

# ***Pepy: Sistema para apoiar os Processos de Ensino e de Aprendizagem de Lógica de Programação utilizando Python***

**Rodrigo Perlin<sup>1</sup>, Ricardo Tombesi Macedo<sup>2</sup>, Sidnei Renato Silveira<sup>2</sup>**

<sup>1</sup>Curso de Bacharelado em Sistemas de Informação, UFSM – Universidade Federal de Santa Maria – Campus Frederico Westphalen – RS

<sup>2</sup>Departamento de Tecnologia da Informação, UFSM – Universidade Federal de Santa Maria – Campus Frederico Westphalen – RS

rodrigo\_perlin@hotmail.com, rmacedo1987@gmail.com,  
sidneirenato.silveira@gmail.com

**Resumo.** *Este artigo apresenta um sistema, para auxiliar na redução dos índices de evasão nas disciplinas iniciais de programação nos cursos de computação, visto que grande parte da evasão ocasiona-se pela dificuldade dos alunos nestas disciplinas. Com o propósito de apoiar os processos de ensino e de aprendizagem destes conteúdos, criou-se um sistema web que possibilita uma maior interatividade entre os alunos e o professor. O sistema, denominado Pepy, utiliza, como domínio, a linguagem de programação Python. O sistema permite que o professor cadastre questões e os alunos as solucionem, sendo que a correção ocorre, automaticamente, comparando a resposta de acordo com a definição do professor: resultado (saída do código) ou lógica.*

**Palavras-Chave.** *Lógica de Programação; Pepy; Processos de Ensino e de Aprendizagem.*

**Abstract.** *This paper presents a system to help reduce the dropout rates in the discipline of programming logic in Degree Computer Science, since part of the evasion is caused by the students' difficulty in matters involving algorithms and programming logic. With the purpose of supporting the teaching and learning processes, an online system was created that allows greater interactivity to the students and the teacher. The system, called Pepy, using the Python programming language. The system allows the teacher to register questions and the students solve them, and the correction occurs automatically, by the teacher defines: comparing the exit response of the code or logic.*

**Keywords.** *Programming Logic; Pepy; Teaching and Learning Processes.*

## **1. Introdução**

Atualmente, a área de Tecnologia da Informação (TI) está entre as que mais crescem no Brasil. Um grande fator para esse crescimento se caracteriza pelo fato de que, cada vez mais setores diferentes, buscam soluções de seus problemas com o apoio da TI (ARAÚJO, 2018). Em países como a Estônia ou o Reino Unido, por exemplo, existe o ensino de programação, nos currículos escolares. Em muitos outros países há o ensino de algoritmos e lógica de programação desde a Educação Básica, analisando a necessidade do mercado de trabalho, no contexto do pensamento computacional (GARLET, BIGOLIN e SILVEIRA, 2018; SOUZA, SILVEIRA e PARREIRA, 2018; SILVA, 2015). Nesse contexto, existem as projeções da empresa *VisionMobile*, compiladas pela

Universidade de *Richmond*, que em 2020, terá uma demanda de quinze vezes mais profissionais especializados em programação (EL PAIS, 2016).

Todavia, a evasão consiste em um problema que atinge as instituições tanto públicas como privadas, o que provoca grandes desperdícios de recursos, visto que alunos abandonam seus cursos antes de os completarem (HOED, 2017; SILVA FILHO *et al.*, 2007). Nesse contexto o nível de desistência também é um problema nos Cursos de Computação, pois existe um grande número de alunos que abandonam o curso devido ao alto nível de dificuldade no aprendizado de algoritmos e de lógica de programação (PEREIRA e RAPKIEWICZ, 2004). Entretanto, esses conteúdos servem como base para a prática da programação de computadores, cada vez mais necessário, visando automatizar processos e inserir os cidadãos na sociedade do conhecimento, focando nos esforços do aprendizado do pensamento computacional (GUSMÃO, 2011; SBC, 2017; SBC, 2018).

Na literatura, são apresentadas diversas estratégias para aprimorar os métodos tradicionais de ensino de algoritmos. Conforme Hui e Min as escolas e os professores devem encontrar uma linguagem de programação para iniciantes, a fim de facilitar o aprendizado (HUI e MIN, 2017). Neste contexto, foram estudados alguns trabalhos para apoiar o desenvolvimento do sistema *Pepy*. Martins (2018) apresentou o *Robocode* como uma ferramenta adequada para utilização em disciplinas que envolvam Programação Orientada a Objetos. Fernandes *et al.* (2018) apresentaram uma experiência que utilizou a robótica na criação de atividades educacionais. Guanghai *et al* (2018). usaram o método do *Axis Flip* no ensino da linguagem de programação *Python*, com foco em melhorar as habilidades de programação. Entretanto, este trabalho apresenta um sistema *web* desenvolvido para apoiar os processos de ensino e de aprendizagem de algoritmos e lógica de programação, utilizando a linguagem *Python* e gamificação. A linguagem de programação *Python* foi escolhida, pois permite que os alunos compreendam lógicas usadas em jogos de computador ou de celulares, que são *softwares* que realmente fazem parte do cotidiano destes alunos (ROSSUM e DRAKE, 2013).

O restante do artigo está organizado como segue: a seção 2 apresenta alguns trabalhos relacionados ao sistema desenvolvido. A seção 3 apresenta a ferramenta gamificada *Pepy*. Encerrando o artigo, são apresentadas as considerações finais e as referências empregadas.

## 2. Trabalhos Relacionados

Martins (2018) apresenta uma experiência com o uso do jogo *Robocode* como recurso didático, para disciplinas que envolvem o estudo de programação de computadores. O jogo tem, como objetivo, o de desenvolver um tanque de guerra para batalha contra outros tanques em uma arena virtual, utilizando a linguagem de programação Java. O estudo de caso foi realizado no Instituto Federal de Goiás/Câmpus Luziânia, envolvendo a participação de alunos dos segundo e terceiro anos do Ensino Médio do Curso Técnico em Informática para Internet e do Curso de Bacharelado em Sistemas de Informação. Os resultados encontrados demonstram que o *Robocode* propiciou uma maior participação dos alunos, além de criar novas habilidades em programação e raciocínio lógico. Embora exista um maior interesse dos alunos nas atividades acadêmicas durante a utilização do *Robocode*, o jogo não permite ao professor ter um registro de atividades realizadas como, também, não permite cadastrar exercícios para os discentes resolverem.

Fernandes *et al.* (2018) relataram a experiência de um estudante do Curso de Engenharia de Controle e Automação no desenvolvimento de estratégias para apoiar os processos de ensino e de aprendizagem de lógica de programação com uso da robótica educacional, em escolas da rede privada de ensino em Salvador-BA, com alunos dos anos finais do Ensino Fundamental (6º ao 9º anos). O estudo de caso apontou um desenvolvimento da criatividade e pensamento crítico, aflorando habilidades dos estudantes em resolver desafios, utilizando algoritmos, estabelecendo uma sequência de instruções para resolver os problemas. Ainda que, o trabalho desperte a construção do pensamento computacional, seria mais proveitoso trabalhar em uma ferramenta gamificada, que possibilite aplicar os conhecimentos aprendidos de forma mais interativa.

Guanghui *et al.* (2018) apresentam um método denominado *Axis Flip* para apoiar os processos de ensino e de aprendizagem de programação utilizando a linguagem *Python*. O trabalho foca na aliança entre a teoria e a prática já que, conforme os autores do trabalho, o importante não é a quantidade de conhecimento e, sim, possibilitar que o aluno aplique o conhecimento na prática da programação. Os resultados do estudo apontam uma melhoria nas habilidades dos alunos em resolver problemas simples até os mais complexos. Em nosso trabalho, além de visarmos a construção do conhecimento dos alunos (o aluno como sujeito, construtor do seu conhecimento) (SILVEIRA *et al.*, 2019), desenvolvemos um sistema que permite que os alunos resolvam questões (que podem ser cadastradas pelos professores) como, também, podem armazenar as atividades desenvolvidas.

O trabalho aqui apresentado relata o desenvolvimento de uma ferramenta para motivar e apoiar os processos de ensino e de aprendizagem da linguagem de programação *Python*, por meio de um ambiente interativo, que proporcione autonomia aos alunos e, também, aos professores. Para o desenvolvimento do *Pepy* nos apoiamos nos conceitos apresentados nos trabalhos de Fernandes *et al.* (2018) e Martins (2018), que desenvolveram os processos de ensino e de aprendizagem baseados em jogos e atividades interativas, estimulando a prática ao proporcionar que os alunos resolvam problemas de programação. Contudo, a ferramenta *Pepy*, possibilita que o professor acompanhe o desempenho dos alunos. Além disso, o professor pode cadastrar exercícios. A ferramenta *Pepy* compreende os processos de ensino e de aprendizagem na linguagem de programação *Python*, como no trabalho desenvolvido por Guanghui *et al.* (2018), que também estimula a prática. Todavia, a ferramenta *Pepy* ainda conta com o recurso de gamificação, para motivar e estimular os alunos a participarem ativamente dos processos de ensino e de aprendizagem.

### 3. O Sistema *Pepy*

O sistema *Pepy* baseia-se na Teoria de Aprendizagem Construtivista, sendo o aluno sujeito ativo na construção de seu conhecimento, visto que o conceito envolve a evolução do aluno conforme resolve os exercícios cadastrados pelos professores (a assimilação, acomodação e equilíbrio, proposta por Piaget na Teoria Construtivista) acumulando pontuação, permitindo que o aluno avance nos diferentes níveis (SILVEIRA *et al.*, 2019). O sistema, além de somar e mostrar os pontos atuais dos alunos, também mostra quantos pontos ainda faltam para atingir o próximo nível. Existem seis níveis no *Pepy*, definidos de acordo com a pontuação do usuário o sistema, como mostra a Figura 1. Cada nível possui *layout* e características diferentes dos outros, baseando-se nos conceitos de

gamificação. Segundo Kapp (2012), a gamificação é o uso de mecânicas, estética e pensamentos dos *games* para envolver pessoas, motivar a ação, promover a aprendizagem e resolver problemas. Sendo assim, esses estes conceitos foram aplicados no *Pepy*, visando *gamificar* o sistema e torná-lo mais interativo e atrativo aos discentes.




Figura 1. Níveis do *Pepy* (Fonte: os autores, 2019)

O sistema permite resolver exercícios cadastrados pelos professores, sendo que os mesmos são corrigidos de forma automática, sem precisar da interferência do professor, tornando o processo mais ágil. Além disso, o sistema armazena, em um banco de dados, todas as informações, tornando possível a visualização de relatórios de desempenho de turmas ou de um aluno em específico, de acordo com a necessidade e interesse do professor. O sistema tem, como interface inicial, uma área de identificação, pois ele possui dois tipos de usuários (professor e aluno) e algumas funções em suas áreas são diferentes. Após a escolha do tipo de usuário, o sistema conduz o usuário para a tela de *login*, que solicita o nome do usuário e a senha para verificar a existência do cadastro no sistema.

Uma das duas áreas que o usuário pode ser direcionado é a do professor, que disponibiliza o menu “questão”. Esse menu possui várias funcionalidades: cadastro de questão, alteração de questão, excluir questão e ver suas questões. Porém a função mais importante se caracteriza no “cadastro de questão” (Figura 2), que possibilita o cadastro de exercício informando a turma em que deve ser aplicado, o valor correto da saída do exercício ou da lógica procurada na resposta, o valor da pontuação, que o aluno ganhará caso acerte a questão e a escolha do estilo da correção (resultado da execução do código-fonte ou lógica), que é fundamental no momento da avaliação da resposta do aluno, que acontece após ele realizar a questão.

## Cadastro de Questões

Nome do Professor: Uelinton Brezolin


Painel Inicial
Questões ▾
Consulta ▾
Serviços ▾

**Questão:**

Faça um programa que apresente qual é a variável de maior valor.  
 ( Utilize a=10; b=20; a<b )

**Turma:**

**Valor para comparação**  
Por resultado Final

OU

**Lógica que deve ser usada**

if (a) < b:

Cadastre o valor da pontuação do aluno  
Min = 0 Max = 10

**Estilo para comparação:**

Resultado Final  
 Lógica

Cadastrar

**Figura 2. Área de Cadastro de Questões (Fonte: os autores, 2019)**

Além das opções ligadas à manutenção das questões, existem outras funções. A opção “fórum”, que possibilita ao professor visualizar e responder dúvidas dos discentes, o que torna mais fácil a comunicação. A opção “consulta desempenho”, onde o docente pode escolher uma turma específica ou aluno, para mostrar dados de exercícios realizados e, caso for de interesse mudar a correção da tarefa, essas duas opções estão dentro do menu “consulta”.

O menu “serviço”, que é comum em ambas as áreas (professor e aluno) disponibiliza três funções: A “troca de senha”, o “*feedback*”, que é um espaço para o usuário escrever como foi sua experiência ao utilizar o *Pepy* e a função “imprimir”, que possibilita ser impresso qualquer dado de aluno, turma ou questão.

Além da área do professor, existe uma área reservada ao aluno. Esta área, além das funções sobre os exercícios têm, ainda, o menu "ajuda", que conta com um fórum, onde o aluno pode cadastrar suas dúvidas, interagir com outros alunos ou professores. Caso persistam dúvidas, há um tutorial com *links* de vídeo-aulas sobre programação na linguagem *Python*. No entanto, a função para “resolver questão” (Figura 3) que fica dentro do menu “questão”, tem maior importância na área do aluno. Neste espaço, o aluno pode escolher entre todos os exercícios de uma turma ou somente um com seu código de identificação. Após a confirmação da tarefa escolhida, o discente, programando em linhas de código, dentro da própria plataforma, resolverá o exercício e o enviará para correção. Ainda existem outras duas funções auxiliares dentro do menu questão: “excluir resposta” e “ver questões resolvidas”. Esta última possibilita ao aluno ver suas questões e respostas.

## Resolvendo questão

Nome do Aluno: Luciano Candaten Pontuação: 22 Nível: Iniciante (Faltam: 28 para o próximo nível)






Painel Inicial	Questão ▾	Ajuda ▾	Serviços ▾
Codigo Identificação da questão 1	Turma 1	Questão Faça um "hello world"	

Bora resolver?

```
print "hello world"
```

Interpretação:  
hello world

Interpretar
Enviar

Figura 3. Área de resolver questão (Fonte: os autores, 2019)

A correção dos desafios e exercícios ocorre no próprio *Pepy*, de forma automática, mas ela é executada conforme a escolha feita pelo professor no momento do cadastro da questão. Se o estilo escolhido pelo docente for “resultado final”, a resposta de saída do que foi interpretado do código-fonte, escrito pelo aluno, será comparada com o que o professor colocou no campo “valor da comparação”. Se o estilo desejado pelo educador for “lógico”, o ambiente verifica se ocorreu a interpretação do código digitado pelo aluno e, se ela ocorrer, irá procurar a lógica inserida no campo "valor da comparação", verificando se a lógica está correta. A interpretação do código acontece no próprio ambiente *Pepy*, por meio da utilização do *PyPy.js* (PYPYJS.ORG, 2019). O *PyPy.js* permite a utilização do interpretador *PyPy Python* (PYPY.ORG, 2019), criando um arquivo *.js* (*JavaScript*) que pode ser executado na *web*.

#### 4. Considerações Finais

Nos últimos, o aprendizado de algoritmos e de lógica e programação tornou-se um desafio. Ao projetar uma ferramenta com interesse de proporcionar um ambiente gamificado com maior interatividade, visando à autonomia, pretendemos apoiar os processos de ensino e de aprendizagem, sendo o aluno sujeito ativo na construção do seu conhecimento. Em comparação com os métodos tradicionais de ensino, essa ferramenta proporciona uma maior interação entre o conteúdo e os alunos e entre os alunos e os professores. Além disso, o *Pepy* possibilita o acompanhamento do desempenho da turma pelo professor.

Por limitações de tempo, por enquanto somente trabalhamos com a linguagem de programação *Python*, não sendo explorada a possibilidade de ensinar outras linguagens de programação, tais como Java, PHP ou C. Estas linguagens serão inseridas na ferramenta, futuramente. Em relação a melhorias na ferramenta desenvolvida, uma

sugestão possível seria implantar mais níveis e apresentar ao professor onde ocorreu o erro no código do aluno ao resolver a questão.

Para trabalhos futuros, seria interessante, também, o desenvolvimento de um aplicativo *mobile*, para que os alunos tenham mais facilidade no uso da ferramenta. Outra possibilidade é realizar um estudo de caso da aplicação da ferramenta *Pepy* com alunos de cursos superiores de Informática.

A ferramenta desenvolvida foi aplicada com alunos do primeiro, segundo e terceiro ano da Escola Estadual de Educação Básica Sepé Tiarajú, na cidade de Frederico Westphalen - RS. Ao todo, 119 alunos participaram do estudo. A proposta teve, como método de avaliação, os desafios realizados na quarta etapa e um questionário objetivo aplicado no primeiro contato com os alunos e, posteriormente, ao final da quarta etapa. Os resultados do estudo de caso realizado são apresentados em Perlin, Macedo & Silveira (2019).

## Referências

- ARAÚJO, S. (2018). *Trusign: tecnologia de gestão do mercado de serviços da construção civil e de design de interiores - módulo simple design*. Disponível em: <<https://repositorio.ufrn.br/jspui/handle/123456789/26040>>. Acesso em Julho, 2019.
- EL PAIS, E. (2016). *Emprego: Onze novas profissões que darão muito que falar*. disponível em: <[https://brasil.elpais.com/brasil/2016/10/26/tecnologia/1477502097\\_89975\\_1.html](https://brasil.elpais.com/brasil/2016/10/26/tecnologia/1477502097_89975_1.html)>. Acesso em: outubro, 2018.
- FERNANDES, M.; SANTOS, C. A. M., SOUZA, E. E. P.; FONSECA, M. G. (2018). *Robótica Educacional: uma ferramenta para ensino de lógica de programação no ensino fundamental*. Disponível em: <<http://www.br-ie.org/pub/index.php/wie/article/view/7900/5599>>. Acesso em Julho, 2019.
- GARLET, D; BIGOLIN, N. M; SILVEIRA, S. R. (2018). Ensino de Programação de Computadores na Educação Básica: um estudo de caso. *Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica*. v. 9, p. 7, 2018.
- GUANGHUI, Z., YANJUN, L., YIXIAO, T., ZHAOXIA, W., CHENGMING, Z. (2018). Case-based teaching organization for python programming that focuses on skill training. In 2018 *13th International Conference on Computer Science Education (ICCSE)*, pages 1–5. Disponível em: <[https://www.researchgate.net/publication/327782529\\_Case-Based\\_Teaching\\_Organization\\_for\\_Python\\_Programming\\_that\\_Focuses\\_on\\_Skill\\_Training](https://www.researchgate.net/publication/327782529_Case-Based_Teaching_Organization_for_Python_Programming_that_Focuses_on_Skill_Training)>. Acesso em julho de 2019.
- GUSMÃO. C. (2011). Jportugol: uma ferramenta de auxílio à aprendizagem de algoritmos. *Brazilian Educational Technology: Research and Learning*, 2. Disponível em: <<http://inseer.ibict.br/betrl/index.php/betrl/article/view/89>>. Acesso em Julho, 2019.
- HOED, R. M. (2017). *Análise da Evasão em Cursos Superiores: o caso da evasão em cursos superiores da área de computação*. Brasília: UnB – Programa de Pós-graduação em Computação Aplicada. (Dissertação de Mestrado). Disponível em: <<http://repositorio.unb.br/handle/10482/22575>>. Acesso em março, 2019.

- HUI, C.; MIN, C. (2017).. On Thoughts and Explorations of Training Emerging Engineering Talents in Comprehensive Universities. *Research in Higher Education of Engineering*, 02):19-23+47
- KAPP, K. M. (2012). *The Gamification of Learning and Instruction: Game-based methods and strategies for training and education*. San Francisco: Pfeiffer.
- PERLIN, R.; MACEDO, R. T.; SILVEIRA, S. R. Uma Abordagem Construtivista no Ensino de Algoritmos e Lógica de Programação com o Auxílio de uma Ferramenta Gamificada. *Revista E-Xacta*, v. 12, n. 1, 2019. Disponível em: <<https://revistas.unibh.br/dcet/article/view/2731>>. Acesso em setembro, 2019.
- SILVA FILHO, R. L.; MOTEJUNAS, P. R.; HIPÓLITO, O.; LOBO, M. B. C. (2007). A evasão no ensino superior brasileiro. *Cadernos de Pesquisa*, v. 37, n 132, p. 641-659. Disponível em: <<http://www.scielo.br/pdf/cp/v37n132/a0737132.pdf> >. Acesso em julho de 2019.
- MARTINS, E. (2018). Uso do Robocode no Ensino de Programação em um Curso Médio e Superior. *Tecnia*, 3(1):198–211. Disponível em: <<http://revistas.ifg.edu.br/tecnica/article/view/90>>. Acesso em Julho, 2019.
- PEREIRA, J. C. R., RAPKIEWICZ, C. (2004). *O Processo de Ensino-Aprendizagem de Fundamentos de Programação: Uma Visão Crítica da Pesquisa no Brasil*, WEI RJES.
- PYPY.ORG (2019). *Bem vindo ao PyPy*. Disponível em: <<https://pypy.org/>>. Acesso em: março, 2019.
- PYPYJS.ORG. (2019). *PyPy.js*. Disponível em: <<https://pypyjs.org/>>. Acesso em: março, 2019.
- ROSSUM, V. G., DRAKE, F. L. (2013) *The Python Language Reference*. Disponível em: <<http://docs.python.org/2/reference/>>. Acesso em julho, 2019.
- SBC. Sociedade Brasileira de Computação. (2017). *Referenciais de Formação em Computação: Educação Básica*. Disponível em: <<http://www.sbc.org.br/documentos-da-sbc/send/131-curriculos-de-referencia/1166-referenciais-de-formacao-em-computacao-educacao-basica-julho-2017>>. Acesso em maio, 2019.
- SBC. Sociedade Brasileira de Computação. (2018) *Diretrizes para o Ensino de Computação Básica*. Documento Interno da Comissão de Educação Básica da SBC.
- SILVA, C. (2015). *Introdução ao Ensino de Lógica de Programação para Crianças do Ensino Fundamental com a Ferramenta Scratch*. Dissertação de Mestrado. Universidade Federal De Roraima.
- SILVEIRA, S. R.; PARREIRA, F. J.; BIGOLIN, N. M.; PERTILE, S. L. (2019) *Metodologia do Ensino e da Aprendizagem em Informática*. Santa Maria: UAB/NTE/UFSM.
- SOUZA, N. G.; SILVEIRA, S. R.; PARREIRA, F. J. (2018) Proposta de uma metodologia para apoiar os processos de ensino e de aprendizagem de Lógica de Programação na modalidade de Educação a Distância. *ECCOM – Educação, Cultura e Comunicação*. v. 9, p. 207.