

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.