação na Escola (PAE), em um curso de pós-graduação (ocultado para submissão) em nível de especialização. O objetivo principal do PAE foi investigar se alunos de graduação, principalmente biólogos, se sentem engajados a aprender esses conceitos por meio de jogos e animações, utilizando o Scratch². A amostra foi composta por doze alunos, sendo seis participantes da área das Ciências Biológicas e cinco alunos das áreas das Exatas.

A temática para aplicação do PAE emergiu das aprendizagens obtidas no curso (ocultado para submissão) sobre as transformações e inovações tecnológicas vivenciadas na nova sociedade digital e pelos progressos educacionais oriundos dessa evolução, que motivou as autoras a repensarem suas práticas pedagógicas em sala de aula, refletirem sobre as contribuições propiciadas pelo uso das

¹ Ácido desoxirribonucleico

² <https://scratch.mit.edu/>

tecnologias digitais em âmbito acadêmico e identificarem as possibilidades de mudanças educacionais que poderiam ser evidenciadas na aplicação do PAE na graduação.

A metodologia do PAE envolveu coleta de dados obtida por meio de questionários e observação de uso direta, onde o modelo ARCS (Attention, Relevance, Confidence, Satisfaction) proposto por Keller (2009) foi utilizado para nortear o desenvolvimento das tarefas, a concepção dos instrumentos e realizar a análise, que teve como suporte abordagens quantitativa e qualitativa. Como resultado percebeu-se que o Scratch pode proporcionar experiências engajadoras para desenvolvimento do pensamento computacional por meio de jogos e animações interativas para alunos iniciantes ou de áreas como áreas de Ciências Biológicas. Durante a análise, percebeu-se que a maioria dos participantes se sentiram motivados, inclusive aqueles que já conheciam mais profundamente as linguagens de programação.

2 FUNDAMENTAÇÃO TEÓRICA

Atualmente, biólogos necessitam experienciar as práticas computacionais para realizar suas análises. Embora implementar seus próprios programas não é uma atividade trivial e depende da sua capacidade lógica e do entendimento de algoritmos e linguagens de programação. Os indivíduos passam a ter não só a capacidade de desenvolver os seus próprios sistemas, mas reforçam competências adjacentes, como o pensamento abstrato (utilização de diferentes níveis de abstração para perceber os problemas e, passo a passo, solucioná-los), o pensamento algorítmico (expressão de soluções em diferentes passos de forma a encontrar a forma mais eficaz e eficiente de resolver um problema), o pensamento lógico (formulação e exclusão de hipóteses) e o pensamento dimensionável (decomposição de um grande problema em pequenas partes ou composição de pequenas partes para formular uma solução mais complexa) (Phillips, 2009; Resnick, 2008, *apud* Sousa & Lencastre, 2014, p.261).

Autores como (Czerkowski & Lyman, 2015; Qin, 2009), ainda abordam o pensamento computacional como uma maneira de pensar que utiliza conceitos e metodologias de computação, como o desenvolvimento da lógica, do pensamento algorítmico, do pensamento recursivo, da abstração, do pensamento paralelo, da correspondência de padrões e os processos relacionados, para abordar questões em uma ampla gama de assuntos e, como tal, oferece um conjunto importante de habilidades nas ciências modernas.

Segundo Marques *et al.* (2011) ensinar programação desde cedo é importante porque os alunos desenvolvem competências de resolução e formalização de problemas, que são úteis em suas respectivas áreas do conhecimento. Nesta perspectiva de aplicações tecnológicas, destaca-se o potencial dos jogos educativos e animações para motivar e dinamizar o processo de ensino e aprendizagem em diferentes contextos, pois permitem uma exploração prática dos conceitos, um

aprendizado lúdico e motivador e servem como complemento aos conceitos teóricos (Soares & Rodrigues, 2016).

As animações, realizadas a partir de simulações de experimentos, histórias e atividades interativas, torna mais acessível o processo de compreensão conceitual pelos alunos, que são fatores motivacionais e estimulantes na busca do conhecimento (Castilho & Ricci, 2006).

Os jogos são instrumentos eficientes pois divertem enquanto motivam, facilitam o aprendizado e aumentam a capacidade de retenção do que foi ensinado, exercitando as funções mentais e intelectuais dos alunos, que revelam a autonomia, criatividade, originalidade e possibilidade de simular e experimentar situações (Tarouco, Roland, Fabre, & Konrath, 2004). Além disso, os jogos fazem parte do cotidiano dos alunos, conforme destaca Almeida e Silva (2011, p.3):

Tais tecnologias passaram a fazer parte da cultura, tomando lugar nas práticas sociais e ressignificando as relações educativas[...] Dentre os artefatos tecnológicos típicos da atual cultura digital, com os quais os alunos interagem mesmo fora dos espaços da escola, estão os jogos eletrônicos, que instigam a imersão numa estética visual da cultura digital; as ferramentas características da Web 2.0, como as mídias sociais apresentadas em diferentes interfaces; os dispositivos móveis, como celulares e computadores portáteis, que permitem o acesso aos ambientes virtuais em diferentes espaços e tempos.

Destarte, jogos e animações favorecem a criação de experiências de aprendizagem mais engajadoras, assim como fomentam a prática e valorizam o esforço e o erro como parte do aprendizado, aspectos importantes para o ensino em programação (Gomes, Tedesco, & Melo, 2016).

Por último, para este estudo considerou-se engajamento como uma área que compreende os aspectos relacionados a fidelidade na fluência das interações e da motivação dos usuários com o sistema de maneira geral, apresentando aspectos que estabelecem objetivamente o envolvimento dos usuários por meio de fatores como atenção, afeto positivo, novidade de experiências, controle, expectativa e interesse do usuário (Herpich & Tarouco, 2016). Também, o engajamento está relacionado com a satisfação, atenção, confiança, relevância, bem como a forma que o aluno realiza com êxito uma tarefa e como este sucesso afeta suas expectativas para o futuro (Keller, 2009).

2.1 TRABALHOS CORRELATOS

Com o intuito de fornecer uma visão sobre as pesquisas que estão sendo desenvolvidas dentro dessa temática, a seguir são descritos trabalhos que envolvem o ensino do pensamento computacional e programação em variados níveis educacionais, para então, apresentar estudos no escopo das Ciências Biológicas.

Zanetti *et al.* (2016) afirma que é possível encontrar práticas de ensino de programação dentro das escolas do ensino fundamental e médio. Este autor realizou uma pesquisa de revisão sistemática da literatura sobre Pensamento Computacional no ensino de programação, de estudos publicados em eventos brasileiros no intervalo de 2012 até 2015. A revisão foi realizada nos anais de importantes

eventos nacionais: Simpósio Brasileiro de Informática na Educação (SBIE), Workshop de Informática na Escola (WIE) e Workshop sobre Educação em Informática (WEI), Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WAlgProg). O autor analisou 16 artigos e identificou que são utilizados a computação “desplugada”, jogos digitais, linguagens de programação, linguagem de programação visual e robótica pedagógica., sendo que alguns estudos utilizam essas práticas em conjunto, e que são desenvolvidas habilidades como coleta, análise e representação de dados, decomposição de problemas, abstração e desenvolvimento de algoritmos. A ferramenta comumente utilizada nos estudos foi o Scratch por ser difundida no meio acadêmico e apresentar resultados positivos para aprendizagem e o uso de jogos digitais aparece como um instrumento de motivação aos alunos, pelo caráter lúdico, desafiador e motivador. Por último, o autor menciona que há uma tendência para a realização de pesquisas no contexto do ensino médio/técnico (Zanetti *et al.*, 2016).

Neste sentido, os estudos de Wangenheim *et al.* (2014) e Alves *et al.* (2016) investigaram o contexto do pensamento computacional em nível de ensino fundamental com o Scratch. Ambas pesquisas propuseram uma unidade instrucional multidisciplinar para o ensino de computação, inserida na disciplina de História e Estudos Sociais. As unidades instrucionais foram aplicadas e avaliadas em estudos de caso, que envolveram turmas do primeiro, quinto e sétimos anos, em uma escola em Florianópolis/SC. As implicações observadas indicam que a unidade instrucional e o uso do Scratch despertam o interesse e motivação dos alunos para a área da computação, assim como permitem a aprendizagem de conceitos básicos de programação, de forma eficaz e divertida por alunos do ensino fundamental (Alves *et al.*, 2016).

Os estudos constataram que é interessante que haja um maior engajamento de alunos de Ciências Biológicas na aprendizagem de programação. Por exemplo, o estudo de Qin (2009) incorporou o ensino da computação em um curso de bioinformática nível de introdução para graduandos em Ciências da Vida, com atividades práticas de laboratório e aprendizado colaborativo por pares. A metodologia envolveu práticas com uso de *softwares* de bioinformática, como MEGA para filogenia, SWISS PDB Viewer para análise de estrutura de proteínas e R para análise de expressão gênica, e a utilização de servidores Linux. O autor traz a realização de duas pesquisas com um total de 39 alunos e aponta que muitos desses estudantes expressam prontamente seus interesses em melhorar suas habilidades computacionais e quantitativas, e que, principalmente, concordam que deve haver um melhor desenvolvimento dessas competências, que são úteis para sua profissão, entretanto poucos deles realmente fazem cursos de computação (Qin, 2009). A pesquisa mostrou que os alunos foram positivos sobre sua experiência de aprendizado e que as atividades melhoraram seus conhecimentos em informática e suas habilidades quantitativas.

O trabalho de Salazar (Salazar, 2018) compreendeu o desenvolvimento de um projeto, relacionado ao ensino e extensão, com o tema de programação e boas práticas de computação científica para biólogos³. Foram elaborados materiais didáticos para programação, envolvendo interfaces de linha de comando e programação em Python, acompanhado de um questionário semi-estruturado de avaliação, incluindo perguntas sobre a temática e o uso de *software* livre. Foi realizada uma edição de um minicurso, aplicando os instrumentos, para 17 participantes. As reflexões que nascem dos resultados apontam nitidamente para a necessidade de pesquisas, na área de educação, a respeito do ensino de programação para estudantes de Ciências Biológicas e áreas afins, sendo que maioria dos sujeitos (14 de 17 participantes) concordaram que conteúdos como este são importantes, porém majoritariamente ausentes na formação de biólogos no Brasil.

Entre os estudos que dialogam com a proposta deste artigo, destaca-se a pesquisa de Santana (2023), que investigou como estudantes de Licenciatura em Ciências Biológicas, em uma universidade pública do Nordeste brasileiro, puderam desenvolver habilidades baseadas no Pensamento Computacional (PC) por meio da programação em blocos com o Scratch. A partir de uma sequência didática com sete momentos, que integravam conteúdos de biofísica, conceitos de PC e atividades práticas na plataforma, os participantes desenvolveram projetos sobre temas como estruturas e funcionamento das membranas biológicas. O estudo evidenciou que a habilidade de PC mais desenvolvida foi o pensamento algorítmico, e concluiu que o uso de programação em blocos pode favorecer a aprendizagem significativa em conteúdos interdisciplinares e de alta complexidade, como os que envolvem química, física e biologia. A pesquisa também destacou como limitação o tempo reduzido da sequência didática e sugeriu a criação de uma disciplina específica sobre Pensamento Computacional no curso de Licenciatura, reforçando a necessidade de formação docente voltada à integração entre computação e ensino de ciências.

3 METODOLOGIA

A natureza metodológica do presente estudo foi uma pesquisa aplicada sobre uma experiência prática e avaliativa de uma ferramenta para o ensino do pensamento computacional com foco em alunos de graduação com a realização de um Projeto Ação na Escola (PAE). A proposta do presente trabalho partiu da seguinte premissa: existe uma carência, no campo da biologia e áreas afins, de treinamento de programação e computação científica, especialmente a nível de ensino superior (Salazar, 2018).

A Linguagem Scratch foi utilizada para implementar os objetos em decorrência da sua interface ser amigável, gratuita, simplificada e permite a aprendizagem da sintaxe básica em programação de forma rápida e facilitada (Alves *et al.*, 2016).

³ Atividades realizadas no Centro de Ciências Biológicas da Universidade Federal de Santa Catarina (CCB / UFSC)

O objetivo de pesquisa se constitui como descritivo, pois se utiliza de coleta de dados (questionário e observação direta de uso do sistema). Questionários especializados são utilizados para a identificação de aspectos subjetivos dos usuários, como seu engajamento com a tecnologia, onde as perguntas só são efetuadas ao final da aplicação (Cybis, Betiol, & Faust, 2015).

E por fim, para validar o trabalho, levou-se em consideração que a pesquisa se constituiu de caráter misto, porém com foco maior em dados qualitativos. Segundo Preece, Rogers e Sharp (2013) o tipo de análise pode ser influenciado pelos objetivos identificados no início da pesquisa e pelos dados realmente coletados, podendo ser identificado uma abordagem de análise qualitativa ou quantitativa, ou uma combinação de ambos.

3.1 CARACTERIZAÇÃO DO LOCAL DE PESQUISA, AMOSTRA E INSTRUMENTOS DE COLETA DE DADOS

A amostra foi composta por doze alunos, sendo seis participantes da área das Ciências Biológicas e cinco alunos das áreas das Exatas. A temática para aplicação do PAE emergiu das aprendizagens obtidas no curso sobre as transformações e inovações tecnológicas vivenciadas na sociedade digital e pelos progressos educacionais oriundos dessa evolução, que já possuíam uma expertise prévia na sintaxe e estruturas que compõe linguagens de programação, como C, Python, JAVA, JavaScript e PHP, assim como conhecimentos de lógica para implementar um algoritmo.

Como avaliação se os alunos se sentiram engajados com o desenvolvimento de jogos e animações por meio da programação com Scratch foram elaborados dois instrumentos de teste, questionários confeccionados no *Google Forms*, que foi submetido em dois momentos: no início do projeto e em sua conclusão. As etapas realizadas para concepção da coleta de dados são descritas na Tabela 1.

Tabela 1: Etapas para execução da coleta de dados

Etapas	Respostas
Definição da amostra de participantes	Usuários finais: foram convidados a participar da aplicação do PAE dos alunos do curso de graduação da (ocultado para submissão). A amostra, não probabilística por conveniência, foi composta por 12 alunos de graduação oriundos das Ciências Exatas ou Biológicas.
Definição do roteiro	As tarefas foram realizadas segundo um roteiro pré-estabelecido baseado na aplicação da metodologia de Ensino por Projetos, onde o aluno teve a possibilidade de escolha em um tema de interesse para o desenvolvimento de seu jogo/animação.
Definição do ambiente	Os testes foram aplicados em um laboratório de informática da instituição onde os mesmos eram pertencentes, apoiados por mídias de apresentações e formulários eletrônicos, em computadores com acesso à internet.
Constituição da equipe de avaliadores	A aplicação do PAE foi executada por um professor/observador, nesta pesquisa considerada como autora.
Definição da técnica de registro dos acontecimentos	Os registros foram realizados via gravação de imagens das respostas dos usuários.
Testes - piloto	Realização de pré-testes, onde um especialista da área de ciência da computação, especialista em IHC e com experiência no ensino aprendizagem, validou o instrumento de teste. Também, mensurou-se o tempo necessário para execução das tarefas.

Execução dos testes de coleta de dados	Esta etapa foi a de aplicação do PAE, também foram tratados as questões éticas, com o termo de consentimento Livre e Esclarecido, e no caso da Instituição, a carta para o conhecimento do responsável pela realização da pesquisa.
--	---

Fonte: o autor. Adaptado de Cybis, Betiol, & Faust (2015).

Para o desenvolvimento e análise do instrumento de coleta de dados utilizou-se como referencial teórico um modelo que favorece o processo avaliativo de jogos e animações educativas. Segundo Soares & Rodrigues (2016) o modelo ARCS proposto por Keller (2009) foi adequado pois tem foco na avaliação da interação dos alunos com os materiais e ambientes de aprendizagem, sendo derivado da teoria expectativa (probabilidade de um indivíduo obter sucesso) e é baseado no valor (satisfação e motivação do usuário).

Neste sentido, as atividades propostas para o desenvolvimento do PAE e questionários buscaram angariar subsídios que pudessem estar relacionados com a motivação dos educandos na aprendizagem. Assim, as tarefas e instrumentos tiveram como embasamento teórico o método ARCS, conforme a Tabela 2. O método ARCS é dividido em 4 categorias, as quais permitem uma visão geral das principais dimensões motivacionais, especialmente no contexto da motivação na aprendizagem, e como criar estratégias para estimular e sustentar a motivação (Keller, 2009).

Tabela 2: Método ARCS utilizado para nortear as tarefas e os instrumentos de coleta de dados.

Método ARCS	Descrição	Questionamento Norteador	Tarefas Relacionadas
Atenção	Estimular o interesse dos alunos, e a curiosidade de aprender	Como posso tornar essa experiência de aprendizado estimulante e interessante?	Aplicação do questionário pré-teste Apresentação do Tema Motivador Atividades de programação com Scratch Desenvolvimento do Projeto Final
Relevância	Atender às necessidades e objetivos pessoais do aluno para efetivar uma atitude positiva	De que maneira essa experiência de aprendizado será valiosa para meus alunos?	Apresentação dos Conceitos Básicos Algoritmo da Amizade Atividades de programação na Linguagem Scratch
Confiança	Motivar os alunos a acreditar e sentir que terão sucesso	Como posso via instruções ajudar os alunos a ter sucesso e permitir que eles controlem o sucesso deles?	Atividades de programação na Linguagem Scratch Desenvolvimento do Projeto Final
Satisfação	Reforçar a realização dos alunos, com sentimentos positivos após experiência de aprendizagem	O que posso fazer para ajudar os alunos se sentirem bem com sua experiência e desejo continuar aprendendo?	Desenvolvimento do Projeto Final Aplicação do questionário pós-teste

Fonte: o autor. Adaptado de Keller (2009).

Após a coleta de dados, iniciou-se com as observações para buscar a identificação de padrões, utilizando-se de *frameworks* estruturados ou de teorias para apoiar a investigação (Preece *et al.*, 2013), alicerçado no método ARCS, fazendo uso das experiências vivenciadas pelos alunos, falas das respostas qualitativas e captura das interfaces dos projetos realizados. Este tipo de análise foi popularizada a partir de Bardin (1977) nas entrevistas e/ou observado pelo pesquisador.

Nesta mesma linha do pensamento o desmembramento do texto em categorias agrupadas analogicamente se respalda em uma melhor escolha quando se quer observar valores, opiniões, atitudes

e crenças por meio de dados qualitativos (Lidiane Vieira Pozzebon, André Ricardo Theodoro Velho, & Regina Barwaldt, 2019). Este estudo ainda apresenta e discute os resultados das respostas quantitativas.

Dessa forma, a coleta de dados serviu de base para discussão sobre a aplicação do PAE, verificação da aprendizagem dos alunos e avaliação das atividades propostas para, enfim, inferir impactos da ação para o grupo de estudantes.

3.2 DESCRIÇÃO DAS ATIVIDADES

Para o desenvolvimento do PAE, foram realizadas duas oficinas, cada uma com 3 horas de duração, o que totalizou 6h de atividades. Cada uma das oficinas contou com um público diferente, sendo a primeira com 5 alunos e a segunda com 7 alunos. Ambas atividades seguiram o mesmo protocolo de execução.

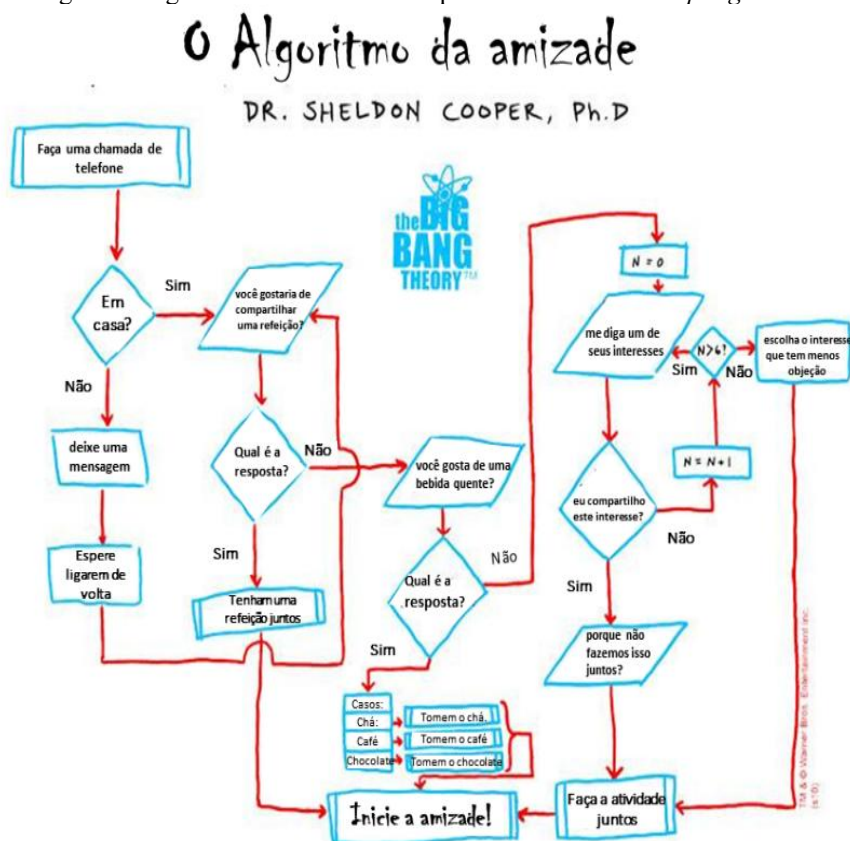
As etapas iniciais envolveram a apresentação da professora ministrante da oficina e assinatura do Termo de Consentimento Livre e Esclarecido (TCLE), que orientava os participantes sobre as questões éticas, objetivos da pesquisa e potenciais riscos. Após, foi realizado o preenchimento do questionário (pré-teste) sobre seu perfil e experiência com programação. Para engajamento dos alunos, foi mostrado o vídeo intitulado “Por que todos deveriam aprender a programar?⁴” e foram discutidos alguns aspectos atuais da tecnologia e a importância do pensamento computacional. Por exemplo, durante a introdução temática houve a apresentação do Algoritmo da Amizade⁵ (Figura 1).

Após a motivação inicial, foram introduzidos conceitos de lógica de programação e pensamento computacional necessários para o aluno entender o contexto das atividades que seriam realizadas. Esses conceitos ofereceram uma abordagem introdutória e menos complexa, abrangendo conceitos como pensamento abstrato, decomposição, reconhecimento de padrões, algoritmos e linguagens de programação.

⁴ <https://www.youtube.com/watch?v=mHW1Hsqli6A>

⁵ Da série The Big Bang Theory. Nele, o personagem fictício Sheldon desenvolve um algoritmo para fazer amizade e ao apresenta a seus amigos por meio de um diagrama de blocos (ou fluxograma), que é uma forma de escrever algoritmos graficamente.

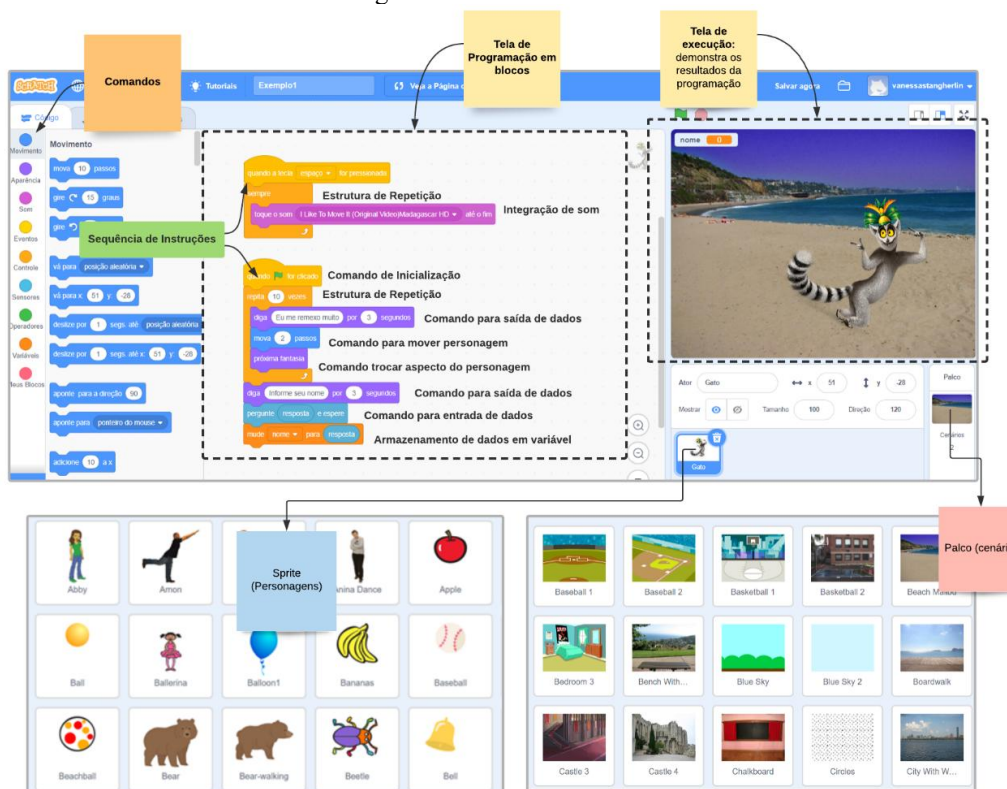
Figura 1: Algoritmo da Amizade. Adaptado de *The Friendship algorithm*⁶.



A próxima etapa foi a apresentação da interface do Scratch (Figura 2), com a demonstração dos comandos, personagens e cenários. Linguagens neste formato têm como diferencial proporcionar conhecimentos em programação sobre: (i) sequências de instruções, (ii) condicionais, (iii) laços, (iv) procedimentos/ funções e (v) parâmetros (Gomes *et al.*, 2016).

⁶ <https://link.estadao.com.br/blogs/alexandre-matias/wp-content/uploads/sites/360/2010/08/algoritmo-da-amizade.jpg>

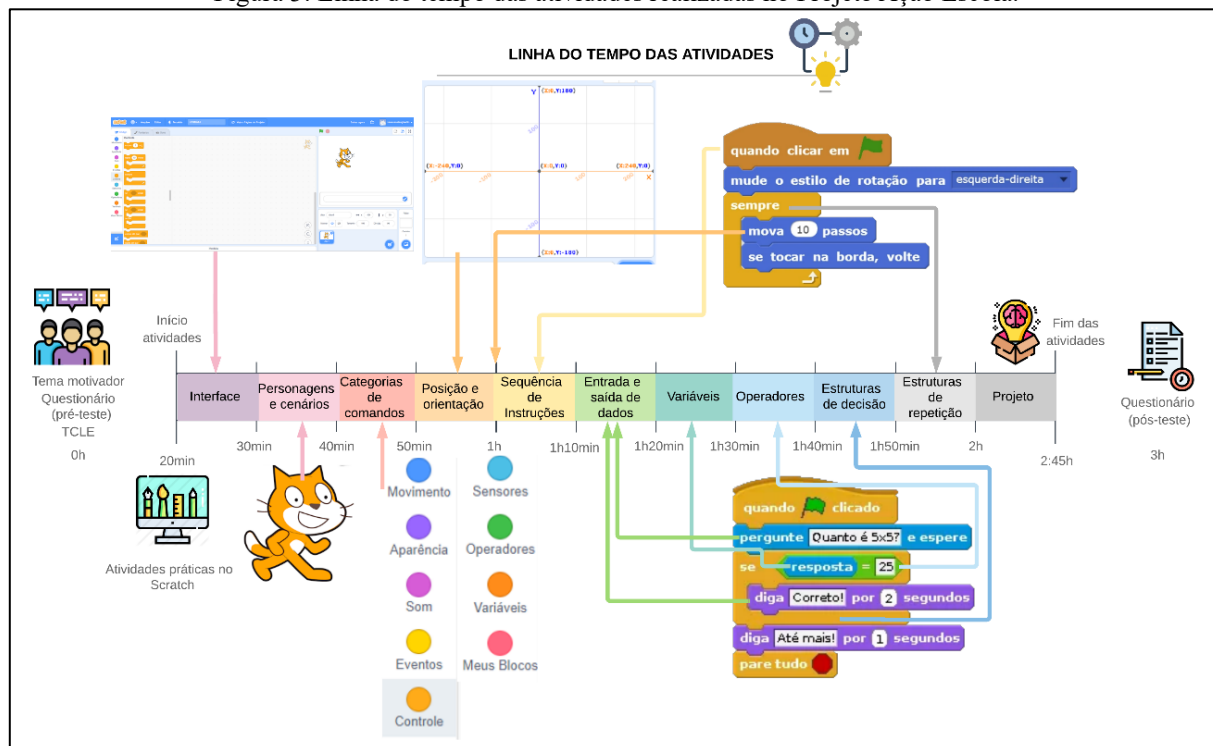
Figura 2: Interface do Scratch



Os alunos puderam explorar o ambiente scratch aprendendo conceitos de lógica de programação na seguinte sequência, conforme a Figura 3:

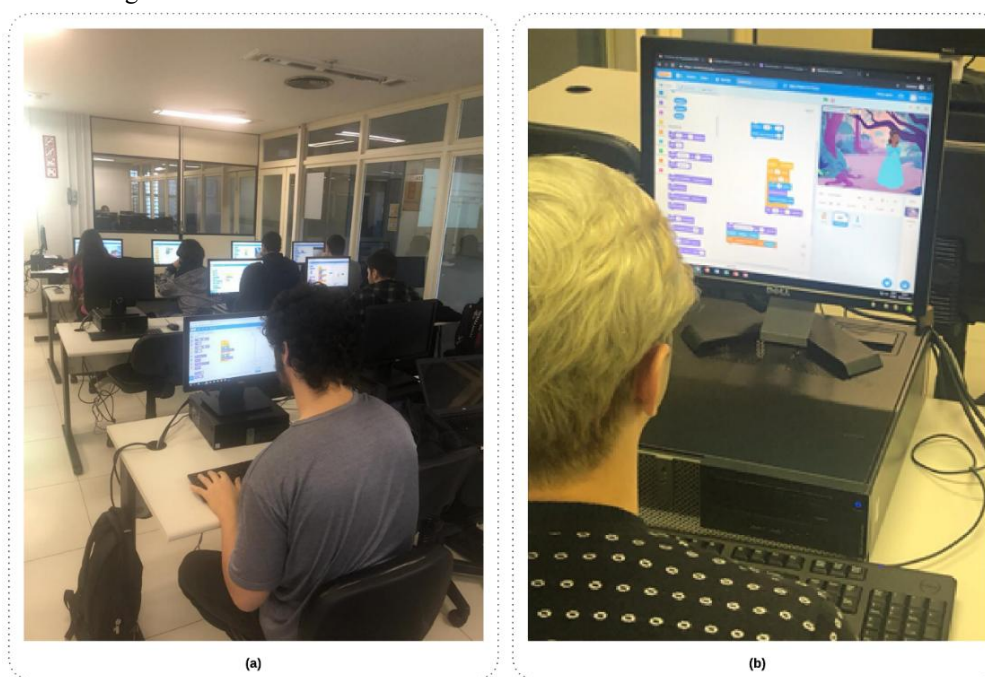
- Exploração da interface, incluindo abrir e salvar um projeto.
- Verificação de personagens e dos cenários.
- Modificação da posição e orientação de objetos, movimentação por passos e por graus e exploração do plano cartesiano.
- Interação com as categorias de comandos: movimento, aparência, som, controle, sensores, operadores e variáveis.
- Utilização dos blocos como uma sequência de comandos, incluindo iniciar e parar o algoritmo.
- Emprego de entrada e saída de dados.
- Criação e manipulação de variáveis.
- Utilização de operadores lógicos.
- Conceituação e desenvolvimento de estruturas de seleção e repetição;

Figura 3: Linha do tempo das atividades realizadas no Projeto Ação Escola.



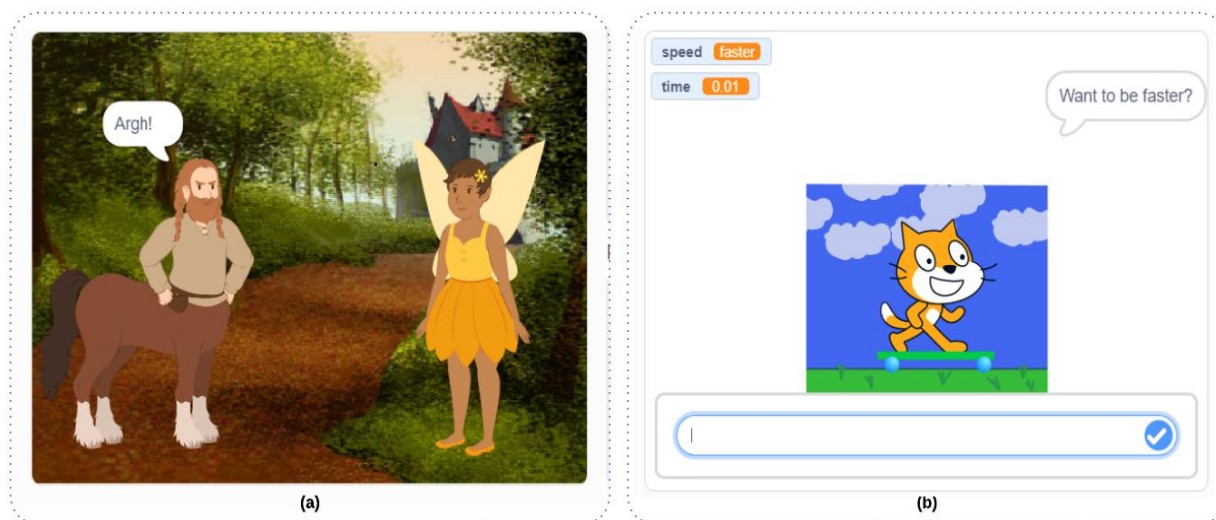
Para cada participante, solicitou-se que utilizasse dos conceitos aprendidos a fim de desenvolver um projeto de sua autoria, que envolvesse uma animação ou jogo. A Figura 4 mostra a realização de uma das oficinas de uma história aplicada no projeto.

Figura 4: - Oficina de aplicação do PAE. (A) Participante de uma das Oficinas. (b) Projeto desenvolvido por participante em área de ciências biológicas.



Por último, a Figura 5 ilustra dois projetos desenvolvidos pelos alunos: uma história com diálogo entre personagens (Figura 5a); e uma animação com interação, onde o personagem principal consegue se mover segundo as informações fornecidas pelo usuário (Figura 5b).

Figura 5: - Projetos realizados pelos alunos. (A) História com dois personagens. (b) Animação com entrada de dados para controlar a velocidade do personagem.



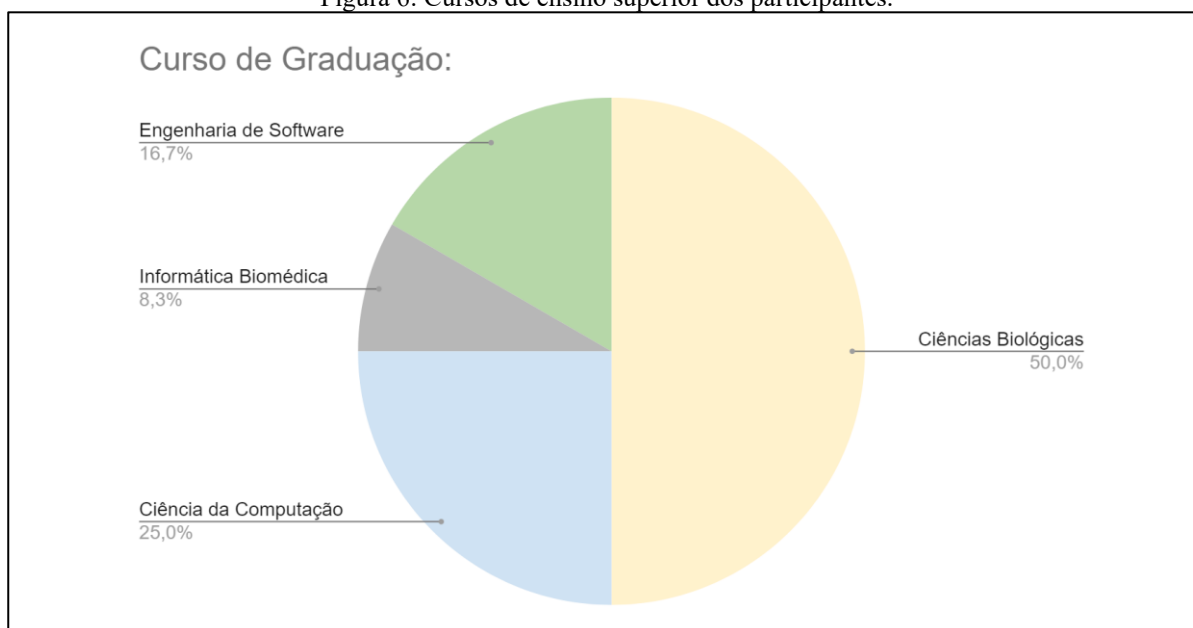
4 ANÁLISE DOS DADOS E RESULTADOS ALCANÇADOS

Esta seção relata os resultados obtidos ao final da experiência prática do PAE por meio da análise quantitativa e qualitativa dos instrumentos de coleta de dados. A pesquisa quantitativa buscou mensurar numericamente os significados dos fenômenos estudados, ou seja, traduzir em números as opiniões e informações para classificá-las e analisá-las mediante o uso de recursos e/ou técnicas estatísticas (Schneider, Fujii, & Corazza, 2017), enquanto a pesquisa qualitativa tem por objetivo dar sentido e significância aos achados, observando e interpretando a realidade estudada, utilizando-se de procedimentos metodológicos diversificados, que buscam explicações para gerar comparabilidade ou exemplaridade (Pádua, 2019).

Neste trabalho os resultados quantitativos são derivados das respostas às questões fechadas e os qualitativos são obtidos pela apreciação crítica da observação de uso, questões abertas e projetos produzidos pelos alunos.

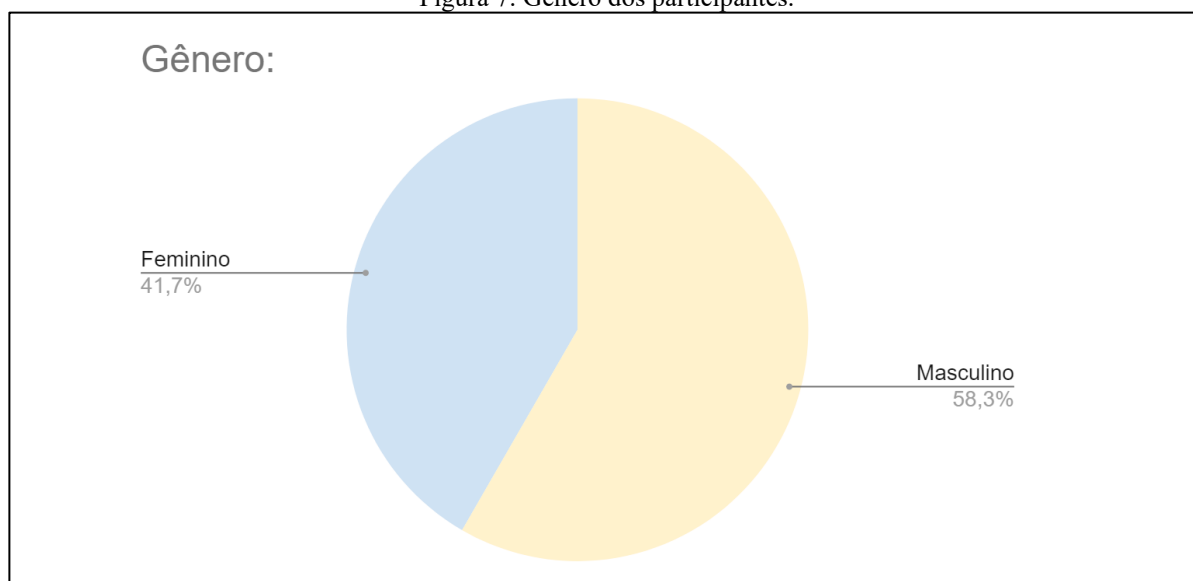
As Figuras 5,6,7,8 ilustram o perfil dos participantes e referem-se ao questionário pré-teste. Na área biológica participaram sete alunos (seis em ciências biológicas e um em informática biomédica), enquanto na área exata participaram cinco alunos (três em ciência da computação e dois em engenharia de software), compondo uma amostra total de doze participantes (Figura 6). Entretanto no decorrer do trabalho, um participante que apesar de ter participado e realizado as atividades propostas, não realizou o preenchimento do questionário pós-teste e não pode ser incluído na etapa de análise.

Figura 6: Cursos de ensino superior dos participantes.



Com relação ao gênero, sete participantes eram do sexo masculino (58,3%) e cinco participantes eram do sexo feminino (41,7%) (Figura 7).

Figura 7: Gênero dos participantes.



A metade da amostra tinha faixa etária entre 18 a 21 anos e a outra metade entre 22 a 30 anos (Figura 8).

Todos os participantes declararam que conheciam alguma linguagem de programação, ou seja, tinham alguma experiência com o conteúdo (Figura 9). As linguagens mais conhecidas pelos alunos eram: Python (100%), Java (58,3%), C (41,7%), JavaScript (33,3%), C++ (25%). Também foram citadas C#, PHP, Haskell, Go, Assembly e Spyder. O Python, além de ser a linguagem mais mencionada, é frequentemente utilizada por biólogos. Observa-se que ainda é raro, especialmente no

Brasil, encontrar disciplinas nos cursos de graduação em biologia que contemplem de forma abrangente o ensino de alguma linguagem de programação (Salazar, 2018).

Figura 8: Faixa etária dos participantes.

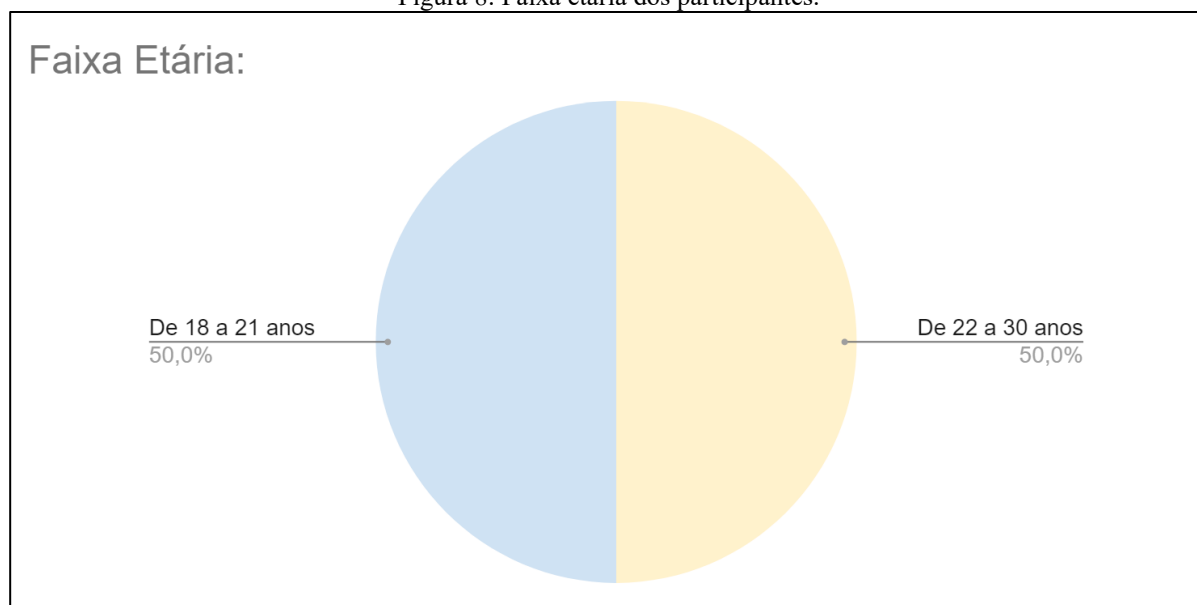
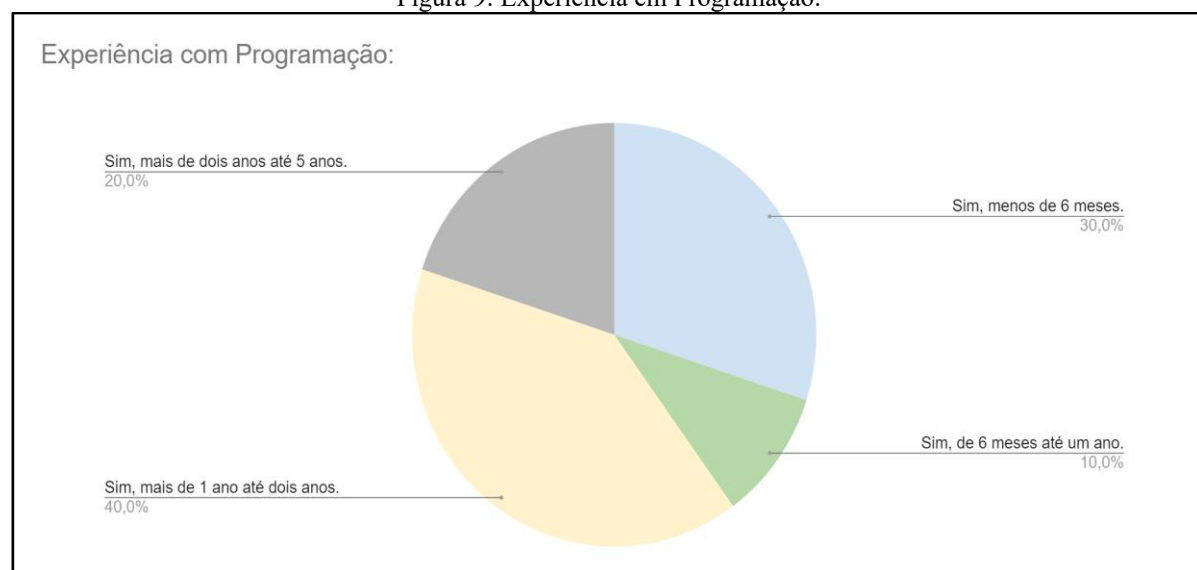


Figura 9: Experiência em Programação.



Após a execução das tarefas os alunos responderam questões quantitativas sobre sua experiência, que consideraram o Scratch adequado para o ensino de programação para iniciantes, 90% dos alunos responderam que concordavam, sendo que um aluno de engenharia de software acredita que poderia ser utilizado na educação básica. Alunos da área de ciências biológicas deram respostas positivas quanto a linguagem ser adequada aos alunos de graduação e que utilizam o Scratch para dar continuidade a prática. Porém, dois alunos da área tecnológica afirmaram que não continuariam

utilizando essa linguagem, bem como não souberam opinar se o Scratch poderia ser utilizado por alunos de outras áreas, enquanto outro aluno afirmou que a linguagem não deveria ser empregada.

Com relação a análise qualitativa, os dados que emergiram foram agrupados e posteriormente discutidos segundo as categorias do método ARCS. As 4 categorias puderam ser evidenciadas, conforme descrito na Tabela 3.

Tabela 3. Categorias evidenciadas durante o PAE

Método ARCS	Conceito Norteador
Atenção	Essa categoria pode ser verificada porque percebeu-se o foco dos educandos durante a execução das atividades práticas, na qual os alunos conseguiram realizar as tarefas. Eles também concluíram o projeto no tempo esperado.
Relevância	Com base nos dados quantitativos, os resultados sugerem que a linguagem de programação Scratch é relevante para os alunos que não são de áreas computacionais.
Confiança	Notou-se a motivação na medida que os alunos percebiam seus avanços na execução dos comandos de programação e sentiam-se à vontade ao explorar os recursos. Assim, a categoria emergiu pela facilidade dos alunos ao demonstrarem confiança, capazes de realizar tarefas e construir seu próprio projeto.
Satisfação	Durante as atividades práticas percebeu-se que os alunos expressavam seus sentimentos, demonstrando alegria e surpresa ao interagir com o Scratch. Também demonstraram criatividade e imaginação para propor o desenvolvimento dos projetos, nota-se o engajamento e a satisfação dos participantes.

A partir da análise das informações contidas na tabela, podemos estabelecer relações com a literatura científica para detectar e interpretar pontos de divergência ou convergência. A análise dos dados qualitativos sugere que a linguagem de programação Scratch pode ser utilizada no ensino superior, principalmente para alunos iniciantes.

No questionário pré-teste, os alunos relataram que a aprendizagem por meio do pensamento computacional e programação proporciona uma melhor compreensão das atuais tecnologias (programas e seu funcionamento) e que seu entendimento permite a implementação facilitada de programas. Os estudantes ainda demonstraram, que a programação é uma base importante para diversos quesitos, tanto para a atualidade quanto para seu futuro, o que demonstra a adequação à relevância e satisfação.

Do ponto de vista à aprendizagem, os alunos acreditam que o pensamento computacional é recurso importante para o desenvolvimento cognitivo, além de possibilitar a inserção no mercado de trabalho e o desenvolvimento de habilidades que podem auxiliá-los no trabalho acadêmico, o que corrobora com Wangenheim, Nunes e Santos (2014) quando afirmam que este é um conhecimento necessário para sua inserção no mercado de trabalho, independente da sua área final de estudo ou carreira profissional escolhida. Neste sentido, Valente (2002) afirma que essas habilidades e competências devem ser construídas por cada aprendiz na interação com objetos e com pessoas que coabitam o seu cotidiano.

Neste contexto a linguagem de programação Scratch torna-se um instrumento com potencial para engajar os alunos na aprendizagem do pensamento computacional, que tem uma forte ressonância

com as atividades nas quais os usuários estão interessados, como a criação de histórias animadas, jogos e apresentações interativas (Valente, 2016). Este engajamento pode ser observado pelos depoimentos dos participantes:

- “o Scratch utiliza os mesmos componentes de linguagem que outras formas de programação, como o Python. Isso facilita que uma pessoa sem experiência do Scratch para outro meio de linguagem, com uma base adquirida de forma descontraída” – Relevância;
- “Foi um jeito mais simples de aprender os comandos e que envolve quem está aprendendo” – Relevância;
- “Acredito que a utilização do Scratch no início do aprendizado seja mais intuitivo e prazeroso para o aluno” – Satisfação;
- “Atualmente, as linguagens de programação codificadas podem ser muito complexas para iniciantes, assustando nas primeiras impressões. O Scratch permite uma aprendizagem mais amigável e interativa” – Relevância e confiança;
- “A ferramenta Scratch proporciona a possibilidade de visualização do que está acontecendo, de maneira que o usuário consegue acompanhar e compreender melhor o que cada comando faz” – Atenção;
- “Com o Scratch, tive uma experiência menos estressante, comparado com outra linguagem de programação propriamente dita, obtendo conhecimentos em estruturas de maneira fácil” – Relevância;
- “A utilização da Scratch auxilia muito no entendimento dos conceitos iniciais de programação, sem contar que acaba sendo um método divertido e dinâmico de aprender a programar. Ao contrário do método tradicional, em que o primeiro contato do aluno de ciências biológicas com a linguagem de programação é um pouco intimidador e acaba se tornando, na maioria dos casos, uma cadeira maçante e difícil” – Relevância, confiança e satisfação;
- “O Scratch abstrai toda a parte de codificação que na aprendizagem tradicional pode assustar e fazer com que a pessoa desista de aprender. Além disso, na ferramenta é possível identificar rapidamente o efeito de cada passo programado e identificar o que deve ser feito para chegar no objetivo final e isso não fica tão claro quando começamos a aprender com uma linguagem codificada” – Atenção;
- “Acredito ser uma forma mais didática na introdução de conceitos que muitas vezes são difíceis de assimilar para quem não tem vivência com a área” – Relevância;
- “A vantagem é que o aluno se envolve mais” – Relevância;
- “O Scratch possui uma interface amigável e simplifica conceitos que inicialmente são difíceis de entender. A abstração da codificação através de blocos coloridos e a utilização de

personagens e cenários auxilia para que a aprendizagem se torne divertida” – Relevância e satisfação;

- *“A didática, o programa apresenta de uma forma simples e divertida os conceitos básicos da programação; pequenos comandos que, se explicados juntamente com uma base teórica, onde tenha algum profissional apto a ensinar adequadamente, pode se tornar proveitoso ao primeiro momento de imersão nesse ramo de estudo*” – Relevância e confiança;
- *“A possibilidade de conseguir visualizar as ações para as quais os comandos estão associados possibilitando que pessoas sem nenhuma experiência possam iniciar a programar*” – Relevância.

Pode-se observar pelos recortes participantes estão de acordo com as definições de Valente (2016) que afirma que o pensamento computacional é um processo de resolução de problemas e que apresenta as seguintes características: formulação de problemas de uma forma que permita usar um computador e outras ferramentas para ajudar a resolvê-los; organização lógica e análise de dados; automação de soluções por meio do pensamento algorítmico (a série de passos ordenados); identificação, análise e implementação de soluções possíveis com o objetivo de alcançar a mais eficiente e efetiva combinação de etapas e recursos; e generalização e transferência desse processo de resolução de problemas para uma ampla variedade de problemas.

Isso é importante porque os objetos e atividades devem ser estimulantes para que o aluno possa estar envolvido, permitir explorá-las e incrementar a qualidade da interação (Valente, 2002).

Em relação à aprendizagem os alunos demonstram que o Scratch permite a compreensão de conceitos básicos de programação de forma interativa e simplifica conceitos que inicialmente são difíceis de entender, facilita o entendimento de algoritmos e estruturas de código, como variáveis, e auxilia na análise dos resultados. Sem nenhum rigor formal, característico da programação tradicional, os conceitos são aprendidos mais facilmente e o feedback direto sobre as ações, via representação gráfica, tornam a interação mais descontraída e divertida.

Questionados sobre as desvantagens, somente um aluno apontou pontos negativos no seu uso no ensino superior: *“Perfeito para ensinar estruturas lógicas para crianças sobre programação, para adultos penso que não seja eficiente e produtivo, mesmo assim, é sempre bom aprender com prática em outras áreas, pode ser mais vantajoso, porque ele não mostra os códigos de cada comando/bloco”*.

Os alunos sugeriram que seria interessante se a interação permitisse acompanhar a linguagem de programação que está vinculada a cada bloco de ação e que poderia haver níveis mais avançados de programação na linguagem. Por exemplo, uma aluna da biologia afirmou que *“a inserção de blocos dentro de blocos permite que se execute tudo de uma vez, porém para executar apenas uma parte (para ver o que faz) é preciso retirar-lo do maior bloco, o que dificulta em entender o que cada parte está*

fazendo. Além disso, construir algo que precise de muitos blocos dentro de outros blocos deixa a construção trabalhosa demais e os conceitos e entendimento como um todo (que deveria ser o principal) fica de lado. Seria interessante que a ferramenta apresentasse estruturas como vetores e matrizes (que são estruturas básicas e importantes)”.

Por último, ao final da aplicação do PAE os alunos relatam o seguinte:

- *“A programação é importante independente da carreira, pois ajuda a ter uma noção de como funcionam ferramentas que utilizamos todos os dias. Na Biologia, diversas áreas utilizam a programação, como na criação de scripts e afins, como em áreas da Bioinformática. A programação não nos permite apenas trabalhar em áreas aplicadas, mas exercitar a lógica e o pensamento”;*
- *“É importante para obter mais conhecimento e base da programação”;*
- *“O aprendizado da programação para pessoas de qualquer idade é importante para o desenvolvimento do raciocínio lógico e formulação de estratégias para resolver problemas”;*
- *Continuo com a mesma opinião de que aprender programação é essencial atualmente. Mas entendi um pouco melhor do porquê isto é importante”;*
- *“Depois de participar da oficina acredito que o aprendizado de programação, além de proporcionar maiores possibilidades de atuação no mercado de trabalho, proporciona a estimulação para a resolução de problemas cotidianos”;*
- *“A programação é pilar fundamental para minha carreira, pois é necessário entender o processo envolvendo a codificação puramente dita, além de desenvolver o raciocínio lógico que os softwares demandam, para buscar implementar soluções mais elegantes”;*
- *“O aprendizado da linguagem de programação por biólogos é de suma importância, pois está muito presente atualmente devido ao crescente desenvolvimento tecnológico e, além disso, desempenha um papel significativo na biotecnologia, análise de dados etc.”;*
- *“Continuo a pensar que aprender a programar é extremamente importante para solucionar problemas, mas não somente isso. Aprender a programar ajuda no processo de pensar, esquematizar e buscar diversas soluções para determinado problema ou simplesmente para evitar o retrabalho”;*
- *“Na área que sigo é de muita importância, mas em um sentido mais amplo, a área acadêmica envolve muita técnica computacional que pode ser facilitada com uma aprendizagem em uma linguagem computacional”.*

5 CONSIDERAÇÕES FINAIS

Com o advento das digitais em meio ao âmbito educacional, tornou-se imprescindível uma reflexão sobre os aspectos relacionados às interações dos usuários com ferramentas voltadas à

educação e seus materiais pedagógicos (Herpich & Tarouco, 2016). Neste sentido, a fim de contribuir com um maior engajamento de alunos de graduação para aprender lógica computacional, este artigo relatou as experiências práticas da aplicação de um Projeto Ação na Escola que envolveu o uso da linguagem de programação Scratch, por meio de jogos e animações, por alunos do ensino superior.

O relato dos participantes sugere que alunos, em sua maioria, consideram a linguagem Scratch adequada para o ensino de programação, principalmente a iniciantes e alunos de áreas não tecnológicas.

Este trabalho discutiu as diferenças de percepção entre os alunos de áreas distintas - Ciências Biológicas e Ciências Exatas - e o quanto os alunos de Ciências Biológicas se sentiram engajados para seguir aprendendo programação em suas carreiras. Neste contexto, emergiram algumas situações que foram passíveis de discussão. A composição da amostra, com alunos de áreas biológicas e da computação, onde percebeu que o uso de jogos digitais e animações, desenvolvidos no Scratch, voltados à aprendizagem de conteúdos iniciais de lógica de programação é um fator motivacional para alunos que estão iniciando sua prática nessa área, bem como pertencentes a áreas como as ciências biológicas.

Portanto, pode-se dizer que o pensamento computacional apresentou indícios como prática pedagógica engajadora, possível de ser utilizada no ensino superior, principalmente para alunos das áreas biológicas. Nesse mesmo sentido, Santana (2023), ao utilizar o Scratch no ensino de biofísica com licenciandos em Biologia, observou que a linguagem em blocos favoreceu a aprendizagem de conceitos complexos relacionados às membranas biológicas, promovendo o desenvolvimento do pensamento algorítmico e contribuindo para uma compreensão interdisciplinar envolvendo biologia, física e química. Isso reforça o potencial do pensamento computacional como ferramenta significativa para a aprendizagem ativa em contextos diversos da educação em Ciências.

Apesar dos resultados promissores, esta pesquisa apresenta limitações. A principal delas refere-se ao número reduzido de participantes, o que restringe a generalização dos dados. Além disso, o tempo destinado à aplicação do Projeto Ação na Escola foi curto, o que pode ter limitado o aprofundamento dos conteúdos de programação e o uso mais autônomo da ferramenta Scratch pelos estudantes. Como proposta para trabalhos futuros, recomenda-se a realização de investigações com amostras maiores, aplicação em diferentes contextos formativos e o desenvolvimento de trilhas formativas contínuas que integrem o pensamento computacional ao currículo de forma interdisciplinar. Também seria relevante explorar outras linguagens visuais e ambientes de programação com foco educacional, a fim de ampliar as possibilidades de ensino-aprendizagem no campo das Ciências Biológicas e em cursos de licenciatura de modo geral.

REFERÊNCIAS

- Alves, N. da C., Wangenheim, C. G. von, Rodrigues, P. E., Hauck, J. C. R., & Borgatto, A. F. (2016). Ensino de Computação de Forma Multidisciplinar em Disciplinas de História no Ensino Fundamental – Um Estudo de Caso. *Revista Brasileira de Informática na Educação*, 24(3), 31.
- Bardin, L. (1977). *Content analysis*. São Paulo: Livraria Martins Fontes.
- Bishop, Ö. T., Adebiyi, E. F., Alzohairy, A. M., Everett, D., Ghedira, K., Ghouila, A., ... H3Africa Consortium. (2015). Bioinformatics Education—Perspectives and Challenges out of Africa. *Briefings in Bioinformatics*, 16(2), 355–364.
- Castilho, M. I., & Ricci, T. dos S. F. (2006). O uso de animações como elemento motivador de Aprendizagem. *Encontro Estadual de Ensino de Física*. (1.: 2005 Nov. 24-26: Porto Alegre, RS). Atas. Porto Alegre: Instituto de Física-UFRGS, 2006.
- Cybis, W. de A., Betiol, A. H., & Faust, R. (2015). *Ergonomia e usabilidade: Conhecimentos, métodos e aplicações* (3rd ed.). São Paulo, Brasil: Novatec.
- Czerkawski, B. C., & Lyman, E. W. (2015). Exploring Issues About Computational Thinking in Higher Education. *TechTrends*, 59(2), 57–65. <https://doi.org/10.1007/s11528-015-0840-3>
- Denning, P. J. (2018). Computational thinking in science. In *The Best Writing on Mathematics 2018* (pp. 67–77). United States: Mircea Pitici. Princeton University Press.
- Gomes, T. C. S., Tedesco, P. C. de A. R., & Melo, J. C. B. de. (2016). Jogos no Design de Experiências de Aprendizagem de Programação Engajadoras. *Jornada de Atualização em Informática na Educação*, 5(1), 39–77.
- Herpich, F., & Tarouco, L. M. R. (2016). Engajamento de usuários em mundos virtuais: uma análise teórica-prática. *Renote*, 14(1). <https://doi.org/10.22456/1679-1916.67387>
- Keller, J. M. (2009). *Motivational design for learning and performance: The ARCS model approach*. Springer Science & Business Media.
- Magana, A. J., Taleyarkhan, M., Alvarado, D. R., Kane, M., Springer, J., & Clase, K. (2014). A Survey of Scholarly Literature Describing the Field of Bioinformatics Education and Bioinformatics Educational Research. *CBE Life Sciences Education*, 13(4), 607–623.
- Marques, D. L., Costa, L. F. S., Silva, M. A. de A., & Rebouças, A. D. D. S. (2011). Atraindo Alunos do Ensino Médio para a Computação: Uma Experiência Prática de Introdução a Programação utilizando Jogos e Python. *Anais do Workshop de Informática na Escola*, 1(1), 1138–1147. <https://doi.org/10.5753/cbie.wie.2011.1138-1147>
- Preece, J., Rogers, Y., & Sharp, H. (2013). *Design de Interação. Além da Interação Homem-Computador* (Edição: 3a). Porto Alegre: Bookman.
- Qin, H. (2009). Teaching Computational Thinking Through Bioinformatics to Biology Students. *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, 188–191. <https://doi.org/10.1145/1508865.1508932>

Pozzebon, Lidiane Vieira; Velho, André Ricardo Theodoro, & Barwaldt, Regina. (2019). Viagem espacial: Um recurso para a alfabetização. In *Formação de Professores em Tecnologias Educacionais: Construindo práticas pedagógicas*. Coleção Cadernos Pedagógicos da EaD. Regina Barwaldt, Joice Rejane Pardo Maurell e Virgínia da Silva Xavier (Orgs.) (Vol. 31, p. 396). Editora da FURG.

Ramos, F., & Teixeira, L. da S. (2015). Significação da Aprendizagem Através do Pensamento Computacional no Ensino Médio: Uma Experiência com Scratch. *Anais do Workshop de Informática na Escola*, 21(1), 217. <https://doi.org/10.5753/cbie.wie.2015.217>

Ranganathan, S. (2005). Bioinformatics Education-Perspectives and Challenges. *PLOS Comput Biol*, 1(6), e52.

Santana, M. P. S. (2023). *Pensamento computacional no ensino de biofísica na formação inicial de professores de Biologia: utilizando programação em blocos com o Scratch* (Dissertação de mestrado). Faculdade de Ciências, Universidade Estadual Paulista "Júlio de Mesquita Filho", Bauru. <https://hdl.handle.net/11449/253143>

Salazar, V. W. (2018). *Ensino de programação para biólogos: Retornando os conhecimentos de um estudo empírico em oceanografia física* (Trabalho de Conclusão de curso, Universidade Federal de Santa Catarina).

Schneider, E. M., Fujii, R. A. X., & Corazza, M. J. (2017). Pesquisas quali-quantitativas: Contribuições para a pesquisa em ensino de ciências. *Revista Pesquisa Qualitativa*, 5(9), 569–584.

Silva, A. H., & Fossá, M. I. T. (2015). Análise de conteúdo: exemplo de aplicação da técnica para análise de dados qualitativos. *Qualitas Revista Eletrônica*, 16(1).

Soares, J., & Rodrigues, A. (2016). Avaliação do Jogo Educativo Lei de Mendel. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 27(1), 826.

Sousa, R. M., & Lencastre, J. A. (2014). Scratch: Uma opção válida para desenvolver o pensamento computacional e a competência de resolução de problemas. In *Atas do 2.o Encontro sobre Jogos e Mobile Learning*. (pp. 256–267).

Tarouco, L. M. R., Roland, L. C., Fabre, M.-C. J. M., & Konrath, M. L. P. (2004). Jogos educacionais. *RENTE: Revista Novas Tecnologias Na Educação [Recurso Eletrônico]*. Porto Alegre, RS.

Valente, J. A. (2002). A espiral da aprendizagem e as tecnologias da informação e comunicação: Repensando conceitos. *A Tecnologia No Ensino: Implicações Para a Aprendizagem*. São Paulo: Casa Do Psicólogo, 15–37.

Valente, J. A. (2016). Integração do pensamento computacional no currículo da educação básica: Diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. *Revista E-Curriculum*, 14(3), 864–897.

Wangenheim, C. G. von, Nunes, V. R., & Santos, G. D. dos. (2014). Ensino de Computação com SCRATCH no Ensino Fundamental – Um Estudo de Caso. *Revista Brasileira de Informática na Educação*, 22(03), 115.

Welch, L. R., Schwartz, R., & Lewitter, F. (2011). Report from the ISCB Education Committee: A draft curriculum for bioinformatics [abstract]. Third Annual RECOMB Conference on Bioinformatics Education (RECOMB-BE), an ISMB/ECCB 2011 Satellite Meeting. Presented at the 19th Annual International Conference on Intelligent Systems for Molecular Biology and 10th European Conference on Computational Biology, Vienna, Austria.

Wing, J. M. (2006). Computational Thinking. *Commun. ACM*, 49(3), 33–35.

Zanetti, H., Borges, M., & Ricarte, I. (2016). Pensamento Computacional no Ensino de Programação: Uma Revisão Sistemática da Literatura Brasileira. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 27(1), 21. <https://doi.org/10.5753/cbie.sbie.2016.21>