

# Introdução à Lógica de Programação com o Sistema ELoS – Análise de uso no ensino superior

**Marcos Vinícius Santos de Lima, Letícia Miyuki Ferreira Yamashita,  
Lara Gonçalves de Souza, Rodrigo Luis de Souza da Silva**

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

{marcos.santos, yamashita.miyuki, larasouza.goncalves}@estudante.ufjf.br  
rodrigo.luis@ufjf.br

**Abstract.** *The use of graphical systems for teaching programming logic offers a playful and interactive way to introduce learners to computing. ELoS is a system that presents programming concepts through a game using commands similar to a programming language. In this study, the complete version of the system will be presented, and the competencies addressed at each level will be described. Finally, a quantitative analysis of the system's application in higher education will be presented and discussed.*

**Resumo.** *O uso de sistemas gráficos para ensino de lógica de programação é uma alternativa lúdica e interativa para promover um primeiro contato com a computação. O ELoS é um sistema que aborda conceitos de programação por meio de um jogo que utiliza comandos semelhantes a uma linguagem de programação convencional. No presente estudo, a versão completa do sistema será apresentada e as competências abordadas em cada nível do sistema serão descritas. Ao final, uma análise quantitativa da utilização do sistema no ensino superior será apresentada e discutida.*

## 1. Introdução

O déficit de profissionais com conhecimento de lógica de programação vem aumentando ao longo dos últimos anos [Feijao et al. 2021]. Nesse sentido, tem sido cada vez mais ampla a adoção de novas metodologias de ensino nesta área. Os benefícios do ensino de lógica de programação, no entanto, vão muito além da sua relevância profissional. Isso porque, tendo conceitos que abrangem todas as ciências, o contato com a lógica de programação possibilita o desenvolvimento de diversas habilidades essenciais para qualquer indivíduo, como a capacidade de dedução e resolução de problemas e organização lógica do raciocínio. Além disso, esse processo viabiliza uma maior inclusão digital, uma vez que apresenta contextos frequentemente só acessados no ensino superior, despertando assim um maior interesse pela área [Scherer et al. 2021].

Segundo [Costa et al. 2016], investir no ensino de programação desde cedo é essencial para o aprimoramento do raciocínio lógico de alunos do ensino fundamental e médio. Por outro lado, apresentar esse conteúdo para crianças e adolescentes pode gerar diversos desafios, visto que trata-se, na maioria das vezes, do primeiro contato desses estudantes com aspectos de programação, sendo necessário o cuidado para que a complexidade esteja alinhada ao nível de compreensão dessa faixa etária. Assim, é interessante

implementar métodos de aprendizagem que integram a computação à vida escolar de maneira lúdica e intuitiva, para que o tema não se torne maçante e os estudantes possam desenvolver o raciocínio lógico com autonomia e criatividade.

O sistema ELoS (acrônimo para **E**nsino de **L**ógica em **S**istema **G**

O presente artigo tem duas contribuições principais. A primeira é apresentar a versão completa do sistema, que é composto por quatro níveis. A segunda é apresentar uma análise quantitativa da aplicação do sistema ao longo de 2 anos na disciplina de Algoritmos oferecido pelo departamento de Ciência da Computação da UFJF.

Este artigo está organizado da seguinte maneira. A Seção 2 apresenta trabalhos relacionados com foco em ensino de lógica de programação. A Seção 3 apresenta o sistema, com ênfase nas competências abordadas em cada um dos níveis. Na Seção 4 a análise da aplicação do sistema é apresentada seguida da seção que conclui este trabalho.

## 2. Trabalhos Relacionados

Ao longo dos últimos anos, várias iniciativas foram propostas com o objetivo de ensinar lógica de programação a estudantes de todos os níveis. A seguir serão destacadas algumas dessas iniciativas e ao final serão discutidas as principais diferenças em relação ao proposto neste artigo.

O artigo [Yuen et al. 2019] explora o uso do ASP (*Answer Set Programming*) para introduzir alunos do ensino médio à Ciência da Computação, com foco no desenvolvimento de habilidades de resolução de problemas e representação do conhecimento. O estudo utilizou a ferramenta ASPIDE, um ambiente de programação para ASP, junto com a abordagem SPARC, que facilita o aprendizado ao organizar programas em seções de tipos, predicados e regras. Essa estrutura promove uma metodologia explícita de modelagem que ajuda os alunos a estruturar seu pensamento logicamente e identificar erros. O público-alvo incluiu estudantes do ensino médio (10º a 12º ano) do programa TexPREP, um curso de verão oferecido pela *Texas Tech University* para jovens de 11 a 18 anos com interesse em áreas de STEM (ciência, tecnologia, engenharia e matemática).

O objetivo do projeto proposto em [Prates et al. 2023] foi melhorar o ensino de programação através do desenvolvimento e aplicação de material didático para o curso “Introdução ao Pensamento Computacional”, conduzido de forma remota através do *Scratch*, adequado para alunos do ensino fundamental e médio. O conteúdo foi dividido em cinco módulos, abordando conceitos digitais e tecnológicos com exemplos práticos do dia a dia. A primeira atividade consistiu na criação de um personagem visualmente representativo pelos alunos, seguido por outras atividades que exploraram os pilares do

pensamento computacional, como a criação de cenários, diálogos, *chatbots*, linguagens e jogos digitais. Ao final do curso, foi aplicado um formulário para avaliar o progresso dos alunos e coletar *feedback*. Observou-se um alto nível de entusiasmo e interesse dos alunos nas atividades, indicando uma maior afinidade com a área de tecnologia.

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. 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 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. 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 artigo [Silva et al. 2020] propõe um jogo educacional para estudantes que nunca tiveram contato com lógica de programação. O jogo proposto no artigo foi um RPG chamado “Projeto Turing”. O objetivo do jogo era coletar pergaminhos, itens e realizar missões, ensinando conceitos de lógica de programação ao longo da narrativa. Um experimento subjetivo foi realizado por meio de uma pesquisa com 19 estudantes de ensino superior, e as principais conclusões indicaram que os estudantes tiveram boas experiências durante o jogo, o que impactou positivamente sua aprendizagem.

Os autores de [Mathrani et al. 2016] utilizaram o jogo *Light Bot* para ensinar práticas de desenvolvimento de software, como planejamento, programação, teste e depuração, além de tópicos de programação como estruturas condicionais, funções, es-

truturas de repetição e recursão. O jogo emprega um conjunto de comandos que são utilizados para movimentar o robô. Os comandos são inseridos de forma semelhante à forma como são inseridos no Scratch. O ambiente do *Light Bot* é classificado como pseudo-3D (ou 2.5D), por utilizar gráficos isométricos para criar a ilusão de profundidade e tridimensionalidade.

Sumarizando as diferenças dessas iniciativas em relação à apresentada aqui, [Yuen et al. 2019, Bouhnik and Giat 2009, Lin et al. 2022, Silva et al. 2019] não utilizam sistemas gráficos no ensino de lógica e [Prates et al. 2023, de Mattos and Ferreira 2021] utilizaram ferramentas baseadas em bloco como o Scratch. Já os trabalhos apresentados em [Silva et al. 2020] e [Mathrani et al. 2016] tem algumas semelhanças com a proposta apresentada neste artigo. Em [Silva et al. 2020] foi utilizado um jogo para ensino de lógica. Porém o jogo era bidimensional e a forma de trabalhar os conceitos de lógica não se assemelha ao uso de linguagens de programação reais. Já [Mathrani et al. 2016] utiliza um sistema lúdico mais próximo desta proposta, mas não é um sistema tridimensional real e a forma de aplicar os conceitos de lógica também difere bastante de uma linguagem de programação padrão, sendo este um dos objetivos do sistema apresentado neste artigo.

Desta forma, o sistema apresentado neste trabalho se difere dos demais por apresentar uma linguagem textual própria, em português, e uma visualização tridimensional real interativa, com o intuito de ensinar as estruturas básicas de lógica de programação de forma semelhante a que os alunos utilizariam na prática em um curso convencional de lógica com linguagens reais.

### 3. Sistema ELoS

Uma versão preliminar do sistema ELoS, apresentada em [Ferrari et al. 2023], foi desenvolvida para apresentar conceitos básicos de programação. Nesta versão apenas o primeiro nível do sistema foi apresentado. Também em [Ferrari et al. 2023] foram apresentados os fundamentos pedagógicos que orientaram as decisões sobre a escolha do tipo de linguagem, formas de interação etc. O sistema pode ser acessado via *github*<sup>1</sup> ou via plataforma MECRED<sup>2</sup>.

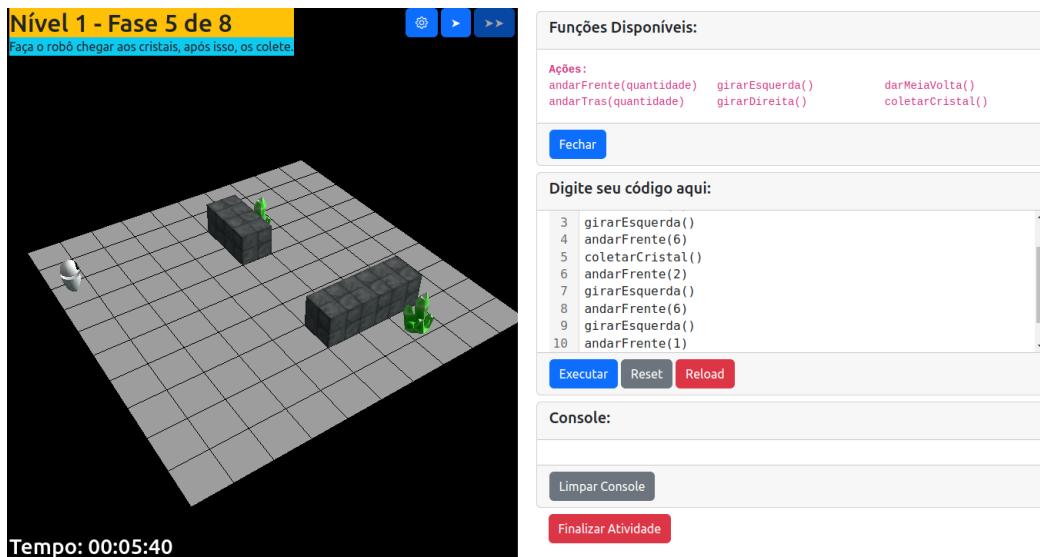
No presente artigo os demais níveis do sistema serão apresentados e ilustrados. A lógica principal do sistema consiste em operar um robô que tem como principal objetivo coletar um ou mais cristais disponíveis no ambiente. Cada nível do sistema tem como finalidade introduzir um dos conceitos de programação através de alguns obstáculos que devem ser superados para alcançar os cristais. Cada nível possui um total de 8 fases com dificuldade progressiva.

O nível 1 do sistema tem como objetivo apresentar o conceito de sequência simples. Nesse nível não é necessário interagir dinamicamente com obstáculos, sendo necessário apenas evitá-los para alcançar os cristais. Para tal, o usuário pode fazer apenas movimentos e ações simples do robô, como, por exemplo, mover em qualquer direção e coletar o cristal. O número de linhas de código necessário para completar cada fase desse nível varia de 2 (fase 1) a 19 (fase 8). A visão completa do usuário ao utilizar o sistema está representada na Figura 1. Nesta figura é possível identificar a área de execução do

<sup>1</sup><https://avrgroup.github.io/ELoS/>

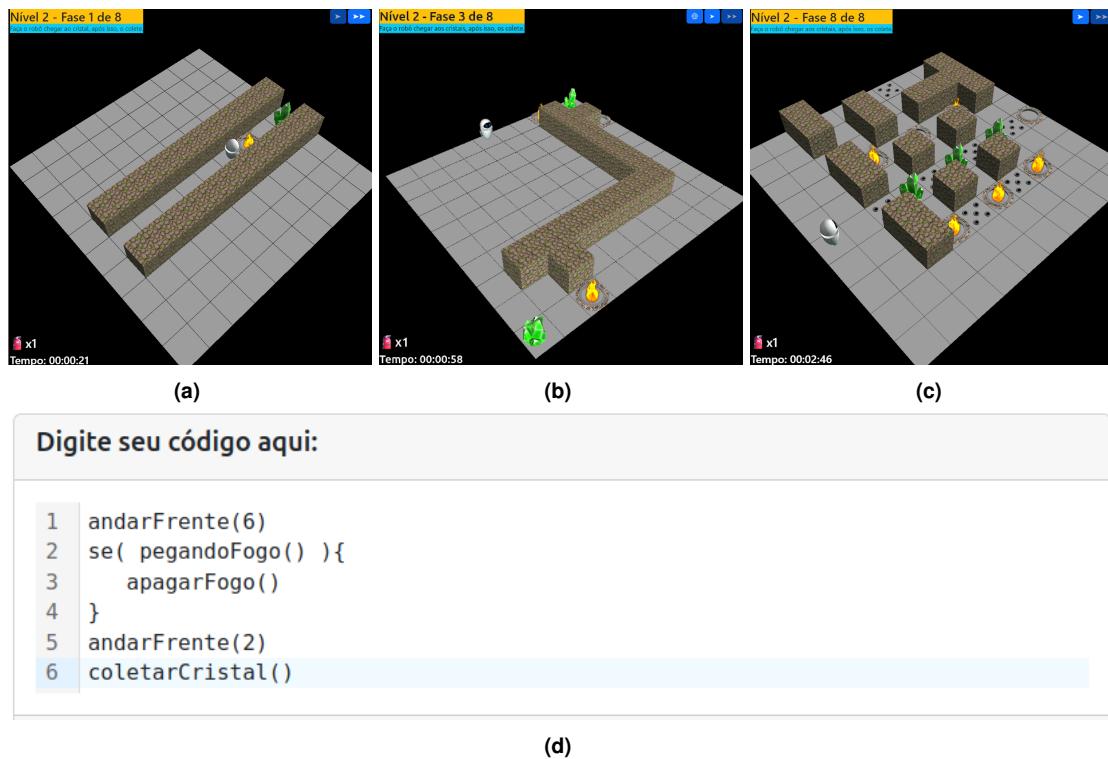
<sup>2</sup><https://mecred.mec.gov.br/recursos/364202>

sistema, a área onde os códigos são inseridos, as funções disponíveis naquele nível e um console onde informações são apresentadas ao usuário.



**Figura 1.** Visão geral dos elementos que compõem a interface do sistema durante sua utilização.

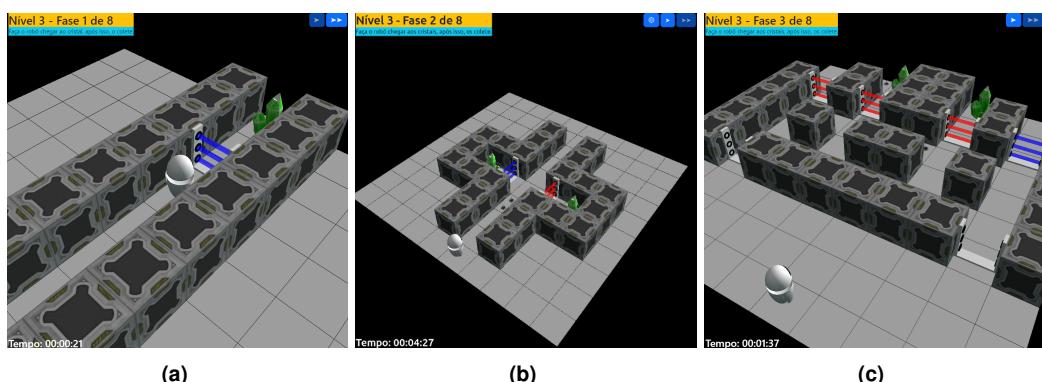
No nível 2, as estruturas condicionais simples são abordadas. Essas estruturas são necessárias para decidir se o robô deve ou não apagar pequenas fogueiras que podem aparecer em seu caminho (Figura 2).



**Figura 2.** Nível 2 do sistema, ilustrando as fases um (a) e três (b) e oito (b) do nível e o código necessário para concluir a fase um (d).

Para lidar com essas fogueiras, o robô possui uma quantidade limitada de extintores que são capazes de apagar o fogo e o usuário não tem como saber, ao executar o sistema, quais fogueiras estarão acesas. Desta forma, através do uso das estruturas condicionais simples, é possível verificar se existe ou não uma fogueira acesa a frente do robô, usando o extintor somente no caso positivo. O número de linhas para completar esse nível varia de 6 a 35.

No nível 3 as estruturas condicionais duplas são apresentadas (Figura 3). Neste nível, os obstáculos são lasers que podem ter duas cores diferentes (vermelho e azul). O robô possui comandos para desativar cada um dos lasers, mas para isso é necessário que o laser a ser desativado seja indicado na função. O usuário é capaz de utilizar comandos para verificar a cor do laser que se encontra na frente do robô, e com o uso das estruturas condicionais duplas tomar a decisão correta para avançar. Neste nível, o número de linhas para completar cada fase variam de 9 a 72.



(a)

(b)

(c)

Digite seu código aqui:

```

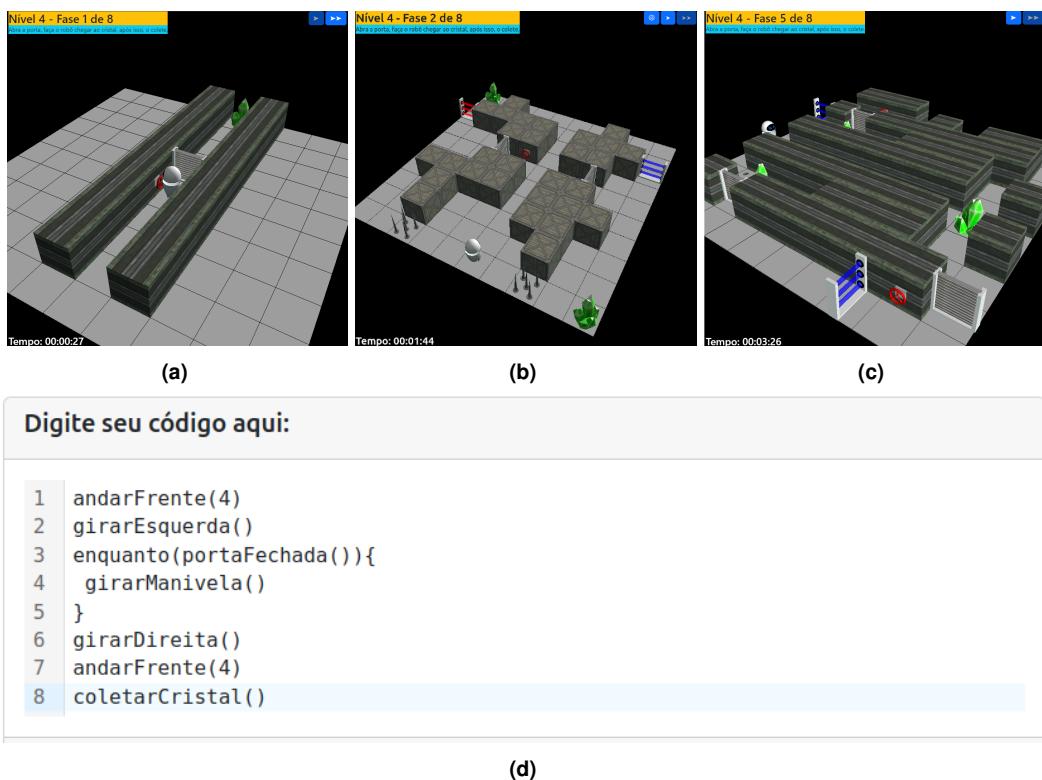
1 andarFrente(6)
2 se(laserAzulAtivo()){
3   desativarLaserAzul()
4 }
5 senão{
6   desativarLaserVermelho()
7 }
8 andarFrente(2)
9 coletarCristal()

```

(d)

**Figura 3. Representação das fases um (a), dois (b) e três (c) do nível 3 e o código para conclusão da fase 1 (d).**

No nível 4, as estruturas de repetição são introduzidas. Nesse nível os principais obstáculos são portas que podem ser abertas através de uma determinada quantidade de interações com uma manivela (Figura 4). As repetições podem ser utilizadas para realizar essas interações até que a porta esteja completamente aberta. São necessárias de 8 a 96 linhas de código para finalizar esse nível. Outra característica importante do nível 4 é que ele utiliza os conceitos abordados de todos os níveis anteriores, fazendo com o discente tenha que identificar qual estrutura lógica deve utilizar para superar os obstáculos apresentados em cada fase.



**Figura 4. Representação das fases um (a), dois (b) e cinco (c) do nível 4 e o código necessário para concluir o nível 1 (d).**

#### 4. Análise da aplicação do sistema

O sistema ELoS está em uso como primeira ferramenta para ensino de lógica de programação nas diversas turmas da disciplina de Algoritmos, conduzidas pelo departamento de Ciência da Computação da UFJF ao longo dos últimos 2 anos. Esta disciplina faz parte do currículo de diversos cursos atendidos por este departamento.

Os alunos são apresentados ao sistema na segunda metade da primeira aula prática da disciplina de Algoritmos. O tempo previsto para a realização da atividade é de 40 minutos. Considerando a limitação do tempo e conhecimento dos alunos, os mesmos são instruídos a utilizar apenas o nível 1 do sistema, que versa sobre sequência simples. Contudo, os demais níveis ficam disponíveis para alunos que eventualmente terminem este nível e tenham interesse em continuar.

##### 4.1. Ameaças à Validade

Durante o planejamento deste estudo, buscou-se minimizar ameaças que pudessem impactar ou limitar a validade dos resultados obtidos [Wohlin et al. 2000]. No entanto, não é possível garantir que tais ameaças não tenham afetado os resultados. Desta forma, as ameaças identificadas no contexto deste estudo são descritas a seguir.

O estudo não foi executado em um único dia por todos os participantes, em função do horário das aulas práticas da disciplina de Algoritmos. Isto pode ter influenciado os resultados, já que não é possível confirmar o tempo que cada turma tinha disponível para a utilização do sistema.

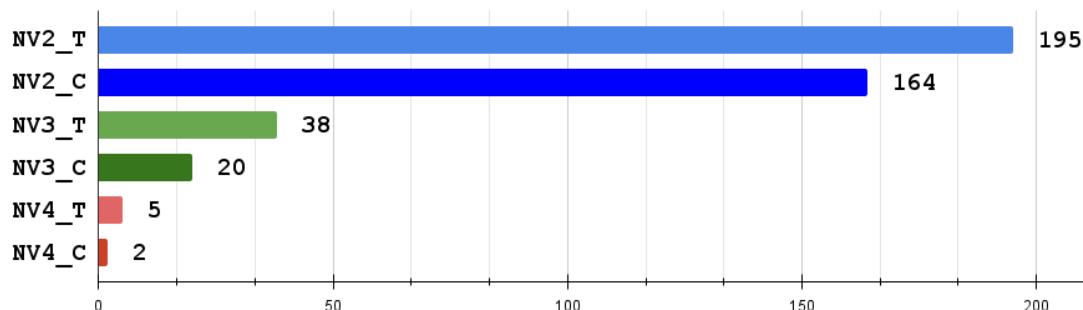
Não foi possível aferir o nível de conhecimento dos alunos que utilizaram o sistema. É possível que alunos com conhecimento de programação ou alunos repetentes na disciplina de Algoritmos utilizaram o sistema juntamente com alunos com pouca ou nenhuma experiência. Essa discrepância de conhecimento prévio pode influenciar na análise do tempo gasto para concluir um determinado nível do sistema.

#### 4.2. Análise Quantitativa

Para a realização da análise quantitativa utilizou-se das informações coletadas ao final de cada nível, a saber, nome do usuário, idade e tempo de conclusão do nível. O tempo de conclusão de um determinado nível é enviado automaticamente pelo sistema ao submeter as informações supracitadas e foi utilizado unicamente para a realização desta análise.

Nos últimos 2 anos, o nível 1 do sistema foi utilizado por mais de 700 alunos. O tempo médio de conclusão desse nível foi de 00:21:14. É interessante notar que alguns alunos finalizaram o nível em menos de sete minutos, enquanto alguns poucos alunos demoraram cerca de uma hora para conclusão do nível.

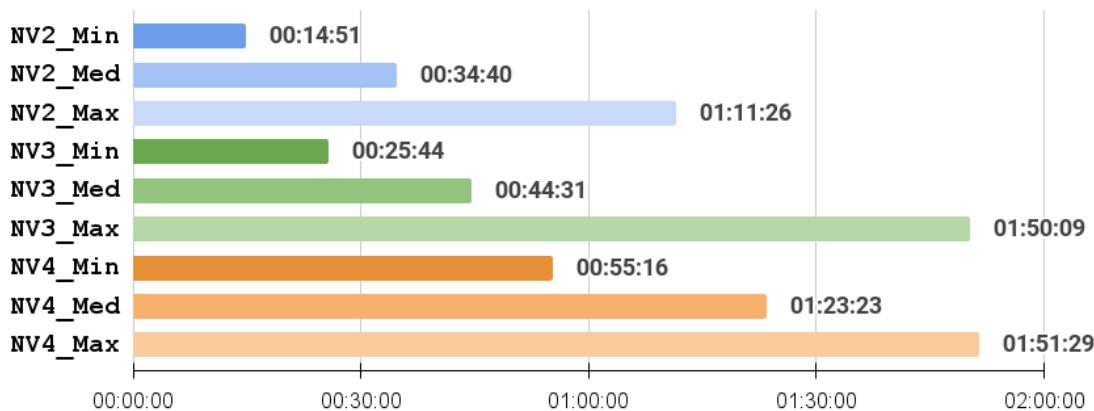
O sistema ELoS foi projetado para ser um sistema que motivasse o aluno a utilizá-lo. Esse fator motivacional pode ser observado ao analisar o número de alunos que acessou o sistema fora do horário de aula para tentar concluir os demais níveis. A Figura 5 sumariza o quantitativo de alunos que tentou utilizar os níveis mais avançados do sistema. Pode-se observar que o nível 2 foi acessado por quase um terço dos alunos e a grande maioria conseguiu concluir o nível.



**Figura 5. Gráfico sumarizando o número de tentativas e conclusões dos níveis 2, 3 e 4. No gráfico, NVX\_T e NVX\_C representam número de alunos que tentaram e concluíram o nível X, respectivamente.**

Em relação ao tempo de conclusão dos níveis, a Figura 6 ilustra os tempos mínimo, médio e máximo que os alunos demoraram para concluir os níveis 2, 3 e 4. Pode-se notar, analisando os tempos de conclusão dos níveis, um razoável nível de engajamento dos alunos na utilização do sistema mesmo fora do ambiente escolar. Vale ressaltar que os conceitos abordados nos níveis 2, 3 e 4 não são triviais para alunos que nunca programaram e cada nível conta apenas com um pequeno vídeo introduzindo esses conceitos.

Em relação às implicações do uso do sistema nos próximos períodos letivos, desde a primeira aplicação do mesmo nas turmas da disciplina de Algoritmos algumas alterações pontuais foram sugeridas pelos professores que ministram a disciplina. Essas alterações já foram incorporadas na versão corrente que é considerável estável.



**Figura 6.** Gráfico summarizando o tempo mínimo (NVX\_Min), médio (NVX\_Med) e máximo (NVX\_Max) de conclusão dos níveis 2, 3 e 4.

É possível inferir que a ferramenta se mostrou adequada para introduzir estruturas elementares de lógica de programação, considerando que a maioria dos alunos que tentou utilizar o sistema foi bem-sucedida. O nível 4, consideravelmente mais complexo, talvez demande uma introdução mais elaborada para que os alunos possam utilizá-lo fora do ambiente escolar.

#### 4.3. Limitações do sistema

O sistema ELoS foi desenvolvido e pensado para ser utilizado em computadores. Apesar de ser possível utilizá-lo em sistemas móveis, esta utilização não é indicada pois as interfaces não foram projetadas para telas de tamanho reduzido, sendo essa a principal limitação de utilização do artefato.

### 5. Conclusão e trabalhos futuros

Este estudo analisou a utilização do sistema ELoS com alunos matriculados na disciplina de Algoritmos, ministrada no primeiro período de diversos cursos atendidos pelo Departamento de Ciência da Computação da UFJF.

Este trabalho traz duas contribuições em relação ao trabalho anteriormente publicado. A primeira é a apresentação dos 3 níveis finais que compõem o sistema. A segunda contribuição é uma análise quantitativa dos dados coletados ao longo de 2 anos de uso da ferramenta. A análise realizada sugere que o sistema possui bom nível de engajamento, uma vez que uma quantidade considerável de alunos utilizou os diversos níveis disponíveis do sistema fora do horário de aula.

Finalmente, como trabalhos futuros, pretende-se fazer uso deste sistema com alunos do ensino fundamental, em seções de maior tempo e com uma breve introdução aos conceitos a serem abordados em cada nível. A ideia é utilizar o sistema em quatro encontros de duas horas cada. Também há a expectativa de criar uma versão do sistema específica para utilização em sistemas móveis, com interfaces mais adequadas para este tipo de dispositivo.

## Referências

- Bouhnik, D. and Giat, Y. (2009). Teaching high school students applied logical reasoning. *Journal of Information Technology Education. Innovations in Practice*, 8:1.
- Costa, T., Cristiano, F., Martins, D., and da Silva, W. (2016). A importância da computação para alunos do ensino fundamental: Ações, possibilidades e benefícios. In *Anais do Workshop de Informática na Escola*, volume 22, pages 593–601.
- 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.
- Feijao, C., Flanagan, I., Van Stolk, C., and Gunashekhar, S. (2021). *The global digital skills gap: Current trends and future directions*. RAND.
- Ferrari, R. O., Dutra, A. L. B., César, V. F., Gomes, R. A., and da Silva, R. L. d. S. (2023). Sistema elos-ensino de lógica de programação com apoio de um sistema gráfico. In *Anais do XXXIV Simpósio Brasileiro de Informática na Educação*, pages 1580–1590. SBC.
- 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.
- Mathrani, A., Christian, S., and Ponder-Sutton, A. (2016). Playit: Game based learning approach for teaching programming concepts. *Educational Technology & Society*, 19:5–17.
- Prates, J. M., Valle, P. H. D., Melo, S. M., and Santos, G. K. M. (2023). Inserção do pensamento computacional em alunos do ensino fundamental e médio. *BARBAQUÁ*, 5(9):70–84.
- Scherer, R., Siddiq, F., and Sánchez-Scherer, B. (2021). Some evidence on the cognitive benefits of learning to code. *Frontiers in Psychology*, 12:559424.
- 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.
- Silva, J. P., Silveira, I. F., Kamimura, L., and Barboza, A. T. (2020). Turing project: An open educational game to teach and learn programming logic. In *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2000). *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers, Norwell, MA, USA.
- Yuen, T. T., Reyes, M., and Zhang, Y. (2019). Introducing computer science to high school students through logic programming. *Theory and Practice of Logic Programming*, 19(2):204–228.