My Trace Table e My Trace Table Manager: Sistemas para Apoiar a Compreensão de Códigos em Disciplinas Introdutórias de Programação

Anna Gabriela de M. Souza[1], José Elkandro do N. Silva[1], Ayla Débora Dantas de S. Rebouças[1]

[1]Departamento de Ciências Exatas (DCX) –
Universidade Federal da Paraíba (UFPB) – Campus IV
Cep 58297-000 – Rio Tinto – PB – Brasil

anna.moura@dcx.ufpb.br, elksandro.nascimento@dcx.ufpb.br, ayla@dcx.ufpb.br

Abstract. *The learning process in programming can be developed using code tracing, but the manual evaluation of trace table exercises hinders the learning cycle. This article presents My Trace Table and My Trace Table Manager, web systems designed for trace table practice with automatic correction. My Trace Table provides feedback using different colors to identify correct answers, value errors, and type errors. My Trace Table Manager, in turn, allows the customization of exercises by the instructor. Initial evaluation demonstrated high engagement of students and the effectiveness of fast feedback.*

Resumo. *O aprendizado de programação pode ser desenvolvido utilizando o rastreamento de código (code tracing), mas a avaliação manual de testes de mesa dificulta o ciclo de aprendizagem. Este artigo apresenta o My Trace Table e o My Trace Table Manager, sistemas web concebidos para a prática de testes de mesa com correção automática. O My Trace Table oferece um feedback com diferentes cores para identificar acertos, erros de valor e erros de tipo. O My Trace Table Manager, por sua vez, permite a personalização de exercícios pelo professor. A avaliação inicial demonstrou elevado engajamento dos estudantes e eficácia do feedback rápido.*

1. Introdução

Segundo D. Santos (2024), o processo de formação de estudantes iniciantes em programação não se limita apenas à escrita de código, sendo de suma importância que o estudante compreenda como o código se comporta e como ele funciona. Neste contexto, a habilidade de rastreamento de código (*code tracing*) — entendida como o processo cognitivo de simular a execução de um programa, linha por linha, para determinar o estado das variáveis e a saída final [Hassan et al. 2023] — é amplamente reconhecida como uma competência essencial, sendo considerada uma habilidade precursora ou de nível intermediário na hierarquia de tarefas de programação, posicionada entre a leitura de código e a escrita de código [Lopez et al. 2008]. No entanto, os estudantes frequentemente enfrentam dificuldades em executar exercícios de rastreamento de forma sistemática, tendendo a se perder em estruturas de repetição (*loops*) e falhando em manter o controle de múltiplas variáveis [Hassan et al. 2023]. Além disso, a avaliação manual de atividades de programação pode ser um processo demorado [D. Santos 2024] e resultar em *feedback* lento ou ausente, comprometendo o ciclo de aprendizagem [Russell 2022]. Assim, as dificuldades envolvidas para execução e correção do

rastreamento de código indicam a necessidade de mais estrutura e orientação nesse processo.

Diante da necessidade de uma ferramenta que forneça a prática estruturada para a habilidade de *code tracing*, este artigo apresenta o *My Trace Table* (disponível no endereço: https://mytracetable.a4s.dev.br), um sistema *web* desenvolvido para auxiliar estudantes de disciplinas introdutórias de programação através de exercícios do tipo teste de mesa (*trace table*). O teste de mesa é a metodologia estruturada que utiliza uma tabela para registrar o valor das variáveis a cada passo do código [Teubl e Zampirolli 2023; Risha et al. 2021]. Para permitir que exercícios de diferentes tipos e configurados de maneira específica possam ser cadastrados e gerenciados por professores para promover autoria pedagógica [Pontes 2007], foi construído o sistema *web My Trace Table Manager* (disponível no endereço: https://mytracetable.a4s.dev.br/manager), também apresentado neste trabalho. Esses sistemas foram inspirados na metodologia de testes de mesa no papel aplicados em sala de aula pela professora Vanessa Dantas da Universidade Federal da Paraíba (UFPB).

O *My Trace Table* e o *My Trace Table Manager* representam a evolução de diferentes trabalhos [J. Santos 2023; D. Santos 2024; Silva Jr. 2025] realizados por estudantes da UFPB, transformando o protótipo inicial de D. Santos (2024) em um sistema funcional, com novas funções e sendo uma versão alternativa à abordagem em formato de aplicativo nativo *Android* [J. Santos 2023; Silva Jr. 2025] ao optar por uma arquitetura *web* responsiva para possibilitar o uso em diferentes dispositivos. O *My Trace Table Manager* concede ao professor controle total sobre o exercício, permitindo a criação e personalização de conteúdo em linguagens como Python e Java e concretiza parte dos trabalhos futuros descritos nos trabalhos anteriores [J. Santos 2023; D. Santos 2024; Silva Jr. 2025]. O desenvolvimento desses sistemas ocorreu no contexto do projeto de extensão Apps4Society(A4S)[1] da UFPB, que objetiva desenvolver aplicativos e sistemas que impactem positivamente a sociedade.

A metodologia para construção e evolução dos sistemas baseia-se na *Design Science Research* (DSR), abordagem epistemológica que legitima o desenvolvimento de um artefato computacional como meio para a produção de conhecimento científico [Pimentel et al. 2019]. Um artefato é definido como qualquer coisa projetada para alcançar um objetivo [Peffers et al. 2007], incluindo modelos e *frameworks* [Vaishnavi e Kuechler Jr. 2015]. A DSR, enraizada nas Ciências do Artificial de Herbert Simon [1969], busca modificar a realidade ao criar soluções tecnológicas para problemas práticos [Pimentel et al. 2019]. O desenvolvimento do *My Trace Table* e do *My Trace Table Manager* cumpre o objetivo da DSR de resolver um problema prático — a necessidade de uma ferramenta com *feedback* rápido para a prática de exercícios de teste de mesa e que suporte a autoria pedagógica [Pontes 2007] — enquanto o uso do artefato busca gerar novo conhecimento sobre a eficácia do *feedback* automático que distingue acerto, erro de valor e erro de tipo para apoiar a compreensão de código utilizando teste de mesa. A avaliação inicial apresentada neste artigo foi realizada com estudantes iniciantes em programação, onde puderam utilizar o *My Trace Table,* sendo

[1] https://a4s.dev.br

observados neste uso, e responderam a um formulário online para fornecer dados a fim de analisar alguns aspectos do artefato no contexto real.

Este artigo tem como objetivo apresentar o *My Trace Table* e o *My Trace Table Manager*, e alguns resultados iniciais de avaliação. A estrutura do trabalho está organizada da seguinte forma: a Seção 2 aborda a fundamentação teórica; a Seção 3 apresenta os sistemas; a Seção 4 dá alguns detalhes sobre a arquitetura e tecnologias utilizadas no desenvolvimento; a Seção 5 realiza uma comparação com plataformas correlatas; a Seção 6 detalha os resultados iniciais; e, finalmente, a Seção 7 conclui o trabalho e apresenta propostas de trabalhos futuros.

2. Fundamentação Teórica

O desenvolvimento do *My Trace Table* e do *My Trace Table Manager* encontra seu suporte teórico na Tecnologia do Ensino apresentada por Skinner (1972), que critica as concepções tradicionais de ensino baseadas em metáforas inadequadas como crescimento e aquisição. Na metáfora do crescimento, a aprendizagem é passivamente entendida como algo que se "desenvolve" com o tempo, e a metáfora da aquisição descreve o ensino como uma transmissão de conhecimento, onde o professor transmite e o aluno recebe. Skinner (1972) argumenta que essas visões falham por não oferecerem instrumentos para intervir de modo eficaz no comportamento. Em contraste, Skinner (1972) define ensinar como arranjar contingências de reforço.

Nessa perspectiva, a eficácia do ensino depende diretamente de contingências bem organizadas, sendo o reforço imediato um princípio importante. Skinner (1972, p. 15) afirma que "o lapso de apenas uns poucos segundos entre a resposta e o reforço destroem quase todo o efeito", algo que é uma falha intrínseca da sala de aula tradicional, onde a avaliação manual pode levar horas ou dias. Russell (2022) corrobora que o *feedback* lento ou ausente compromete o ciclo de aprendizagem em programação. Outro princípio é a modelagem por aproximações sucessivas, pela qual um repertório complexo é estabelecido por meio de uma sequência graduada de passos, cada um reforçado na direção do desempenho final. Skinner (1972) observa que a instrução programada é a forma ideal de proporcionar essa gradação, estruturando tarefas em pequenas unidades que exigem respostas frequentes e reforço adequado.

O *My Trace Table* aplica esses princípios à prática de rastreamento de código (*code tracing*), que é uma habilidade correlacionada à performance na escrita de código [Lopez et al. 2008]. A metodologia de teste de mesa [Teubl e Zampirolli 2023; Risha et al. 2021] fornece a estrutura de ocasião e resposta exigida por Skinner (1972), decompondo a execução do código em passos sequenciais e verificáveis, com o ciclo de *feedback* rápido e automático do sistema atuando como o reforço imediato. O *My Trace Table Manage*r, por sua vez, capacita o professor a arranjar essas contingências com funções como configurar os exercícios, definir a sequência de passos e as respostas esperadas, concretizando a proposta de Skinner (1972) de que o ensino é o arranjo sistemático de contingências eficazes.

3. O My Trace Table e o My Trace Table Manager

O *My Trace Table* é um sistema *web* que foi construído para auxiliar os alunos iniciantes de programação na compreensão de códigos por meio de exercícios de teste de mesa com *feedback* automático, como ilustrado pelas Figuras 1 e 2. Em cada atividade, o estudante visualiza uma imagem de código em alguma linguagem de programação (Python ou Java) e deve indicar os valores assumidos por algumas ou todas as variáveis a cada passo da execução, considerando a versão atualmente disponível.

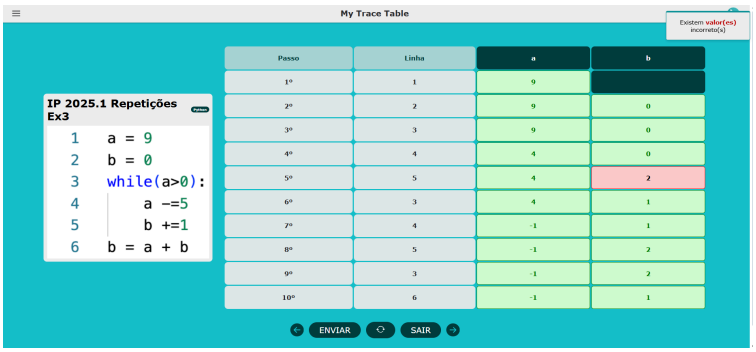

Figura 1. Captura de Tela de Exercício do *My Trace Table* com *Feedback* Indicando Células Corretas e Célula com Erro de Valor. Fonte: Autoria Própria

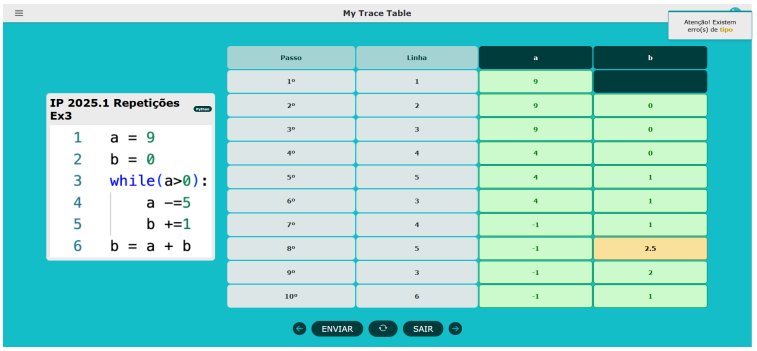

Figura 2. Captura de Tela de Exercício do My Trace Table Com Feedback Indicando Células Corretas e Célula Com Erro de Tipo. Fonte: Autoria própria.

Para o aluno, a experiência é suportada por uma interface *web* simples e interativa, onde o código a ser rastreado é apresentado ao lado ou acima da *trace table*, a depender da largura da tela. O sistema suporta exercícios que cobrem diversos tópicos que variam conforme as preferências do professor, como operadores aritméticos, estruturas condicionais, *Strings* e estruturas de repetição. A principal vantagem que a ferramenta busca trazer ao aprendizado de programação é o ciclo de *feedback* rápido, que se inicia com a submissão das respostas e resulta na correção automática da tabela. O sistema oferece um diagnóstico visual por meio de cores nas células (verde, vermelho ou amarelo) e um texto temporário que informa os acertos ou tipos de erros cometidos.

Um possível cenário de uso é o caso ilustrado na Figura 1 em que ao analisar um código em Python, o aluno comete um erro de cálculo no valor esperado para a variável "b" em um passo específico da execução do programa (Passo 5, referente à linha de código 5). Nesse caso, o sistema marca a célula em vermelho (erro de valor), como

ilustrado na Figura 1, indicando que o valor digitado é diferente do esperado. Contudo, se o aluno, por falta de conhecimento sobre a coerência de tipos, preenche a célula com um valor decimal, como 2.5, quando na verdade espera-se um inteiro, o sistema marca a célula em amarelo (erro de tipo de dado), conforme mostrado na Figura 2 no Passo 8, referente à linha de código 5. Esta distinção visual clara entre os tipos de erro permite que o aluno identifique a natureza de sua falha – se foi na lógica de execução ou no conceito de tipo de dado – facilitando a autocorreção e aprimorando o aprendizado dos fundamentos da programação.

O *My Trace Table Manager*, por sua vez, foi construído com foco nos professores, permitindo que estes possam criar atividades específicas para suas turmas. As Figuras 3 e 4 ilustram algumas de suas telas. Esse sistema oferece ao professor autonomia sobre a criação de temas, que podem corresponder a um conteúdo específico, como Operadores, ou mesmo uma atividade envolvendo diferentes conteúdos (ex: Exercício 1 da Unidade 2 da Disciplina IP).

Além disso, o sistema permite a personalização de exercícios de testes de mesa. O cadastro de um exercício inclui a seleção da imagem com o código a ser analisado, a escolha da linguagem (Python ou Java) e do(s) tema(s) em que o exercício pode ser enquadrado, além de especificações da tabela indicando se devem ser mostradas ou não as colunas de “Passo” e “Linha” ou apenas a coluna referente à linha de código (“Linha”) ou também ao passo de execução (“Passo”). Conforme ilustra a Figura 3, pode-se também configurar outras informações como o nome do exercício, a quantidade de variáveis a mostrar na tabela e a quantidade de linhas que a *trace table* vai ter. Além do preenchimento deste formulário, a criação de um exercício também consiste no preenchimento de três tabelas: a Tabela Mostrada, a Tabela Esperada e a Tabela de Tipos. Na Tabela Mostrada, que é a configuração de como o exercício deve aparecer para o aluno, pode-se usar o caractere “?” para definir as células que devem ser preenchidas pelo estudante e o “*#*” para bloquear células que não devem ser preenchidas. Já a Tabela Esperada representa o gabarito para correção. Nela se indica os valores corretos esperados para as variáveis a cada passo de execução do código do exercício. A Tabela de Tipos indica o tipo esperado para os valores a serem preenchidos em cada célula. Esse processo de preenchimento das tabelas é exemplificado na Figura 4. Além de cadastrar exercícios, os professores cadastrados no sistema com o papel “Admin” podem também cadastrar outros professores.

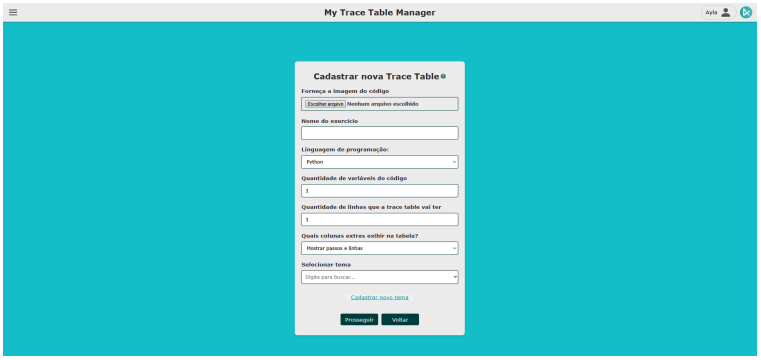

Figura 3. Captura de Tela Inicial de Cadastro de Exercício do *My Trace Table Manager*. Fonte: Autoria própria.

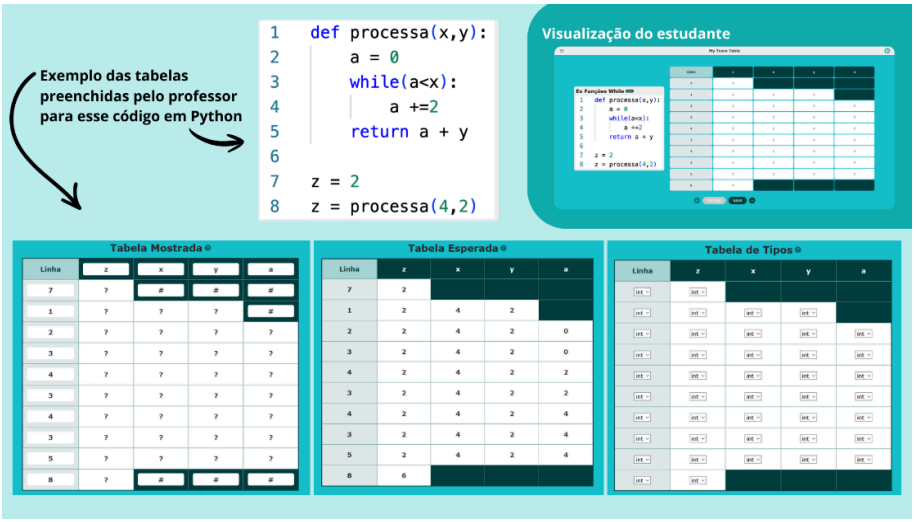

Figura 4. Exemplificação do Preenchimento das Tabelas (Mostrada, Esperada e de Tipos) e a Visualização do Estudante. Fonte: Autoria própria.

4. Arquitetura e Tecnologias utilizadas nos Sistemas

O *My Trace Table* e o *My Trace Table Manager* foram concebidos com foco em facilidade de uso, responsividade e a autonomia do professor e do estudante. Para a construção desses sistemas, foram utilizados a biblioteca ReactJS[2] e o *framework* Spring Boot[3]. O sistema consiste em três componentes principais: o *front-end* (interface do usuário) do aluno, o *front-end* do Professor (*Manager*) e a API RESTful [Richardson et al. 2013] responsável por processar as requisições dos usuários e gerenciar os dados. A arquitetura do sistema está ilustrada em termos gerais na Figura 5.

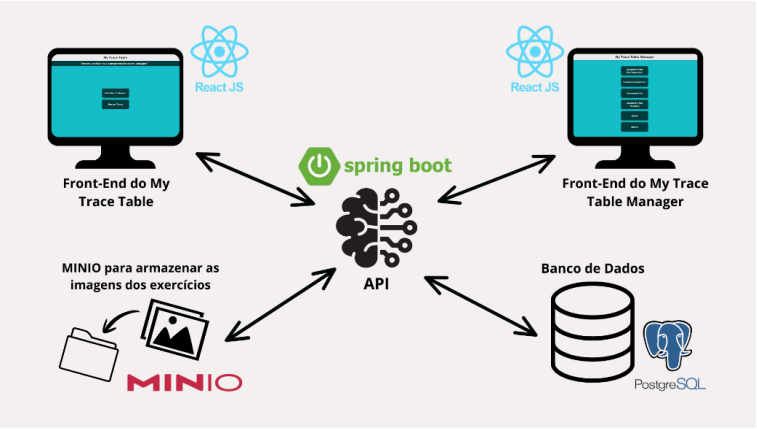

Figura 5. Diagrama da Arquitetura Básica do *My Trace Table* e *My Trace Table Manager*. Fonte: Autoria própria.

A API foi desenvolvida em Java com o *framework* Spring e atua como o *back-end* do sistema, processando as requisições de ambos os *front-ends*. Ela incorpora

[2] https://react.dev/
[3] https://spring.io/projects/spring-boot

módulos como o Spring Data JPA[4] para a persistência de dados no PostgreSQL[5] e o Spring Security[6] para a segurança (autenticação e autorização). Adicionalmente, toda a infraestrutura é feita com *containers* Docker[7], o que facilita a implantação e a manutenção do ambiente de produção. A lógica de correção dos exercícios está centralizada na API. Esta lógica é essencial, pois não só compara a resposta submetida pelo aluno com a Tabela Esperada, mas também realiza a validação do tipo de dado inserido com auxílio da Tabela de Tipos. Para a flexibilidade na adição futura de novos tipos de dados, essa validação foi estruturada por meio do padrão de projeto *Factory* [Gamma et al. 1995]. Além disso, para gerenciar as imagens de códigos dos exercícios, o sistema utiliza o MinIO[8], um serviço externo de armazenamento de arquivos, de modo que o banco de dados armazena apenas a referência única do arquivo para não sobrecarregar. Para cada sistema existe um *front-end* distinto desenvolvido em ReactJS. O *front-end* do *My Trace Table* é utilizado para a exibição de exercícios e do *feedback*. Após o estudante solicitar a correção de um exercício previamente cadastrado pelo professor através do *My Trace Table Manager*, o *front-end* do *My Trace Table* recebe da API um retorno detalhado sobre cada célula, permitindo que a tabela mostrada para o aluno possa ser colorida (verde para correto, vermelho para valor errado e amarelo para erro de tipo).

5. Comparação com Plataformas Correlatas

O *My Trace Table* e o *My Trace Table Manager* representam uma evolução de uma série de recursos educacionais digitais sobre testes de mesa desenvolvidos na UFPB [J. Santos 2023; D. Santos 2024; Silva Jr. 2025], onde puderam ser incorporados *feedbacks* sobre estes considerando diferentes pontos de crítica. O sistema atual herda o nome e o *design* do protótipo *web* de D. Santos (2024) e também compartilha o objetivo central de apoiar o ensino introdutório de programação através de testes de mesa desses trabalhos anteriores, mas com evoluções em diferentes aspectos. Primeiramente, foi implementado o *My Trace Table Manager*, que permite o cadastro dinâmico de exercícios e temas, superando as limitações de conteúdo estático das propostas iniciais [J. Santos 2023; D. Santos 2024; Silva Jr. 2025 ; D. Santos et al. 2025]. Além disso, o sistema foi aprimorado para identificar a natureza do erro, distinguindo entre erro de tipo (amarelo) e erro de valor (vermelho). Esta funcionalidade avança em relação aos trabalhos anteriores, nos quais o feedback era mais generalista, sinalizando apenas a localização do erro, através de símbolos como interrogações [J. Santos 2023] ou células com destaque em vermelho [D. Santos 2024; Silva Júnior 2025], sem categorizar explicitamente o motivo da falha. Outro aspecto é o fato de que com os sistemas atuais, as atividades podem ser realizadas em diferentes tipos de dispositivo, sendo eles móveis ou não, o que não era possível nas versões de aplicativo móvel *Meu Teste de Mesa* de J. Santos (2023) e *My Trace Table Mobile* de Silva Jr. (2025).

[4] https://spring.io/projects/spring-data-jpa
[5] https://www.postgresql.org/docs/
[6] https://spring.io/projects/spring-security
[7] https://docs.docker.com
[8] https://docs.min.io/enterprise/aistor-object-store/

Em um panorama mais amplo, a comparação do *My Trace Table* com plataformas correlatas também revela distinções, especialmente na abordagem à criação de conteúdo e ao mecanismo de *feedback*. A maioria dos sistemas correlatos concentra-se na geração automática e parametrizada de questões a partir de um código base fornecido pelo professor. O *VPL/MCTest*, conforme descrito por Teubl e Zampirolli (2023), utiliza o *MCTest* como gerador de testes, que seleciona, parametriza e distribui algoritmos únicos em pseudo-código para a avaliação teórica de um curso. Da mesma forma, o *Trace Generator*, apresentado por Russell (2022), cria questões parametricamente diferentes em C e Python a cada nova tentativa, focando na prática através da repetição de problemas variados. Até mesmo o sistema de *scaffolding* interativo desenvolvido por Risha et al. (2021), que se integra ao *QuizJet*, gera problemas parametrizados em Java para converter falhas em problemas de rastreamento tradicionais em uma experiência passo a passo. Em contraste com o modelo de geração paramétrica automática adotado por esses sistemas, o *My Trace Table* se baseia na autoria de conteúdo manual do professor por meio do *My Trace Table Manager*. Nesse sentido, o *My Trace Table* oferece um ambiente estruturado para a prática dos testes de mesa e fornece controle total ao professor sobre o conteúdo e estrutura do exercício, além também de lhe permitir utilizar atividades propostas por outros professores. Atualmente o *My Trace Table Manager* pode cadastrar exercícios de teste de mesa em Python e Java, mas pretende-se incluir outras linguagens.

No que tange ao mecanismo de *feedback*, cada sistema adota uma filosofia distinta. O *Trace Generator* [Russell 2022] para questões de linha única (*single-line questions*), utiliza uma estrutura de árvore (*tree-structure*) animada que ilustra a ordem de avaliação das sub-expressões. Para questões de múltiplas linhas (*multi-line questions*), o *feedback* apresenta um fluxograma animado que descreve o fluxo de controle do programa, sincronizando o destaque no fluxograma com a linha de código em execução. O sistema de Risha et al. (2021) utiliza uma tabela interativa com *scaffolding* adaptativo e feedback imediato por célula. O sistema impede o avanço do aluno em caso de erro, exigindo a correção de cada etapa para garantir o rastro correto. Já a plataforma *VPL/MCTest* [Teubl e Zampirolli 2023] retorna a nota e um *feedback* textual indicando se tem um ou mais erros ou se há linhas inválidas ou outros erros estruturais. No *My Trace Table*, o sistema de *feedback* atua diretamente nas células da tabela sem exibir a resposta correta após o aluno submeter o exercício para correção, concentrando-se em identificar a natureza do erro, quando houver. Isso é feito através de um esquema de cores que categoriza a falha em erro de valor (vermelho) e erro de tipo (amarelo), indicando se o equívoco foi na lógica ou incoerência no tipo de dado inserido em relação ao esperado. O aluno é estimulado então a tentar identificar a resposta correta, podendo inclusive pedir ajuda ao professor que media a atividade, ou podendo discutir com colegas para compreender seu erro. As principais distinções entre os sistemas analisados nesta seção estão sintetizadas na Tabela 1.

Tabela 1. Comparativo entre o *My Trace Table* e *My Trace Table Manager* e sistemas correlatos

Sistema/Aplicativo	Criação de Conteúdo	Mecanismo de Feedback	Plataforma
My Trace Table / Manager (Este	Manual (Autoria do Professor)	Diferencia erro de tipo (amarelo) de erro de valor	*Web*

trabalho)		(vermelho)	
Propostas Anteriores da UFPB [J. Santos 2023; D. Santos 2024; Silva Jr. 2025; D. Santos et al. 2025]	Conteúdo estático	Sinaliza a localização do erro com interrogações ou destaque em vermelho	Aplicativo [J. Santos 2023; Silva Jr. 2025] e *Web* [D. Santos 2024][D. Santos et al. 2025]
VPL/MCTest [Teubl e Zampirolli 2023]	Automática e Parametrizada	Nota e feedback textual (Erros estruturais)	*Web*
Trace Generator [Russell 2022]	Automática e Parametrizada	Animações (Árvore ou Fluxograma)	*Web*
Risha et al. [2021]	Automática e Parametrizada	Bloqueia o avanço por célula até que o erro seja corrigido.	*Web*

6. Resultados

Os principais resultados deste trabalho são a apresentação dos sistemas *My Trace Table* e *My Trace Table Manager*, bem como sua avaliação inicial. Estes sistemas encontram-se disponíveis online e possuem seus códigos-fonte publicados sob a licença MIT no GitHub do projeto de extensão A4S da UFPB. A estrutura está organizada em três repositórios — *front-end* do aluno[9], *front-end* do professor[10] e *back-end* integrado[11] — e todos incluem um *readme* orientando a configuração do ambiente e a execução local, visando facilitar a evolução, utilização e customização dos sistemas pela comunidade acadêmica e demais interessados.

6.1. Avaliação Inicial em Campo e Questionário Online

A avaliação inicial das ferramentas concentrou-se em verificar a aceitação dos sistemas em ambientes educacionais reais e coletar *feedbacks* para sua melhoria. Um dos ambientes em que o *My Trace Table* foi utilizado foi a Escola Técnica Estadual (ETE) Antônio Arruda de Farias, localizada na cidade de Surubim, Pernambuco (PE), em uma sessão de 50 minutos com 43 alunos, sendo 21 homens e 22 mulheres, do curso de Desenvolvimento de Sistemas integrado ao ensino médio. A observação do uso do *My Trace Table* foi registrada utilizando um Protocolo de Observação de Uso de Sistema/Aplicativo[12] previamente aprovado pelo comitê de ética da UFPB, que continha os seguintes itens a serem observados: Engajamento, Falta de Engajamento, Aprendizado, Facilidade de Uso e Dificuldade de Uso. O experimento revelou um elevado engajamento dos estudantes, percepção de aprendizado e facilidade de uso do sistema, dado que os estudantes não demonstraram dificuldades para acessar e navegar na plataforma. A escola forneceu a infraestrutura adequada para o experimento (datashow, internet, mesas compartilhadas e notebooks individuais) e o professor

[9] https://github.com/a4s-ufpb/My-Trace-Table
[10] https://github.com/a4s-ufpb/My-Trace-Table-Manager
[11] https://github.com/a4s-ufpb/My-Trace-Table-Manager-API
[12] https://drive.google.com/file/d/1-IqdUyqUrGNW4-sJVnzc9o0LHEIIAOeQ/view?usp=drive_link

sugeriu os exercícios mais adequados para cada momento. Embora cada estudante tivesse um computador para prática individual, eles se mostraram bastante engajados e sanaram dúvidas entre si. A princípio foi apresentado o contexto geral sobre a criação da plataforma, seguida de uma demonstração de como o sistema funciona resolvendo alguns exercícios exibidos com o datashow, e por fim, iniciou-se a prática individual dos alunos que foram auxiliados pelos pesquisadores, pelo professor ou pelos próprios colegas quando necessário.

Uma observação relevante identificada foi a eficácia do *feedback* visual: células vermelhas (erro de valor) ou amarelas (erro de tipo) rapidamente se transformavam em verdes (acerto) durante a sessão usando o sistema. Esse fato comprova que os estudantes estavam identificando e corrigindo seus equívocos de forma ativa. Os alunos demonstraram em muitos momentos expressões corporais e verbais de "estalos" de compreensão, com exclamações como "Aaahhh, entendi", evidenciando que o *feedback* rápido é funcional para orientar a correção. Estas constatações corroboram o princípio do reforço imediato, essencial para o arranjo de contingências eficazes, conforme postulado por Skinner (1972). Resultados semelhantes foram percebidos em experiência no Campus IV da UFPB em Rio Tinto, onde a professora Ayla Rebouças utilizou os sistemas em turmas de Introdução a Programação com cerca de 40 alunos em laboratórios de informática da instituição, reportando que os estudantes também se mostraram bem engajados nas atividades e demonstravam claramente que a ferramenta estava ajudando em sua compreensão de código. Adicionalmente, percebeu-se que a implementação do *My Trace Table Manager* provê o suporte necessário para a autoria pedagógica (Pontes 2007) ao permitir que os professores possam cadastrar temas e exercícios e gerenciá-los conforme as preferências de códigos e configurações desejadas. Os sistemas alinham-se à ideia de que é importante automatizar a correção de exercícios para apoiar o aprendizado [Russell 2022; Teubl e Zampirolli 2023] e amenizam o problema da correção manual de atividades de programação [D. Santos 2024], permitindo que o professor dedique tempo à mentoria individualizada quando necessário.

6.2. Análise dos Dados Obtidos Através do Questionário Online

As percepções observadas em campo foram corroboradas pelos resultados da avaliação inicial feita através de um questionário online do Google Forms[13]. A análise dos dados obtidos do questionário é parte da etapa de avaliação da metodologia DSR [Pimentel et al. 2019]. É importante destacar que tanto o "Protocolo de Observação de Uso de Sistema/Aplicativo", quanto o formulário de avaliação do *My Trace Table* foram previamente aprovados pelo Comitê de Ética, sob o CAAE 87795825.6.0000.5188 e estes continham links para os TCLEs e TALE. A coleta de dados através do questionário correspondente a esta avaliação inicial, resultou em apenas 12 respostas, o que não permite generalizar seus resultados, embora traga indícios de que a ferramenta tem sido útil e bem aceita. As tendências iniciais observadas para as métricas de satisfação do usuário, facilidade de uso e potencial percebido para a aprendizagem de programação

[13] https://forms.gle/FqLfGcMeYKNybho3A

são ilustradas na Figuras 6, que exibe graficamente as respostas obtidas relacionadas a essas métricas.

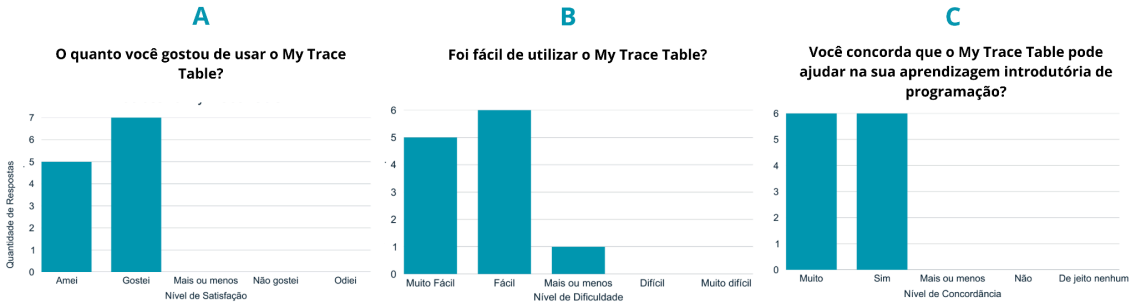

Figura 6. Percepção dos participantes quanto à satisfação (A), facilidade de uso (B) e contribuição para a aprendizagem de programação (C) do My Trace Table. Fonte: Autoria própria.

A satisfação do usuário utilizando o sistema foi muito positiva. Conforme ilustra a Figura 6 (A), a soma das respostas "Amei" (5 respostas) e "Gostei" (7 respostas) totaliza 12 respostas. Isso significa que 100% dos participantes classificaram positivamente a experiência de uso. Especificamente, 41,7% ("Amei") demonstraram alto nível de satisfação, e 58,3% ("Gostei") indicaram satisfação, não havendo respostas nas categorias neutras ou negativas ("Mais ou menos," "Não gostei," "Odiei"). A facilidade de uso também foi um ponto forte. A Figura 6 (B) indica que 50% dos respondentes classificaram o *My Trace Table* como “Fácil” de usar e 41,7% classificaram como “Muito Fácil”. Isso indica que a grande maioria dos participantes que responderam confirmam facilidade de uso do sistema (11 das 12 respostas), visto que apenas 8,3% classificou o sistema na categoria neutra “Mais ou menos”, o que corresponde a 1 resposta, e não houve respostas para as categorias “Difícil” ou “Muito difícil”. O potencial do sistema em auxiliar na aprendizagem introdutória de programação foi unanimemente reconhecido. A Figura 6 (C) demonstra que 100% dos participantes concordam que o sistema pode ajudar no aprendizado de programação, sendo 50% das respostas registradas como “Sim”, 50% como “Muito” e nenhuma resposta na categoria neutra “Mais ou menos” ou nas negativas “Não” e “De jeito nenhum”.

A avaliação inicial positiva referente ao *My Trace Table* demonstra, portanto, o valor de se aplicar o *feedback* automático ao ensino de programação. Ao transformar a metodologia de teste de mesa em uma ferramenta digital, o projeto permite que o foco permaneça na melhoria da habilidade de compreensão do código por parte dos estudantes. Com a possibilidade do professor cadastrar variados temas e exercícios com as especificações desejadas através do *My Trace Table Manager*, expande-se o leque de conteúdos que podem ser abordados, o que permite aos alunos desenvolver e aperfeiçoar o conhecimento em mais assuntos.

7. Conclusões e Trabalhos Futuros

Pode-se concluir que o objetivo de apresentar o *My Trace Table* e o *My Trace Table Manager* deste trabalho foi alcançado. Estes sistemas oferecem ambientes estruturados

para o cadastro e a prática de exercícios de teste de mesa com *feedback* rápido para auxiliar estudantes iniciantes em programação na compreensão de código, fornecendo autoria pedagógica ao professor e facilitando o processo de correção de múltiplos exercícios.

Os resultados iniciais da avaliação, embora limitados a 12 participantes, indicam que o sistema é altamente satisfatório (100% de aprovação) e fácil de usar (91,7% de facilidade percebida). Mais importante, a avaliação demonstra indícios do potencial pedagógico da ferramenta, com 100% dos respondentes concordando que o *My Trace Table* pode auxiliar na sua aprendizagem introdutória de programação, que é o objetivo que levou ao desenvolvimento desses sistemas. Tais achados, somados à observação de campo, sugerem que a estratégia de fornecer *feedback* rápido e visualmente distinto (acerto, erro de valor e erro de tipo) é eficaz para promover a correção ativa dos erros cometidos pelos estudantes, cumprindo a missão de auxiliar na compreensão de código.

Como trabalhos futuros, algumas frentes de melhoria são a realização de mais avaliações e testes, principalmente com o *My Trace Table Manager*. É crucial ampliar a cobertura de testes automatizados para evoluir a robustez a longo prazo dos sistemas. Outras melhorias devem ser priorizadas para resolver algumas limitações identificadas, como a possibilidade de substituir imagens de código por blocos de texto para otimizar o carregamento no navegador e persistir as respostas dos alunos no banco de dados para permitir ao professor melhor acompanhar os erros dos estudantes, realizando intervenções em sala de aula com base nessas observações. Com base nas análises das respostas, os professores poderiam facilmente identificar dificuldades acerca de determinados temas. Esse processo de identificação de dificuldades poderia ser enriquecido através da implementação de funcionalidades estatísticas para exibir gráficos de desempenho para o professor. Diante dos resultados iniciais promissores, busca-se ampliar suficientemente a coleta de respostas para prover melhorias aos sistemas e compreender os impactos que eles causam.

Agradecimentos

Os autores expressam gratidão à UFPB pelo suporte institucional. Um agradecimento especial é dedicado ao estudante Ronyelison Abreu, do curso de Licenciatura em Ciência da Computação da UFPB, pelo empenho e apoio nas etapas iniciais de desenvolvimento do My Trace Table e também à Professora Vanessa Dantas da UFPB, cujo incentivo e prática pedagógica exemplar — marcada pelo uso intensivo de testes de mesa em sala de aula — serviram de inspiração para o desenvolvimento desses sistemas. Agradecemos também o apoio da escola ETE Antônio Arruda de Farias e do professor Everton Figueiredo durante a avaliação com os estudantes desta escola.

Uso de Inteligência Artificial

Para colaborar com a escrita deste artigo foram utilizadas tecnologias de Inteligência Artificial (IA) Generativa. O Google Gemini foi a ferramenta de IA utilizada para deixar mais formal o texto de alguns trechos deste trabalho. Além disso, essa ferramenta também foi útil para traduzir pequenas partes dos textos das fontes utilizadas que foram escritas em língua estrangeira.

Referências

Gamma, Erich et al. (1995). Design patterns: elements of reusable object-oriented software, Pearson.

Hassan, M. and Zilles, C. (2023). On Students' Usage of Tracing for Understanding Code. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, Association for Computing Machinery, New York, NY, USA, p. 129–136. p. 129–136. https://doi.org/10.1145/3545945.3569741

Lopez, M., Whalley, J., Robbins, P. and Lister, R. (2008). Relationships between reading, tracing and writing skills in introductory programming. In *Proceedings of the Fourth international Workshop on Computing Education Research (ICER '08)*, Association for Computing Machinery, New York, NY, USA, p. 101–112. https://doi.org/10.1145/1404520.1404531

Peffers, Ken et al. (2007). A Design Science Research Methodology for Information Systems Research. In *Journal of Management Information Systems*, v. 24, n. 3, p.45–77. https://doi.org/10.2753/MIS0742-1222240302

Pimentel, M., Filippo, D., Santoro, Flávia M. (2019). Design Science Research: fazendo pesquisas científicas rigorosas atreladas ao desenvolvimento de artefatos computacionais projetados para a educação. In *Metodologia de Pesquisa em Informática na Educação: Concepção da Pesquisa*. Porto Alegre: SBC, p. 5–29.

Pontes, Rosana Aparecida Ferreira. (2007). A construção da autoria pedagógica na formação de educadores. *Dissertação (Mestrado em Educação e Formação)*, Universidade Católica de Santos, Santos.

Richardson, L. et al. (2013). RESTful Web APIs: Services for a Changing World, O'Reilly Media.

Risha, Z., Barria-Pineda, J., Akhuseyinoglu, K. and Brusilovsky, P. (2021). Stepwise Help and Scaffolding for Java Code Tracing Problems With an Interactive Trace Table. In *Proceedings of the 21st Koli Calling International Conference on Computing Education Research (Koli Calling '21),* Association for Computing Machinery, New York, NY, USA, Article 27, p. 1–10. https://doi.org/10.1145/3488042.3490508

Russell, S. (2022). Automated Code Tracing Exercises for CS1. In *Proceedings of the 6th Conference on Computing Education Practice (CEP '22)*, Association for Computing Machinery, New York, NY, USA, p. 13–16. https://doi.org/10.1145/3498343.3498347

Santos, D. T. Q. (2024). My Trace Table: Um Sistema Web para Apoiar no Ensino Introdutório de Programação. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação). Universidade Federal da Paraíba, Rio Tinto.

Santos, D. T. Q., Abreu, R. O., Rebouças, A. D., Dantas, V. F. (2025) My Trace Table: Um Sistema Web para Apoiar no Ensino Introdutório de Programação. IX Seminário Educacional de Práticas Educativas (SECAMPO 2025).

Santos, J. S. (2023). Meu teste de mesa: um recurso educacional digital para apoiar a aprendizagem de operadores em programação. Trabalho de Conclusão de Curso (Licenciatura em Ciência da Computação). Universidade Federal da Paraíba, Rio Tinto.

Silva Júnior, P. G. (2025). My Trace Table Mobile: Uma ferramenta para apoiar o ensino introdutório de programação através de dispositivos móveis. Trabalho de Conclusão de Curso (Licenciatura em Ciência da Computação). Universidade Federal da Paraíba, Rio Tinto.

Simon, Hebert A. (1969). The Sciences of the Artificial, MIT Press.

Skinner, B. F. (1972). Tecnologia do Ensino, EPU.

Teubl, F. and Zampirolli, F. (2023). Automated Correction for Trace Tables in a CS1 Course. In *Simpósio Brasileiro De Informática Na Educação (SBIE), 34., Anais [...]*, Sociedade Brasileira de Computação, Porto Alegre, p. 1546-1556. https://doi.org/10.5753/sbie.2023.233468.

Vaishnavi, Vijay K., Kuechler, William. (2015). Design Science Research Methods and Patterns: Innovating Information and Communication Technology, CRC Press.