

Sistema ELoS - Ensino de Lógica de Programação com apoio de um Sistema Gráfico

Rodrigo Oliveira Ferrari, André Luís Braga Dutra, Vitória Ferreira César,
Rayssa Amaral Gomes, Rodrigo Luis de Souza da Silva¹

¹Departamento de Ciência da Computação – Universidade Federal de Juiz de Fora (UFJF)
Juiz de Fora – MG – Brasil

{rodrigo.ferrari, andre.dutra, vitoria.cesar, rayssa, rodrigoluis}@ice.ufjf.br

Abstract. *The ELoS (Teaching Logic in Graphic Systems) is a system designed to creatively and interactively introduce the fundamental principles of programming logic. The system employs a gaming-inspired approach, featuring challenging levels and stages that students must navigate by inputting commands in a versatile programming language. The ELoS was developed for web browsers, utilizing technologies such as HTML, CSS, JavaScript, and ThreeJS. In this study, the ELoS was implemented in 14 classes of the DC5199 course (Algorithms - Practice) at the Federal University of Juiz de Fora, engaging students from various disciplines in the exact sciences field. An initial assessment was conducted, gathering data on participant profiles, including age, completion time, and prior programming experience. The results revealed a diverse participation, with a significant presence of female participants and a majority in the early stages of their academic paths. The system catered to both novice learners and those with some previous programming exposure, offering a valuable learning opportunity for students with different backgrounds. Based on these findings, we can confidently assert that the ELoS has demonstrated its potential in enhancing the teaching and learning experience in programming.*

Resumo. *O ELoS (Ensino de Lógica em Sistemas Gráficos) é um sistema desenvolvido para introduzir os conceitos básicos de lógica de programação de forma lúdica e interativa. O sistema utiliza uma abordagem semelhante a videogames, com níveis e fases desafiadoras que os alunos devem completar através da digitação de comandos em uma linguagem de programação própria. O ELoS foi implementado em navegadores web, utilizando tecnologias como HTML, CSS, JavaScript e ThreeJS. Neste estudo, o ELoS foi implementado em 14 turmas da disciplina DC5199 (Algoritmos - Prática) na Universidade Federal de Juiz de Fora, envolvendo estudantes de diversos cursos da área de exatas. Uma avaliação inicial foi realizada, coletando dados sobre o perfil dos participantes, como idade, tempo de conclusão e experiência prévia em programação. Os resultados mostraram uma participação diversificada, com presença significativa de mulheres no estudo, estando a maioria dos participantes em estágios iniciais de suas trajetórias acadêmicas. O sistema atendeu tanto a alunos iniciantes quanto àqueles com alguma experiência prévia em programação, oferecendo uma oportunidade de aprendizado para diferentes perfis de estudantes. Com base nessas informações, podemos afirmar que o ELoS demonstrou potencial para auxiliar o ensino e aprendizagem de programação.*

1. Introdução

O aprendizado de lógica de programação pode ser desafiador, especialmente para estudantes iniciantes. Uma das dificuldades encontradas é a transição de uma tarefa em linguagem natural para uma forma lógica e algorítmica em linguagem de programação. Softwares educacionais têm como objetivo superar essas dificuldades, oferecendo suporte ao ensino de programação.

Uma abordagem promissora para o desenvolvimento de softwares educacionais é tornar o aprendizado mais lúdico e divertido. Aplicações como Scratch ou Lightbot buscam apresentar os fundamentos da lógica de programação de forma simplificada, sem as complexidades da sintaxe e gramática das linguagens convencionais, oferecendo uma maneira descontraída de aprendizado [Kuz and Ariste 2022]. Além disso, eles exigem pouco ou nenhum conhecimento prévio dos usuários, o que contribui para maior inclusão no ensino de computação, expandindo o aprendizado para além dos contextos técnicos ou universitários. Embora os softwares educacionais lúdicos não possam aprofundar muito além dos conceitos básicos em comparação com abordagens mais formais de ensino de lógica de programação, eles se apresentam como uma porta de entrada interessante, despertando o interesse por estudos mais aprofundados para aqueles que estão tendo seu primeiro contato com o assunto.

De acordo com a análise de [Kuz and Ariste 2022], é comum encontrar o uso de linguagens de programação em blocos em ferramentas educacionais lúdicas. Essas linguagens desempenham bem o papel introdutório para estudantes no campo da computação. No entanto, surgem desafios quando esses estudantes precisam avançar em seus estudos e começar a aplicar seus conhecimentos em linguagens de programação escritas. Em [Moors et al. 2018], os autores destacam as barreiras enfrentadas pelos alunos ao tentar fazer essa transição. Uma das barreiras é a falta de confiança em escrever em linguagens com sintaxes complexas, uma vez que as linguagens de programação em blocos eliminam a possibilidade de erros de sintaxe. Além disso, as linguagens em bloco podem iludir os alunos ao permitir a criação de tarefas complexas de maneira simples, em comparação com o que é feito nas linguagens escritas. Portanto, é importante que os professores que desejam utilizar softwares que empregam esse tipo de linguagem em suas aulas analisem cuidadosamente como evitar a dependência excessiva dos alunos por elas, garantindo que eles se sintam estimulados a avançar em seus estudos.

Para evitar esse problema em um software educacional lúdico, uma abordagem eficiente pode ser utilizar uma linguagem de programação escrita, apresentando os conceitos e possíveis dificuldades que o aluno poderá enfrentar, mas de forma gradual. Essa abordagem permite que o estudante se familiarize gradualmente com as ferramentas que usará em aplicações acadêmicas ou profissionais, oferecendo um primeiro contato com o estudo da lógica de programação, preparando-o também para os próximos passos em seu aprendizado.

O presente artigo tem como objetivo apresentar o Sistema ELoS (acrônimo para **Ensino de Lógica em Sistema Gráficos**), um software educacional que utiliza a computação gráfica como recurso tecnológico para o ensino de lógica de programação. O sistema busca proporcionar um primeiro contato lúdico e divertido com a programação, atendendo a estudantes de diferentes níveis de escolaridade, especialmente alunos do ensino fundamental e médio. Desenvolvido para ser executado em navegadores web, o

sistema é de fácil acesso, podendo ser utilizado em diversos dispositivos como *desktops*, *smartphones* e *tablets*, necessitando para tal apenas de uma conexão com a internet. A abordagem adotada utiliza uma linguagem de programação escrita para emular a experiência de programar em uma linguagem real, introduzindo gradualmente os recursos gerais de lógica de programação, como atribuição de comandos, estruturas condicionais e de repetição.

2. Trabalhos Relacionados

Ao longo dos últimos anos, várias iniciativas foram propostas com o objetivo de ensinar lógica de programação a alunos nos mais diversos níveis, indo do ensino fundamental ao ensino médio.

Em [Saito et al. 2019], objetivando analisar os efeitos do aprendizado de programação em texto com crianças do ensino fundamental, foi introduzido um curso que utilizou a linguagem Python em conjunto com o Raspberry Pi. O curso foi composto de sete assuntos variando de noções básicas até o desenvolvimento de um produto, com o resultado exposto no último tópico como uma apresentação. Neste período, 19 alunos foram preparados para criar um projeto acerca de um problema real na programação. Houve três métodos de avaliação, dois questionários feitos pelas crianças (pré-teste e pós-teste) e rubricas que eram preenchidas pelo professor. Observou-se ao final uma melhora significativa no entendimento de programação. É válido mencionar que o estudo de caso passou por muitos problemas na obtenção de parâmetros, já que com o método avaliativo escolhido, não foi possível estimar nota em muitos assuntos abordados no curso.

Introduzir o ensino de programação com jogos digitais foi o foco do projeto em sala de aula apresentado em [Seralidou and Douligeris 2021]. Em uma turma de ensino fundamental, os alunos aprenderam conceitos de programação utilizando o Scratch, uma ferramenta online que funciona por programação em blocos. O jogo criado chamava-se “Objects game” e nele os estudantes relacionaram as ideias de um conteúdo complexo da programação orientada à objetos com um método interativo e didático. Um questionário de satisfação foi aplicado aos jovens que mostraram-se muito satisfeitos com o conteúdo.

Foi apresentado em [Lin et al. 2022] a plataforma LV4LP como ferramenta de aprendizagem assistida para alunos do ensino médio, tendo como objetivo ensinar lógica através de palestras tradicionais e aulas de programação ao vivo. Houve a participação de 38 alunos, que já haviam cursado pelo menos um semestre de programação fundamental anteriormente. Durante as aulas na plataforma, divididas em conceitos de programação e tutorias ao vivo, o aluno poderia realizar um teste das questões e era incentivado a realizar anotações. Como resultado os alunos assistiram 17408 vídeos e realizaram 632 anotações. Além disso foi realizada a aplicação de um pré-teste e um pós-teste, onde foi identificada uma melhora nas habilidades, principalmente em relação a sintaxe.

Foi proposto em [Silva et al. 2019] um desafio utilizando a técnica de Computação Desplugada (CD), com o objetivo de estimular e verificar o raciocínio lógico dos alunos das aulas de programação nas escolas públicas. O desafio foi aplicado em 5 escolas públicas, sendo três turmas de Ensino Fundamental compostas por 108 alunos com idade entre 9 a 14 anos e duas turmas de Ensino Médio compostas por 90 alunos com idade entre 15 a 17 anos. A primeira CD aplicada, “Labirinto de Programação”, foi realizada com tabuleiro impresso, baseado em atividades lúdicas do portal *code.org*, tendo como objetivo

principal encontrar o caminho utilizando a quantidade mínima de retângulos. A primeira etapa da atividade foi realizada de forma individual e na segunda os estudantes formaram grupos. Os resultados das primeiras oficinas não foram apresentados no trabalho.

Em [de Mattos and Ferreira 2021] foi proposto um workshop apenas para meninas do terceiro ano do Ensino Médio, visando introduzir o ensino de programação e influenciar as jovens a seguir uma carreira na área de computação. O curso foi aplicado em uma escola da Rede Estadual de Educação na cidade de São Carlos. As aulas foram realizadas em um laboratório de informática disponível na própria instituição. O curso foi estruturado em 12 encontros aos sábados, com atividades de programação utilizando a ferramenta Scratch e com ambientes de programação com objetivos semelhantes como o Alice, Game Maker, Kodu e Greenfoot, além de palestras com profissionais na área de TI. O conteúdo foi organizado por complexidade. Para a coleta de dados foram realizados questionários, registros de atividades e entrevistas. As alunas avaliaram o curso como excelente, relatando que o mesmo foi importante para praticar o raciocínio lógico e para aprender vários aspectos relacionados a Tecnologia da Informação. O único ponto negativo relatado foi que a ferramenta Scratch era muito infantil para a idade delas. Ainda assim, elas consideraram que programar com a ferramenta era agradável e os conteúdos eram de fácil compreensão.

O sistema apresentado neste trabalho se difere dos demais por apresentar uma linguagem própria, em português, e uma visualização rica ao empregar uma biblioteca gráfica totalmente em 3D, com o intuito de motivar sua utilização por parte dos alunos.

3. Referencial Teórico

As duas principais abordagens para o ensino de lógica de programação focam na utilização de blocos ou texto para a criação dos algoritmos. A seguir serão indicadas as principais características dessas duas abordagens.

3.1. Ensino com Linguagem de Programação em Blocos

O ensino através de linguagens de programação em blocos é uma abordagem que utiliza recursos visuais para substituir a escrita tradicional de comandos, visando enfatizar a lógica de programação ao invés da sintaxe. Os comandos são apresentados ao usuário como um quebra-cabeça e, para atribuir e conectar esses comandos, basta clicar e arrastá-los para a posição correta. Essa abordagem tem se tornado popular em ambientes educacionais para introduzir alunos à programação, pois dispensa a necessidade de empregar sintaxe complexa e estruturação rígida, tornando o código mais compreensível. Além disso, softwares educacionais que utilizam essa abordagem disponibilizam os comandos de forma mais acessível para os programadores.

Embora essa abordagem seja uma alternativa viável para superar as barreiras iniciais do aprendizado de lógica de programação, estudos mostram que novos obstáculos surgem para aqueles que desejam aprofundar seus estudos, especialmente ao passarem para linguagens escritas [Moors et al. 2018]. O contato tardio com a sintaxe de uma linguagem escrita pode abalar a confiança do aluno em prosseguir com seu aprendizado.

Além disso, eliminar o desafio de lidar com uma linguagem de programação escrita nem sempre se mostrou motivador para o ensino, como observado em [DiSalvo 2014]. Um estudo foi realizado sobre um treinamento com estudantes do ensino

médio usando uma linguagem de programação escrita chamada Jython e um ambiente de programação em blocos, o Alice. Em entrevista, os alunos afirmaram que preferiram a linguagem Jython ao invés do ambiente de programação em blocos, alegando que o último era mais lento e trabalhoso de usar durante o treinamento, além de ser menos versátil.

3.2. Ensino com Linguagem de Programação Escrita

Como apontado na seção anterior, privar os alunos do contato com sintaxes de linguagens de programação escritas pode resultar em dificuldades nos estudos mais avançados, e em [Weintrop and Wilensky 2015] é apresentado como o contato frequente com programação escrita mostrou resultados positivos. Utilizando a ferramenta Snap!, um ambiente que combina a linguagem de programação em blocos com a programação escrita, os participantes de um curso de Ciência da Computação no ensino médio puderam visualizar a tradução dos blocos para a linguagem escrita e até mesmo programar diretamente em linguagem escrita. Os resultados indicaram que essa abordagem híbrida foi eficaz na introdução ao ensino de programação e estabeleceu uma base sólida para estudos posteriores.

Contudo, ao utilizar linguagens de programação escrita na introdução à lógica de programação, é crucial escolher uma linguagem adequada para garantir a eficácia do ensino. Linguagens com estruturas e sintaxes complexas podem dificultar o aprendizado inicial. Em [Mladenović et al. 2018] foi realizado um estudo comparativo entre turmas do ensino básico que aprenderam programação em diferentes linguagens, como a linguagem em blocos Scratch e as linguagens escritas Logo e Python. Os resultados mostraram que as turmas que aprenderam com linguagens escritas, especialmente Python, tiveram mais dificuldades em exercícios de identificação de estruturas de repetição aninhadas. A hipótese levantada é que linguagens fortemente baseadas em indentação, como Python, são mais desafiadoras para iniciantes na programação.

4. Sistema ELoS

Para introduzir os alunos à lógica de programação de maneira lúdica e divertida, o Sistema ELoS foi concebido para apresentar os conceitos básicos de lógica de programação com dinâmica semelhante à vista em jogos eletrônicos. O sistema foi desenvolvido para ser executado em navegadores web. Para o design das páginas, foram utilizadas as linguagens de marcação e estilo HTML e CSS, com o auxílio do framework de interfaces web Bootstrap. A programação foi feita em Javascript, e a parte que envolve computação gráfica foi implementada com a biblioteca gráfica ThreeJS.

A lógica principal do sistema consiste em operar um robô que tem como principal objetivo coletar um ou mais cristais disponíveis no ambiente apresentado no sistema gráfico. O sistema conta com obstáculos como paredes e armadilhas como ilustrado na Figura 1. A dinâmica do sistema consiste em indicar quais comandos o robô deve realizar para chegar até o cristal. Cada tópico de ensino foi dividido em níveis. O nível 1, por exemplo, introduz o conceito de sequência simples. O nível 2 aborda estruturas condicionais simples e o nível 3 estruturas condicionais com alternativa dupla. Cada nível possui oito fases, que consistem em desafios apresentados em um sistema gráfico. A dificuldade dos desafios aumenta progressivamente. Para resolver os desafios, o usuário deve digitar os comandos necessários em uma caixa de texto. Os comandos foram estruturados de

forma a se assemelham a uma linguagem de programação convencional. Em todos os níveis, os comandos disponíveis são dispostos na interface para que o usuário possa usar como referência ou mesmo copiar e colar na área de execução. Ao executar os comandos, o sistema gráfico reage ao que foi programado, fornecendo *feedbacks* em forma de texto em um console.

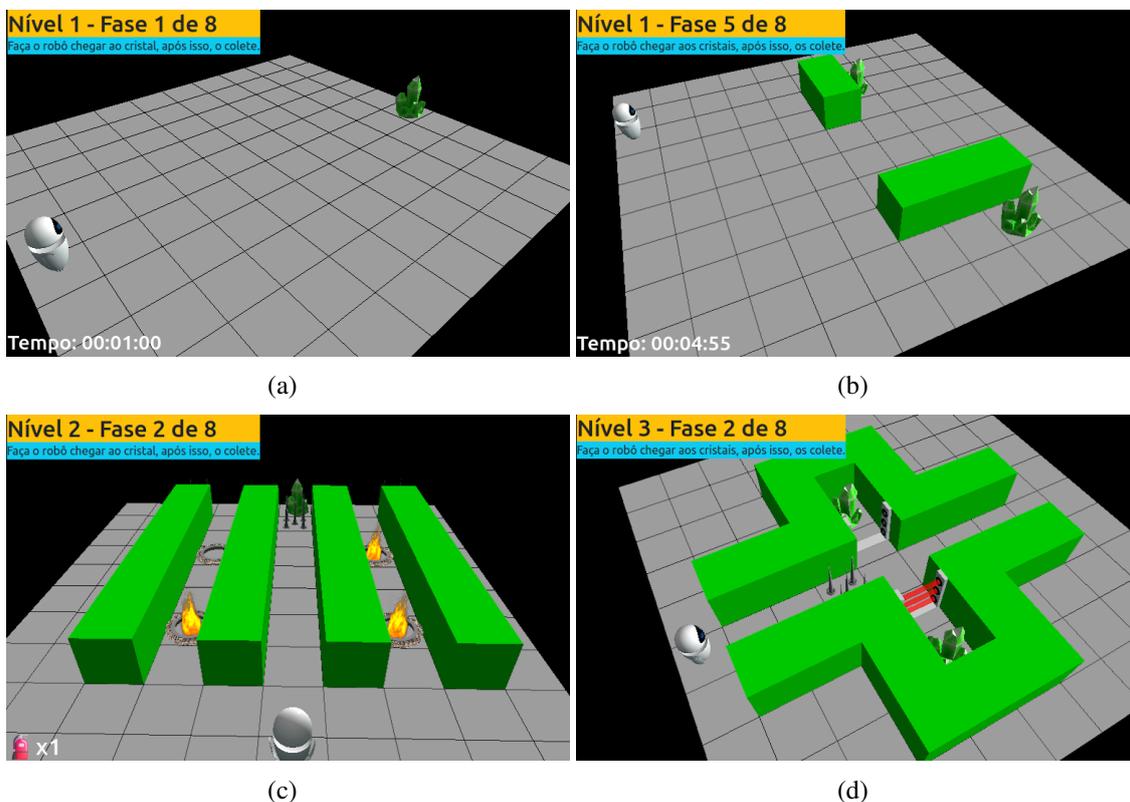


Figura 1. Interface gráfica do sistema apresentando diversos níveis e fases.

Ao acessar o sistema, o usuário é direcionado para a sua página inicial (Figura 2), onde é possível escolher um nível a ser completado e assistir a vídeos de apresentação do projeto e a um breve tutorial sobre como utilizar os recursos do sistema para completar os desafios.

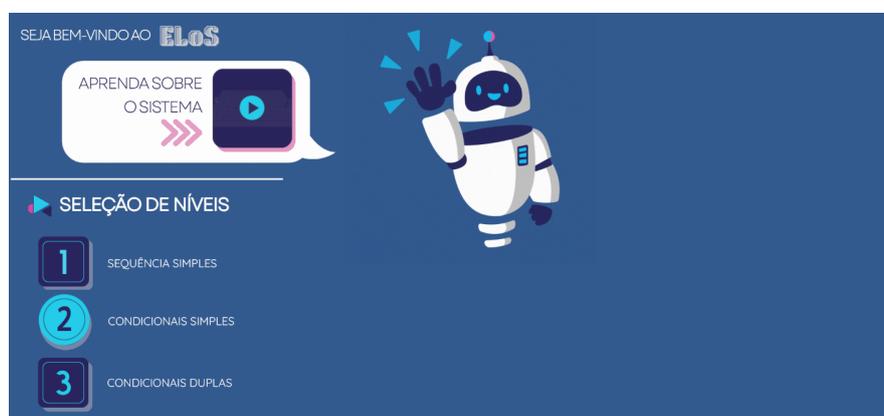


Figura 2. Página inicial do sistema.

Para o design dos níveis, a página foi dividida em duas partes. A primeira metade consiste na parte gráfica (como visto na Figura 1). A segunda metade apresenta a área onde o usuário insere os comandos para controlar as ações do robô (Figura 3).

Funções Disponíveis:

Ações: andarFrente(quantidade) andarTras(quantidade) girarEsquerda() girarDireita() darMeiaVolta() apagarFogo() coletarCristal()	Verificadores: pegandoFogo()	Estruturas: se(condição){ ação() }
--	--	--

Fechar

Digite seu código aqui:

```
1 girarEsquerda()
2 andarFrente(2)
3 se(pegandoFogo()){
4   apagarFogo()
5 }
6 andarFrente(4)
7 girarDireita()
8 andarFrente(1)
9 coletarCristal()
10
```

Executar Reset

Console:

Limpar Console

Figura 3. Área que o usuário edita para interagir com o sistema.

Na área de interação há, na parte superior, uma caixa que pode ser expandida para exibir todos os comandos que o usuário pode digitar para completar a fase. No centro, há uma caixa de texto onde o usuário deve digitar os comandos. Essa caixa de texto segue o design e as mecânicas padrão de vários editores de linguagens de programação, com coluna de marcação de linhas, tabulação automática e indicadores de início e fim de blocos. Ao terminar de escrever o código, o usuário pode utilizar o botão “Executar” para observar a execução do código na parte gráfica. Por fim, na parte inferior, há um console que fornece *feedbacks* em texto sobre a execução do código digitado.

5. Estudo Experimental

Com o intuito de avaliar a eficácia do ELoS como uma ferramenta de introdução à lógica de programação, um estudo experimental foi conduzido. O estudo foi realizado com alunos de primeiro período de diversos cursos como Ciência da Computação, Engenharia Computacional, Elétrica, Civil etc. O estudo foi realizado durante a primeira aula de uma disciplina de algoritmos, sendo este o primeiro contato destes alunos com lógica de programação em nossa universidade. Os resultados deste estudo contribuirão para a obtenção de *insights* mais precisos sobre a eficácia do ELoS como uma ferramenta de ensino de programação, levando em consideração a interação dos alunos com os jogos e o impacto em sua aprendizagem.

5.1. Definição e Planejamento

Durante as aulas introdutórias de programação na Universidade Federal de Juiz de Fora, o ELoS foi divulgado e implementado experimentalmente como parte do currículo. Para coletar dados dos alunos um questionário foi aplicado ao final da execução do experimento. Por meio desse questionário, foi possível obter informações como a fase completada, nome, idade, tempo de conclusão, gênero e experiência prévia do aluno com programação. Esses dados foram essenciais para análise e avaliação do desempenho dos alunos, bem como para a obtenção de informações relevantes sobre o uso e a eficácia do ELoS como ferramenta de ensino.

O nível 1 do sistema foi introduzido em 14 turmas da disciplina DC5199 (Algoritmos - Prática). Dos alunos participantes, recebemos um total de 263 respostas. No entanto, após validação e análise, constatamos que apenas 229 respostas foram consideradas válidas e utilizadas como base para a análise do estudo.

Por meio deste questionário foi possível coletar dados dos participantes e dividi-los em grupos com base nas informações recebidas, como fase completada, idade, tempo de conclusão, gênero e experiência prévia do aluno com programação. Essa segmentação permitiu uma análise mais detalhada e uma compreensão dos resultados, considerando diferentes variáveis que podem influenciar o desempenho e a percepção dos alunos em relação ao ELoS.

5.2. Caracterização dos Participantes

Os participantes do estudo experimental são graduandos do primeiro período dos diversos cursos atendidos pelo Departamento de Ciência da Computação da instituição.

Considerando as 229 respostas válidas, 68 indivíduos eram do gênero feminino e 161 do gênero masculino, representando uma porcentagem de mulheres de aproximadamente 30%.

A faixa etária predominante dos alunos que cursaram a disciplina e participaram do experimento foi entre 17 e 18 anos (162). Observou-se a presença de 53 alunos com idade entre 19 e 21 anos e 14 alunos com idade superior a 22 anos.

Por último, em relação à experiência prévia com programação, observamos que a maioria dos participantes teve seu primeiro contato com lógica de programação no próprio experimento (126 alunos). Alguns alunos tiveram pequena experiência de programação no colégio (41) e uma quantidade mais relevante já havia desenvolvido pequenos projetos (56). Apenas 6 alunos se consideraram programadores com alguma experiência.

5.3. Análise Quantitativa

Como explicitado na seção anterior, a maior parte dos participantes teve sua primeira experiência com programação ao utilizar o ELoS. Esses participantes tiveram a maior média de tempo para completar o nível 1 (27m28s). Como esperado, os resultados comprovam que usuários que nunca programaram apresentaram mais dificuldades em realizar as tarefas solicitadas se comparado a alunos que já tiveram contato com programação. Os alunos que tiveram alguma experiência com programação ou que já haviam desenvolvido pequenos projetos tiveram médias de 23m02s e 23m55s, respectivamente. Aqueles que se consideravam programadores experientes finalizaram o nível mais rápido (19m19s), como esperado.

Em relação à análise por idade, não foi possível identificar um padrão que indicasse que um determinado grupo foi mais eficiente que outro. A variação entre o grupo mais rápido (22 a 23 anos - 20m40s) e o mais lento (acima de 23 anos - 28m09s) não é conclusiva, visto que os grupos mais jovens (17 a 18 anos e 19 a 21) tiveram médias piores que o grupo mais eficiente (25m56s e 24m15s, respectivamente).

5.4. Correlação entre as variáveis

Para finalizar a análise dos dados foi criada uma matriz de correlações, ilustrada na Tabela 1. Os valores com asterisco (*), referentes às correlações, indicam que as medidas são estatisticamente significativas, pois os valores de *p-value* são menores que 0,05. Dessa forma, é possível fazer uma análise sobre esses resultados.

Através da Tabela 1 é possível notar que as únicas variáveis que possuem algum tipo de correlação são as variáveis experiência e tempo. Como os valores são negativos em ambos os coeficientes, à medida que uma variável aumenta, a outra diminui, indicando inicialmente que um usuário mais experiente completaria o nível 1 do ELoS em menos tempo. Entretanto, mesmo que as medidas sejam significativas, as intensidades das correlações entre as variáveis são consideradas fracas, pois não passam de 0,3. Desse modo, não podemos afirmar que a experiência de um usuário influenciou no seu tempo para completar o nível 1 do sistema.

		Idade	Tempo	Experiência
Idade	R de Pearson	--		
	p-value	--		
	Rho de Spearman	--		
	p-value	--		
Tempo	R de Pearson	0.099	--	
	p-value	0.136	--	
	Rho de Spearman	-0.048	--	
	p-value	0.467	--	
Experiência	R de Pearson	0.022	-0.192**	--
	p-value	0.739	0.004	--
	Rho de Spearman	0.096	-0.293***	--
	p-value	0.148	<.001	--

Tabela 1. Matriz de Correlações.

6. Conclusão

Neste trabalho foi apresentado o Sistema ELoS (Ensino de Lógica em Sistemas Gráficos), um sistema desenvolvido com o objetivo de introduzir de forma lúdica e divertida os conceitos básicos sobre lógica de programação. O ELoS foi concebido para proporcionar uma experiência de aprendizado interativa, na qual os alunos enfrentam desafios apresentados em um sistema gráfico. Cada tópico de ensino é dividido em níveis e cada nível contém oito fases de dificuldade progressiva. Os alunos devem digitar os comandos necessários para resolver os desafios em uma caixa de texto, utilizando uma linguagem de programação estruturada de forma semelhante a outras linguagens disponíveis no mercado. O sistema gráfico reage à programação do aluno e fornece *feedbacks* visuais e em um console de texto.

Para o desenvolvimento do ELoS, foram utilizadas as linguagens de marcação e estilo HTML e CSS, juntamente com o framework de interfaces web Bootstrap. A programação foi realizada em Javascript, enquanto a parte gráfica foi desenvolvida em ThreeJS. Essas tecnologias permitiram que o sistema fosse executado em navegadores web modernos, com um layout responsivo em diferentes dispositivos.

A implantação do ELoS proporcionou uma oportunidade de avaliar sua eficácia como ferramenta de ensino. Embora não tenhamos acesso aos dados de satisfação dos alunos, é possível inferir que o sistema tenha proporcionado uma abordagem mais envolvente e prática para o aprendizado de lógica de programação. A natureza lúdica e interativa do sistema, aliada à progressão gradual dos desafios, pode ter despertado o interesse dos alunos e facilitado a compreensão dos conceitos abordados.

Além disso, o uso de navegadores web como plataforma de execução ofereceu uma ampla acessibilidade, permitindo que os estudantes utilizassem o ELoS em diferentes dispositivos, de acordo com suas preferências individuais. Essa flexibilidade e conveniência podem ter contribuído para uma maior autonomia dos alunos e um engajamento mais efetivo com o processo de aprendizagem.

Como trabalhos futuros, sugere-se o aprimoramento contínuo do ELoS com base nos *feedbacks* dos alunos e em análises mais detalhadas de sua eficácia. A inclusão de novos recursos e funcionalidades pode enriquecer ainda mais a experiência de aprendizagem e atender às necessidades específicas dos estudantes. Além disso, a expansão do sistema para outras áreas do conhecimento e o aprofundamento de estudos sobre o impacto do ELoS no desempenho dos alunos podem ser explorados. Finalmente, pretende-se aplicar o projeto em escolas de nível fundamental e médio onde as tratativas iniciais já foram iniciadas.

7. Agradecimento

Os autores deste trabalho gostariam de expressar sua sincera gratidão à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (Fapemig) pelo suporte financeiro fornecido para a realização desta pesquisa (Edital 011/2022 - APOIO A PROJETOS DE EXTENSÃO EM INTERFACE COM A PESQUISA, Processo APQ-03903-22). O apoio da Fapemig foi fundamental para o desenvolvimento deste estudo, permitindo a investigação de novas perspectivas e contribuindo para o avanço do conhecimento em nossa área de pesquisa.

Referências

- de Mattos, F. and Ferreira, V. (2021). Despertando o interesse em meninas pela computação com o ensino de programação em um ambiente apoiador. *Revista Psicologia e Transdisciplinaridade*, 1(1):39–58.
- DiSalvo, B. (2014). Graphical qualities of educational technology: Using drag-and-drop and text-based programs for introductory computer science. *IEEE computer graphics and applications*, 34(6):12–15.
- Kuz, A. and Ariste, M. C. (2022). Análise e revisão de softwares educacionais para a aprendizagem da programação em ambientes lúdicos. *Tecné, Episteme y Didaxis: TED*, 1(52):117–136.
- Lin, Y.-T., Yeh, M. K.-C., and Tan, S.-R. (2022). Teaching programming by revealing thinking process: Watching experts' live coding videos with reflection annotations. *IEEE Transactions on Education*, 65(4):617–627.
- Mladenović, M., Boljat, I., and Žanko, Ž. (2018). Comparing loops misconceptions in block-based and text-based programming languages at the k-12 level. *Education and Information Technologies*, 23:1483–1500.
- Moors, L., Luxton-Reilly, A., and Denny, P. (2018). Transitioning from block-based to text-based programming languages. In *2018 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*, pages 57–64. IEEE.
- Saito, D., Washizaki, H., Fukazawa, Y., Yoshida, T., Kaneko, I., and Kamo, H. (2019). Learning effects in programming learning using python and raspberry pi: Case study with elementary school students. In *2019 IEEE International Conference on Engineering, Technology and Education (TALE)*, pages 1–8. IEEE.
- Seralidou, E. and Douligieris, C. (2021). Motivating students in distance programming learning using games. In *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pages 1–7. IEEE.
- Silva, A. V. B., Paixão, A. C. P., Souza, T. V. S., Coelho, M. R. S., Homci, M., Rodrigues, G., Kawasaki, R., and Couto, D. (2019). Manas digitais: um relato sobre ensino de programação em escolas públicas no estado do pará. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 8, page 367.
- Weintrop, D. and Wilensky, U. (2015). To block or not to block, that is the question: students' perceptions of blocks-based programming. In *Proceedings of the 14th international conference on interaction design and children*, pages 199–208.