

Programação para todos: Análise Comparativa de Ferramentas Utilizadas no Ensino de Programação

Silvino M. da Silva Junior¹, Sônia V. Alves França²

¹Campus Floriano – Instituto Federal do Piauí (IFPI)
CEP – 64.808-753 – Floriano – PI – Brazil

²Programa de Pós-Graduação em Tecnologia e Gestão em Educação a Distância
Universidade Federal Rural de Pernambuco (UFRPE) – Recife, PE – Brazil

silvinomarques@ifpi.edu.br, soniafranca@gmail.com

***Abstract.** The expansion of computer programming teaching to different audiences and areas is an eminent reality, and, at the same time, subjects related to Computer programming courses presents more learning difficulties by the students. This article aims to analyze and make a comparison with some tools that were proposed in previous papers and that are used in the Teaching of this subject to observe if these tools supply this new reality of the initial teaching of programming for children and young people. It was observed that although some of the projects found in the computer programming literature are extremely successful, some deficiencies can still be noticed.*

***Resumo.** A expansão do ensino de programação de computadores para diferentes públicos e áreas é uma realidade eminente, ao mesmo tempo que os assuntos relacionados a essa disciplina são considerados campeões em apresentar dificuldades de aprendizagem por parte dos estudantes. Este artigo tem por objetivo analisar e fazer um comparativo com algumas ferramentas que foram propostas em trabalhos anteriores e que são utilizadas no ensino dessa disciplina, buscando observar se essas ferramentas atendem a essa nova realidade do ensino inicial de programação para crianças e jovens. Observou-se que embora alguns dos projetos encontrados na literatura sejam extremamente exitosos, ainda se pode notar algumas deficiências.*

1. Introdução

O ensino de programação tem se tornado cada vez mais importante no trajeto escolar das pessoas, não se limitando a acadêmicos dos cursos de computação, mas atingindo a todas as áreas e faixas etárias, em particular as crianças e jovens. Proporcionando um melhor entendimento e utilização das tecnologias computacionais atuais, e permitindo o desenvolvimento do raciocínio lógico e a capacidade de resolução de problemas.

Atualmente, em alguns países desenvolvidos, como é o caso do Reino Unido, as disciplinas de programação já pertencem ao currículo escolar, abordando temas como pensamento computacional, lógica e algoritmos. (PAULA; VALENTE; BURN, 2014).

Existem também várias iniciativas mundiais voltadas para disseminação do aprendizado de programação, como o *Code.org*¹ e o *Khan Academy*².

No Brasil, apesar de ainda em fase inicial e apenas em alguns estados, já existem escolas adotando esse modelo. Recentemente, o prefeito da cidade de São Paulo anunciou que pretende implementar aulas de programação nas escolas administradas pelo município através de uma parceria entre a prefeitura e o Ministério da Educação para equipar as unidades escolares e promover o ensino de programação no currículo. (DORIA, 2017).

Entretanto, em vários trabalhos sobre aprendizagem de programação, foram encontrados exemplos da dificuldade de conhecer e se habituar aos ambientes e às linguagens, tarefa considerada difícil e que necessita de muito tempo e esforço por parte dos estudantes e, dos docentes.

As disciplinas de Lógica de Programação e Algoritmos e Programação podem ser ensinadas de diversas formas, cuja metodologia tradicional ainda prevalece, na qual o professor transmite conhecimentos e soluções de diversos problemas e o aluno recebe essas informações sem discutir e/ou criar as próprias soluções.

O uso de recursos como animações e vídeos torna-se facilitador do processo de aprendizagem, visto que a apresentação de conceitos abstratos é mais didática, melhorando a qualidade do material de apoio à aula (TAVARES et al., 2016).

Neste contexto, surgiram os seguintes questionamentos, que instigaram o desenvolvimento deste artigo: As ferramentas disponíveis e utilizadas atualmente estão aptas a atender essa nova demanda e público? Essas ferramentas possuem características necessárias para a aprendizagem inicial de programação?

O presente trabalho tem caráter qualitativo e realizou-se através de uma pesquisa bibliográfica sobre os softwares e ferramentas tecnológicas existentes que auxiliam os professores no ensino de algoritmos e programação. Dentre os softwares estudados, destacaram-se: *Feeper*, *WH-IDE*, *TutorICC*, *Portugol Studio*, *AAPW*, *TSTView*, *The Huxley*, *SOAP* e *Scratch*.

2. Ferramentas Utilizadas no Ensino de Programação

Esta seção aborda algumas ferramentas tecnológicas existentes que são utilizadas no apoio ao ensino das disciplinas de programação, abordando suas principais características, pontos positivos e negativos.

As ferramentas foram selecionadas com base em Revisões Sistemáticas da Literatura realizadas por Silva et al. (2015) e Ramos et al. (2015), que analisaram iniciativas brasileiras na criação de ambientes de aprendizagem de programação, nos principais eventos brasileiros da área, que englobam o Congresso Brasileiro de Informática na Educação (CBIE) e a Revista Brasileira de Informática na Educação (RBIE). Foi excluída deste trabalho a avaliação de softwares criados antes de 2010.

¹ <https://code.org/>

² <http://www.khanacademy.org/>

2.1 Ambiente Virtual Feedback Personalizável (Feeper)

O Feeper é um ambiente virtual para apoiar o ensino/aprendizagem de programação no ensino superior. Utiliza um mecanismo para melhorar a comunicação entre aluno e professor, permitindo sinalizar partes do código fonte (ALVES; JAQUES, 2014).

A ferramenta funciona com o envio do código fonte produzido pelo estudante para validação de um Juiz On-line. Este então solicita ao banco de dados os testes existentes para o exercício que está sendo respondido. O Juiz On-line compara a resposta gerada pelo programa enviado com as respostas esperadas. Caso haja divergência, uma mensagem personalizada de alerta é emitida, contendo orientações para que este consiga resolver o problema gerado, do contrário, então o programa é aceito e a resposta de sucesso é emitida ao estudante.

O feedback fornecido pelo Juiz On-line foi personalizado para apresentar mensagens que indiquem onde aconteceu o erro, por que ele aconteceu e como corrigi-lo e para permitir a execução de diferentes tipos de teste no programa implementado (ALVES; JAQUES, 2014).

Através do *Feeper* o aluno pratica conceitos de programação orientada a objetos, utilizando a linguagem Java. Além da correção automática, a ferramenta emite dicas sobre os erros gerados pelo programa do aluno permitindo que o mesmo possa adicionar marcações em seu programa que serão exibidas ao professor. (ALVES; JAQUES, 2014).

2.2 Ambiente Personalizável para Aprendizado de Programação – WH-IDE

É um ambiente para ensino de programação de computadores, no qual os estudantes programam os algoritmos em português. O professor define todas as instruções que estarão à disposição dos aprendizes, personalizando, assim, a linguagem a ser utilizada e ignorando o máximo de detalhes técnicos. (FRANTZ; PONTES, 2014).

A ferramenta possui algumas limitações: ela não captura possíveis erros de sintaxe existentes no código, pelo fato de usar o compilador da linguagem Java para a geração do programa executável, além de não possuir depurador visual e área para exibição das variáveis criadas em memória (FRANTZ; PONTES, 2014).

2.3 Ambiente Interativo e Adaptável para ensino de Programação - TutorICC

O TutorICC é um ambiente interativo e adaptável utilizado para o ensino de programação em Pascal. O conteúdo está dividido em níveis de dificuldade, permitindo ao estudante estabelecer o caminho mais conveniente para ele, dentro do conteúdo da disciplina (PÍCCOLO et al., 2010).

Há grande interatividade com o aprendiz, a cada passo é apresentado a ele um problema, para que construa um programa. Esses programas são corrigidos imediatamente pelo TutorICC, que recomenda ao estudante o melhor caminho a seguir dentro do conteúdo da disciplina.

A ferramenta foi desenvolvida utilizando a linguagem Java. A principal contribuição desta em relação aos demais trabalhos encontrados na literatura, é que permite que o professor personalize a apresentação visual em cada exemplo. Pode

escolher posição, tamanho e cor de cada uma das variáveis, dispendo-as da forma que achar mais didática para captar a atenção dos alunos (PÍCCOLO et al., 2010).

2.4 Portugol Studio

O Portugol Studio é um Ambiente Integrado de Desenvolvimento (IDE) para auxiliar a aprendizagem de algoritmos para iniciantes. Foi desenvolvida no idioma português, facilitando o acesso ao público com baixa fluência no idioma inglês (NOSCHANG et al., 2014).

A IDE possibilita o desenvolvimento, a execução e a depuração de programas escritos em Portugol, sendo eficaz para iniciantes em programação, principalmente por utilizar o português como língua nativa (ANIDO, 2015).

O fato de ser um aplicativo para desktop dificulta um pouco para que o estudante possa praticar em casa, pois é necessário instalar as ferramentas do ambiente (editor, compilador, IDE, etc.), o que pode ser uma tarefa complexa (ANIDO, 2015).

2.5 Ambiente de Aprendizagem de Programação Web - AAPW

O AAPW permite o aprendizado de conceitos envolvidos na programação *Web*, de forma simples e objetiva, podendo ser utilizada como recurso auxiliar em disciplinas de cursos que abordam desenvolvimento *Web*. (GOMES; D'EMERY; FILHO, 2014).

A ferramenta possui uma IDE de programação desenvolvida em Java, que é composta de barra de ferramentas; das telas de edição do programa fonte e do código compilado; de um browser para visualizar as páginas desenvolvidas; e de um console que exibe mensagens de erros presentes no código pertencente ao programa fonte.

A ferramenta trabalha com formulários, não necessita de *tags* no código e utiliza instruções simples com comandos em português. A IDE propicia ao usuário a visualização tanto do código construído em português quanto da codificação gerada, a qual pode ser interpretada em um browser. (GOMES; D'EMERY; FILHO, 2014).

2.6 Ferramenta de Acompanhamento de Exercícios de Programação - TSTView

O TSTView permite acompanhar o desempenho dos estudantes em um certo período de tempo, além de monitorar em tempo real o desenvolvimento dos estudantes frente aos exercícios planejados possibilitando detectar, com mais facilidade e eficiência, dificuldades individuais dos estudantes, dificuldades comuns a grupos de estudantes e dificuldades gerais da turma (GAUDENCIO, 2013).

Os estudantes respondem aos exercícios propostos utilizando a linguagem de programação Python e submetem a um testador automático que aplica os testes criados pelos professores para as questões propostas. Os dados de controle e os resultados dos testes aplicados as submissões compõem a sua base de dados.

A ferramenta possui dois módulos principais, o do professor e o do aluno. O primeiro exibe uma lista dos aprendizes que estão realizando alguma atividade no momento da consulta. Já o segundo, exibe um relatório histórico individual para cada um deles (GAUDENCIO, 2013).

Ao submeter uma solução, o estudante recebe feedback na forma de resultados de testes sobre seu código, podendo visualizar que tipos de erros ele cometeu e o número de casos de teste em que falhou.

2.7 Ferramenta para a Avaliação de Aprendizado de Alunos em Programação de Computadores - The Huxley

The Huxley é uma ferramenta *web* que permite aos estudantes responderem a exercícios de uma base de dados com centenas de problemas e submeterem às respostas através de código em diversas linguagens de programação.

O ambiente corrige os exercícios automaticamente através de análise sintática do código e testes de aceitação e emite feedback. O professor tem acesso à quantidade de problemas resolvidos, porcentagem de acertos/erros, tipos de problemas com mais erros, detecção de plágio e erros específicos de cada aluno (PAES, 2013).

Através do ambiente, o docente faz uma avaliação personalizada do aluno, baseada na identificação de erros específicos durante as avaliações. A definição e correção de exercícios e provas, verificação de plágio, definição de notas e publicação dos resultados pode ser feita automaticamente através do The Huxley (PAES, 2013).

2.8 Sistema On-line de Atividades de Programação - SOAP

É um sistema *web* que possibilita ao professor disponibilizar tarefas para as turmas e, ao aluno, realizar submissões de exercícios. Os exercícios submetidos, que são programas de computador em Linguagem C, são compilados e executados em um servidor remoto, gerando relatórios que são disponibilizados aos professores.

As atividades são processadas pelos núcleos (Núcleo de Avaliação Diagnóstica NAD) e Núcleo de Avaliação Formativa (NAF). O NAD processa essas atividades, gerando planilhas e mapas de perfis de aprendizagem, o NAF, com base no diagnóstico fornecido pelo NAD, recomenda atividades diretamente para que os estudantes com dificuldades possam melhorar os desempenhos (OLIVEIRA; OLIVEIRA, 2014).

O software combina técnicas de reconhecimento de padrões para classificar alunos e recomendar-lhes atividades conforme os seus perfis e possibilita ao docente realizar melhor gestão da aprendizagem individual de seus estudantes (OLIVEIRA; OLIVEIRA, 2014).

2.9 Scratch

O ambiente de programação *Scratch* traz linguagem que contribui para a aprendizagem de programação através da criação de histórias, animações, simulações e outros produtos multimídia, arrastando e agrupando diferentes conjuntos de blocos de comandos coloridos (MARQUES, 2013).

Pelo fato de os comandos *Scratch* se basearem em uma estrutura de blocos de montar, o aprendiz pode concentrar esforço na busca pela construção do algoritmo, não se preocupando com erros de sintaxe. Desta maneira, pode-se afirmar que o caráter mais didático do *Scratch* projeta no aluno a possibilidade de ele se concentrar no exercício do pensamento algorítmico e na criatividade durante a construção das soluções (SCAICO et al., 2013).

3. Comparativo das Ferramentas

Os *softwares* apresentados compartilham objetivos, como desenvolver a criatividade, o raciocínio, a descoberta e interação social. Todos exploram as interfaces visuais como meio de motivar a aprendizagem dos conceitos de programação. O que os difere, entretanto, é que uns são apenas IDE, onde o aluno pode escrever e testar códigos, alguns são voltados para o ensino de lógica de programação e outros para conceitos mais avançados de programação, como orientação a objetos.

Para uma efetiva comparação das ferramentas analisadas, foram utilizados alguns critérios como a forma de interação do usuário com o ambiente, a plataforma de execução da ferramenta, as linguagens de programação suportadas e se está disponível no idioma Português.

No Quadro 1, pode-se observar a relação das ferramentas e os critérios que foram utilizados na análise.

Quadro 1 – Características Técnicas das Ferramentas. Fonte: Elaborado pelo autor, 2017.

Ferramenta	Forma de Interação	Plataforma	Linguagem de Programação	Disponível em Português
<i>Feeper</i>	Textual	<i>Web</i>	Java	Sim
<i>WH-IDE</i>	Textual	Java	Portugol	Sim
<i>TutorICC</i>	Visual e Textual	Java	Pascal	Sim
<i>Portugol Studio</i>	Visual e Textual	Java	Portugol	Sim
<i>AAPW</i>	Visual e Textual	Java	Portugol	Sim
<i>TSTView</i>	Visual e Textual	<i>Web</i>	<i>Phyton</i>	Sim
<i>The Huxley</i>	Textual	<i>Web</i>	Várias	Sim
<i>SOAP</i>	Visual e Textual	<i>Web</i>	C	Sim
<i>Scratch</i>	Visual e Textual	<i>Web</i>	Blocos	Sim

Com exceção do *Scratch*, que utiliza blocos, todos os demais utilizam uma ou várias linguagens de programação. A maioria deles é composta de recursos visuais e textuais e podem ser executados na *web* através de um *browser*. Embora, todos os *softwares* apresentados sejam indicados para iniciantes, apenas três deles utilizam a linguagem *Portugol*, que é um fator considerado importante em disciplinas introdutórias.

Na literatura sobre aprendizagem de programação, encontram-se relatos de que o processo de conhecer e se habituar aos ambientes e às linguagens é uma tarefa árdua, com grande dispendiosidade de tempo por parte dos estudantes e, por vezes, dos docentes ou tutores (RIBEIRO, 2015).

No que se refere as funcionalidades implementadas, verificou-se acerca da possibilidade do usuário visualizar os valores das variáveis durante a execução de um algoritmo, o uso de animações e vídeos na explicação dos conceitos e técnicas utilizadas, a forma utilizada para emitir *feedback* e também se possuem ou não características de adaptatividade.

No Quadro 2, tem-se a comparação com relação às funcionalidades disponíveis:

Quadro 2 – Funcionalidades. Fonte: Elaborado pelo autor, 2017.

Ferramenta	Visualização de variáveis	Animações	Vídeos	Feedback	Adaptatividade
<i>Feeper</i>	NÃO	NÃO	NÃO	Manual	NÃO
<i>WH-IDE</i>	NÃO	NÃO	NÃO	Manual	NÃO
<i>TutorICC</i>	SIM	SIM	NÃO	Automático	SIM
<i>Portugol Studio</i>	SIM	NÃO	NÃO	Manual	NÃO
<i>AAPW</i>	NÃO	NÃO	NÃO	Manual	NÃO
<i>TSTView</i>	NÃO	NÃO	NÃO	Automático	NÃO
<i>The Huxley</i>	NÃO	NÃO	NÃO	Automático	NÃO
<i>SOAP</i>	NÃO	NÃO	NÃO	Automático	NÃO
<i>Scratch</i>	NÃO	SIM	SIM	Automático	NÃO

Pode-se perceber que a maioria das ferramentas não utiliza recursos de animações e vídeos para apresentação de conteúdo, explicação dos códigos e funcionamento das variáveis.

A animação de algoritmos é importante, pois, muitas vezes, o professor tem a necessidade de usar representações visuais para ajudar os alunos a entenderem os algoritmos, o que na prática se concretiza, muitas vezes, mostrando o comportamento de programas (TAVARES et al., 2016).

Outra característica apresentada pela maioria dos *softwares* estudados é o *feedback* automático. Segundo Tavares et al. (2016), *feedback* imediato é crucial para a aquisição de conhecimento, independentemente da estratégia de aprendizagem adotada, pois motiva os alunos. Por outro lado, o *feedback* individual pode consumir muito tempo do professor com o risco de que os estudantes possam não se beneficiar dele no devido tempo (QUEIRÓS; LEAL, 2015).

Com relação à adaptatividade, apenas a ferramenta TutorICC apresentou essa funcionalidade, no qual o ritmo de aprendizagem se adapta ao perfil do aluno, podendo ser mais rápido ou mais lento, dependendo de sua maior ou menor facilidade em aprender programação.

A adaptatividade tem como propósito ajustar-se a diferentes níveis do educando, com diferentes conhecimentos prévios, facilitando o aprendizado e reforçando os conceitos apresentados nas aulas expositivas (BRAGA, 2015).

Segundo Gomes et al. (2008), para se atingir a capacidade de resolução de problemas são necessárias três fases: (1) resolução de problemas de diversos domínios, não utilizando algoritmos ou programação; em seguida, (2) uma amostra da utilidade da programação com aplicação dos conhecimentos adquiridos na fase anterior; e, finalmente, (3) passando para a construção dos algoritmos.

Na fase de resolução de problemas, a preocupação maior é com o tipo de problemas que serão apresentados. A utilidade da programação tem por objetivo mostrar

aos estudantes para que serve e como se faz um programa. E na fase de construção os alunos teriam que analisar a resolução de problemas através de programas de computador e também propor e testar novas soluções.

O Quadro 3 analisa se as ferramentas estudadas possibilitam a aplicação dessas fases:

Quadro 3 – Competências trabalhadas (GOMES et al., 2008). Fonte: Elaborado pelo autor, 2017.

Ferramenta	Resolução de problemas de domínios diversos	Utilidade da programação	Construção dos algoritmos
<i>Feeper</i>	NÃO	NÃO	SIM
<i>WH-IDE</i>	NÃO	SIM	SIM
<i>TutorICC</i>	NÃO	NÃO	SIM
<i>Portugol Studio</i>	NÃO	SIM	SIM
<i>AAPW</i>	NÃO	SIM	SIM
<i>TSTView</i>	NÃO	NÃO	SIM
<i>The Huxley</i>	NÃO	NÃO	SIM
<i>SOAP</i>	NÃO	SIM	SIM
<i>Scratch</i>	SIM	SIM	NÃO

Observa-se que a grande maioria das ferramentas estudadas prioriza a etapa de construção dos algoritmos, sem a devida importância às competências anteriores.

Em disciplinas iniciais de programação o foco principal deve ser a capacidade de resolução de problemas, e o uso da linguagem de programação fica para fases posteriores, como meio de concretizar essa resolução através de um algoritmo (BORGES, 2016).

Embora todos os *softwares* apresentados sejam indicados para iniciantes, dentre as ferramentas citadas, a que mais atende a critérios importantes para aprendizagem inicial de programação e discutidos neste trabalho é o *Portugol Studio*.

A ferramenta *Portugol Studio* prioriza o uso do idioma português, e possui diversos elementos presentes nas IDE profissionais, mas preservando sempre o enfoque na aprendizagem dos novatos em programação. Alguns elementos da interface auxiliam o docente a explicar processos como a depuração.

Embora alguns dos projetos encontrados na literatura sejam extremamente exitosos, ainda se pode notar algumas deficiências, como: (1) foco na capacidade de resolução de problemas, (2) uso de animações e vídeos para explicação de conteúdos complexos e (3) utilização do *Portugol*.

4. Conclusão

O ensino de algoritmos e programação é um dos grandes desafios na área de ensino de computação, pois são grandes as dificuldades encontradas pelos alunos.

Considerando a evolução tecnológica e a visível necessidade de expansão desses conteúdos para além dos cursos de computação e uma possível inclusão dessa disciplina no currículo escolar brasileiro, é necessário verificarmos se as ferramentas utilizadas atendem as necessidades de aprendizagem dos estudantes.

Segundo Anido (2015), a primeira e mais importante barreira para estudantes brasileiros usarem as ferramentas existentes e que elas foram desenvolvidas para um público que tem inglês como língua oficial e, apesar dos meios de internacionalização, a maioria dos ambientes exige algum conhecimento da língua inglesa, e não apenas referente às palavras-chave da linguagem de programação.

Se adaptar a realidade desta nova geração, utilizando um ambiente mais lúdico e que considere as especificidades da aprendizagem dos estudantes são características indispensáveis as instituições de ensino. Portanto, é muito importante o desenvolvimento de ferramentas tecnológicas que busquem despertar o interesse do aluno, assim como facilitar o entendimento dos conteúdos e da lógica de programação.

Espera-se que este trabalho incentive a reflexão acerca de metodologias inovadoras e utilização efetiva da tecnologia a favor da educação. Tornar o aluno mais participativo, motivado e autônomo da sua aprendizagem é um desafio diário a ser conquistado.

Referências

- ALVES, F. P.; JAQUES, P. Um ambiente virtual com feedback personalizado para apoio a disciplinas de programação. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. 2014. p. 51.
- ANIDO, R. Saci-ainda outro ambiente para o ensino de programação. In: *Anais do XXIII WEI - Workshop sobre Educação em Computação*. 2015.
- BORGES, M. J. M. *Ferramenta Multimídia de Apoio à Programação Algorítmica e Programação Pascal*. Nº de Folhas 217, Dissertação de Mestrado em Tecnologias de Informação e Comunicação. Universidade de Trás-os-Montes e Alto Douro, 2016.
- BRAGA, O. R. M. *Objeto de aprendizagem adaptativo no ensino da lógica booleana*. 2015. Nº de Folhas 93, Dissertação de Mestrado em Engenharia Elétrica. Universidade Presbiteriana Mackenzie, São Paulo, 2015.
- DORIA, J. [comentário pessoal]. *Facebook*. 10 mar 2017. Disponível em: <<https://www.facebook.com/jdoriavr>>. Acesso em: 11 mar. 2017.
- FRANTZ, W. L. S; PONTES, H. P. Um Ambiente de Desenvolvimento Personalizável para o Ensino de Programação. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. 2014.
- GAUDENCIO, M. et al. Eu Sei o que Vocês Fizeram (Agora e) na Aula Passada: o TSTView no Acompanhamento de Exercícios de Programação. In: *Anais do Simpósio Brasileiro de Informática na Educação*. 2013. p. 204
- GOMES, A. et al. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. In: *Educação, Formação & Tecnologias*; vol.1(1), pp. 93-103, 2008.

- GOMES, M. E. N.; D'EMERY, R. A.; FILHO, G. A. A. C. AAPW: Uma Ferramenta para facilitar o Aprendizado de Programação Web. In: *Anais do XXII WEI - Workshop sobre Educação em Computação*. 2014.
- MARQUES, M. C. P. O. *O ensino da programação no desenvolvimento de jogos através do ambiente Scratch*. Nº de Folhas 177. Dissertação do Mestrado em Ensino de Informática. Universidade do Minho, 2013.
- NOSCHANG, L. F. et al. Portugol studio: Uma ide para iniciantes em programação. In: *Workshop sobre Educação em Computação*. 2014. p. 1287-1296.
- OLIVEIRA, M.; OLIVEIRA, E. Metodologia de Diagnóstico e Regulação de Componentes de Habilidades da Aprendizagem de Programação. In: *Anais do XXII WEI - Workshop sobre Educação em Computação*. 2014.
- PAES, R. B. et al. Ferramenta para a Avaliação de Aprendizado de Alunos em Programação de Computadores. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. 2013.
- PAULA, B. H.; VALENTE, J. A.; BURN, A. O uso de jogos digitais para o desenvolvimento do currículo para a Educação Computacional na Inglaterra. *Currículo sem Fronteiras*, v. 14, n. 3, p. 46-71, 2014.
- PÍCCOLO, H. L. et al. Ambiente Interativo e Adaptável para ensino de Programação. In: *Anais do Simpósio Brasileiro de Informática na Educação*. 2010.
- QUEIRÓS, R.; LEAL, J. P. Ensemble: An Innovative Approach to Practice Computer Programming. In: QUEIRÓS, R., *Innovative Teaching Strategies and New Learning Paradigms in Computer Programming*. Hershey: Information Science, 2015. cap. 9.
- RAMOS, V. et al. A Comparação da Realidade Mundial do Ensino de Programação para Iniciantes com a Realidade Nacional: Revisão sistemática da literatura em eventos brasileiros. In: *Anais do Simpósio Brasileiro de Informática na Educação*. 2015. p. 318.
- RIBEIRO, R. S. *Construção e uso de ambiente visual para o ensino de programação introdutória*. Nº de Folhas 59. Dissertação do Mestrado em Ciência da Computação. Universidade de São Paulo, 2015.
- SCAICO, P. D. et al. Ensino de Programação no Ensino Médio: Uma Abordagem Orientada ao Design com a Linguagem Scratch. *Revista Brasileira de Informática na Educação*, v. 21, n. 02, p. 92, 2013.
- SILVA, P. et al. Um Mapeamento Sistemático sobre Iniciativas Brasileiras em Ambientes de Ensino de Programação. In: *Anais do Simpósio Brasileiro de Informática na Educação*. 2015. p. 367.
- TAVARES, P. et al. Técnicas para aumentar o envolvimento dos alunos na aprendizagem da programação. In: *VII Congresso Mundial de Estilos de Aprendizagem*. Bragança, 2016. p. 1565-1577.