Inovação no Ensino de Lógica: Desenvolvimento e Avaliação de uma Ferramenta para Construção de Tabelas Verdade

Guilherme Moura[1], **Eduarda Munhoz Sifuentes**[1], **Ildevana Poltronieri**[1], **Alice Finger**[1]

[1]Universidade Federal do Pampa (UNIPAMPA), *Campus* Alegrete
Av. Tiarajú, 810, Ibirapuitã – Alegrete, RS – Brasil

guilhermemoura.aluno,eduardasifuentes.aluno

ildevanarodrigues,alicefinger@unipampa.edu.br

Abstract. *This paper introduces the* Construir Tabelas*, integrated into the Trueble learning environment, which teaches propositional logic by guiding students through the manual construction of truth tables with immediate cell-level feedback. The implementation features a custom logical keyboard, automatic sub-formula detection, and an interactive validation workflow. Usability was assessed through a heuristic inspection involving four experts, who identified 42 issues—the most critical concerning error prevention, status visibility, and visual consistency. We outline a set of prioritized fixes and, as future work, propose user studies with students, gamification features, teacher dashboards, and accessibility enhancements to broaden the tool's educational reach.*

Resumo. *Este artigo apresenta a ferramenta* Construir Tabelas*, integrada ao ambiente Trueble, voltada ao ensino de lógica proposicional por meio da construção manual de tabelas verdade com feedback. A implementação inclui teclado lógico customizado, detecção de subfórmulas e fluxo interativo de validação. A usabilidade foi avaliada por meio de inspeção heurística com quatro especialistas. Foram identificados 42 problemas, sendo os mais críticos relacionados à prevenção de erros, visibilidade de status e consistência visual. Propomos correções prioritárias e, como trabalhos futuros, sugerimos testes com estudantes, recursos de gamificação, relatórios para docentes e funcionalidades de acessibilidade para ampliar o uso da ferramenta em contextos educacionais.*

1. Introdução

O ensino de lógica matemática ocupa um papel central na formação de estudantes de Computação, sendo essencial para o desenvolvimento do pensamento algorítmico e da capacidade de resolver problemas complexos [Silva et al. 2006]. Entre os conceitos fundamentais dessa área, destacam-se as tabelas verdade, por sua ampla aplicabilidade no contexto da programação, ao permitirem a verificação de expressões booleanas e o entendimento de estruturas condicionais recorrentes em linguagens como *if, while e for* [Huth and Ryan 2004]. Dominar a construção e interpretação de tabelas verdade possibilita aos estudantes validar a correção de suas expressões lógicas e produzir códigos mais consistentes e robustos [Sebesta 2018].

O estudo das tabelas verdade, embora essencial, costuma ser uma fonte de dificuldades para os estudantes, especialmente em cursos de Computação. Caracteriza-se por

DOI: 10.5753/sbie.2025.12780

um elevado nível de abstração que frequentemente torna sua assimilação extremamente desafiadora [Valente 2020]. Essa dificuldade não é pontual, mas recorrente, e tem sido apontada como um dos principais fatores que comprometem o desempenho acadêmico e fomentam a evasão nos cursos da área. Na Universidade Federal Rural de Pernambuco, por exemplo, a ausência de uma base sólida em lógica foi identificada como causa central do abandono no curso de Licenciatura em Computação [Vasconcelos and Andrade 2018]. De forma ainda mais alarmante, entre 2001 e 2008, a Universidade Federal de Mato Grosso registrou taxas médias de reprovação superiores a 47% na disciplina de Lógica Matemática, afetando gravemente o progresso dos estudantes já nos primeiros semestres [Prietch and Pazeto 2010]. Esses dados evidenciam a urgência de repensar estratégias pedagógicas que tornem o ensino de lógica mais acessível, engajador e eficaz.

Entre as diferentes estratégias de ensino voltadas ao apoio na aprendizagem, destacam-se as tecnologias educacionais, como softwares que oferecem suporte ao ensino de lógica [Dias and Finger 2020]. Há, portanto, uma demanda crescente por soluções tecnológicas eficazes, capazes de melhorar o desempenho discente e reduzir os índices de evasão nos cursos de Computação [Lugli 2018]. Nesse contexto, Dias e Finger (2020) realizaram uma pesquisa na Google Play Store e identificaram 19 aplicativos relacionados à Lógica Matemática, dos quais apenas oito abordavam a construção de tabelas verdade, uma habilidade relevante, porém limitada frente à complexidade do raciocínio lógico exigido na formação em Computação. Esses dados evidenciam que, embora já existam iniciativas voltadas ao ensino de lógica, ainda persistem lacunas significativas no desenvolvimento de ferramentas educacionais mais abrangentes e alinhadas às necessidades específicas dos estudantes da área.

Diante dessas lacunas, este trabalho apresenta o Trueble, uma aplicação educacional voltada ao ensino de lógica proposicional por meio da construção manual e interativa de tabelas verdade. Diferentemente das soluções existentes, o Trueble adota uma abordagem centrada na atividade do estudante, permitindo que ele elabore cada etapa da tabela verdade aprimorando progressivamente sua autonomia cognitiva. Trata-se, portanto, de uma proposta inovadora que busca aliar interatividade, orientação pedagógica e usabilidade em um único ambiente digital. Além do desenvolvimento da ferramenta "Construir Tabelas", esta pesquisa inclui uma avaliação heurística com base nos princípios de usabilidade de Nielsen [Nielsen 1994], com o objetivo de analisar a interface da ferramenta, identificar suas limitações e propor melhorias que ampliem sua eficácia pedagógica.

O restante do artigo está organizado da seguinte forma: na Seção 3, revisam-se trabalhos relacionados e aplicações existentes; na Seção 2, apresenta-se a fundamentação teórica sobre lógica matemática e tabelas verdade; na Seção 4, detalha-se o desenvolvimento da ferramenta *Construir Tabelas*; na Seção 5, descrevem-se o método e os resultados da avaliação heurística de Nielsen; e, por fim, na Seção 6, apresentam-se as considerações finais e trabalhos futuros.

2. Fundamentação Teórica

A lógica matemática é uma disciplina essencial para o desenvolvimento do raciocínio formal e para a construção de argumentos válidos. Conforme destaca [Avigad 2022], essa área examina as limitações e capacidades dos sistemas formais, constituindo-se como um dos pilares da computação moderna, especialmente no que diz respeito à modelagem

de algoritmos, linguagens formais e verificação de propriedades lógicas. Entre os instrumentos fundamentais para essa compreensão estão as tabelas verdade, que permitem representar todas as combinações possíveis dos valores de variáveis proposicionais e seus respectivos resultados lógicos.

Segundo [Huth and Ryan 2004], essas tabelas são essenciais para verificar a validade de argumentos, definir formalmente operadores e compreender as interações entre proposições. No contexto computacional, [Lewis and Papadimitriou 1997] afirmam que a capacidade de construir e interpretar tabelas verdade capacita os estudantes a projetar e depurar algoritmos com maior eficiência e correção. Do ponto de vista didático, o uso sistemático dessa ferramenta facilita o raciocínio sequencial e aproxima a linguagem natural da linguagem formal, uma competência indispensável para a formação em Computação.

Entre os conceitos centrais da lógica matemática, destacam-se as proposições e os operadores lógicos, que permitem compor expressões e avaliar seus valores de verdade. Os principais operadores incluem: a conjunção ($\wedge$), que representa o "e" lógico e é verdadeira apenas quando ambas as proposições são verdadeiras; a disjunção ($\vee$), que corresponde ao "ou" lógico e é verdadeira se ao menos uma das proposições for verdadeira, sendo falsa apenas quando ambas forem falsas; a negação ($\neg$), que inverte o valor lógico de uma proposição, se p é verdadeira, então $\neg p$ é falsa; a implicação ($\rightarrow$), que expressa a relação "se... então" e é falsa somente quando o antecedente p é verdadeiro e o consequente q é falso, sendo verdadeira nos demais casos; e a bicondicional ($\leftrightarrow$), que representa o "se e somente se" e é verdadeira quando ambas as proposições compartilham o mesmo valor lógico. Esses operadores são a base para a construção de expressões complexas e fundamentam a análise formal por meio das tabelas verdade.

Dada a complexidade dos operadores lógicos e a abstração da lógica matemática, as interfaces digitais têm papel fundamental no apoio à aprendizagem. Diante deste cenário, destacam-se as heurísticas de usabilidade de Nielsen [Nielsen 2005], que visam reduzir a carga cognitiva e tornar a interação mais intuitiva. A avaliação heurística, baseada nesses princípios, consiste em uma análise conduzida por especialistas que identificam problemas de usabilidade e os classificam por grau de severidade. Aplicada a plataformas educacionais, essa técnica contribui para tornar conteúdos abstratos, como as tabelas verdade, mais acessíveis e engajadores, especialmente para iniciantes em lógica. Entre os princípios avaliados, incluem-se visibilidade do status do sistema (H1), correspondência com o mundo real (H2), controle do usuário (H3), consistência (H4), prevenção de erros (H5), reconhecimento em vez de memorização (H6), flexibilidade (H7), design minimalista (H8), recuperação de erros (H9) e ajuda e documentação (H10). A severidade dos problemas é classificada em uma escala de 0 a 4, onde:

- **0**: Não é um problema.
- **1**: Problema leve, mas não afeta a usabilidade.
- **2**: Problema moderado que afeta a experiência de usuário, mas não impede o uso.
- **3**: Problema sério que afeta significativamente a usabilidade.
- **4**: Problema crítico que impede o uso da ferramenta.

3. Trabalhos Relacionados

Esta seção revisa trabalhos relacionados ao ensino de lógica proposicional, com foco em ferramentas educacionais destinadas à construção e compreensão de tabelas verdade.

A pesquisa foi conduzida de forma *ad hoc*, abrangendo artigos acadêmicos e fontes de literatura cinzenta, como relatórios técnicos e documentações de software, sem seguir um protocolo sistemático. Foram utilizados os termos "propositional logic", "truth tables", "education", "tools" e "software" como palavras-chave, considerando apenas materiais publicados nos últimos seis anos, em inglês ou português, que abordassem explicitamente soluções para o ensino ou prática de tabelas verdade. A seleção priorizou a relevância educacional e a conexão com os objetivos do presente trabalho.

O estudo conduzido por [De Troyer et al. 2019], investigou as baixas taxas de aprovação em lógica proposicional e desenvolveram o jogo educacional "TrueBiters" para prática de tabelas verdade de forma interativa e competitiva em smartphones. Inspirado em jogos de cartas, o aplicativo evoluiu por meio de feedback dos usuários e provou-se eficaz para engajar estudantes e reforçar a compreensão dos conceitos lógicos. Avaliações mostraram que "TrueBiters", quando integrado ao curso, pode substituir atividades tradicionais, promovendo uma aprendizagem mais sólida.

Além das abordagens gamificadas, diversas ferramentas digitais têm sido desenvolvidas com o intuito de auxiliar estudantes na visualização e verificação de expressões lógicas por meio da geração automática de tabelas verdade. O Truth Table Generator, da Universidade de Stanford[1], por exemplo, permite que o usuário insira expressões lógicas e obtenha instantaneamente a tabela correspondente, facilitando o entendimento das operações proposicionais sem exigir sua construção manual.

De forma semelhante, o Truth Table Tool[2] oferece uma interface intuitiva para a visualização prática do comportamento lógico de proposições. Na mesma linha, o Truth Table Calculator, disponibilizado pelo portal eMathHelp[3], apresenta os resultados das expressões de lógica proposicional de maneira automática, servindo como ferramenta de verificação rápida. Em comum, essas ferramentas eliminam a necessidade de elaboração manual das tabelas, promovendo um aprendizado mais visual, ágil e acessível.

As ferramentas analisadas permitem a geração automática de tabelas verdade, mas não proporcionam um ambiente que estimule a construção ativa do conhecimento pelo estudante. Nestas soluções, o usuário apenas insere uma expressão lógica e visualiza o resultado final, sem acesso às etapas intermediárias do raciocínio. Em contraste, o Trueble propõe uma ruptura metodológica significativa, indo além de uma simples "calculadora de lógica proposicional". A plataforma promove um aprendizado estruturado e interativo, simulando o processo manual de preenchimento da tabela verdade, linha por linha, e permitindo que o aluno explore cada operação lógica de forma individual.

4. Desenvolvimento da Ferramenta *Construir Tabelas*

A ferramenta *Construir Tabelas*, integrada à aplicação Trueble, foi desenvolvida com o objetivo de apoiar o ensino e a aprendizagem de tabelas verdade de maneira interativa, aliando usabilidade, feedback imediato e reforço pedagógico. Trata-se de um dos módulos do ambiente educacional, projetado para estimular o raciocínio lógico por meio de

[1] https://web.stanford.edu/class/cs103/tools/truth-table-tool/
[2] https://truth-table.com
[3] https://www.emathhelp.net/en/calculators/discrete-mathematics/truth-table-calculator/

atividades manipulativas e orientadas por regras formais. A concepção inicial e os resultados preliminares da ferramenta foram apresentados no XXXV Simpósio Brasileiro de Informática na Educação (SBIE 2024) [Moura et al. 2024], demonstrando seu potencial para o ensino de lógica proposicional no contexto digital.

Para compreender sua estrutura atual, é essencial contextualizar o processo de desenvolvimento adotado. A construção da ferramenta teve início com a definição do escopo e a prototipação da interface, seguida por uma fase de testes preliminares e coleta de feedback com usuários reais. As melhorias sugeridas nessa etapa foram incorporadas na implementação funcional do módulo. Posteriormente, a aplicação passou por testes manuais e uma avaliação heurística, cujos resultados serviram de base para ajustes técnicos e pedagógicos que orientaram as iterações seguintes do sistema.

As próximas subseções detalham os principais aspectos técnicos e pedagógicos de sua implementação. Inicialmente, é apresentada a arquitetura do sistema. Em seguida, descrevem-se as estratégias para validar a inserção de fórmulas lógicas, automatizar a geração da tabela verdade e oferecer mecanismos de verificação e feedback ao usuário.

4.1. Estrutura Técnica

A aplicação foi desenvolvida em React Native com TypeScript, garantindo compatibilidade com dispositivos iOS e Android a partir de uma base de código unificada. Caso necessário, pode ser adaptada para ambientes web utilizando React. A navegação entre telas é gerenciada por meio do *Expo Router*; o controle de estado e os efeitos colaterais são tratados com *React Hooks*; e o respeito às áreas sensíveis da interface — como entalhes e barras de navegação — é assegurado pelo *React Native Safe Area Context*.

A estrutura da ferramenta "Construir" organiza-se em quatro componentes. O arquivo `Construir.tsx` representa a página inicial, na qual o aluno insere a fórmula lógica a ser analisada. O componente `Keyboard.tsx` oferece um teclado customizado que permite a inserção de variáveis proposicionais (P, Q, R, S, T), conectivos lógicos ($\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$), parênteses e demais símbolos formais ($\vdash$, $\mid$ $=$, $\equiv$), além de realizar a validação sintática das expressões. O componente `TruthTable.tsx` é responsável por extrair variáveis, identificar subfórmulas e calcular os valores de cada célula da tabela verdade. Por fim, o arquivo `tabela-verdade.tsx` corresponde à tela principal de interação, onde o aluno preenche a tabela, verifica a correção das células e, conforme o tipo de exercício, realiza julgamentos lógicos (como validade, equivalência ou consequência) ou classificações (tais como tautologia, contradição, contingência e satisfazibilidade).

4.2. Personalização do Teclado

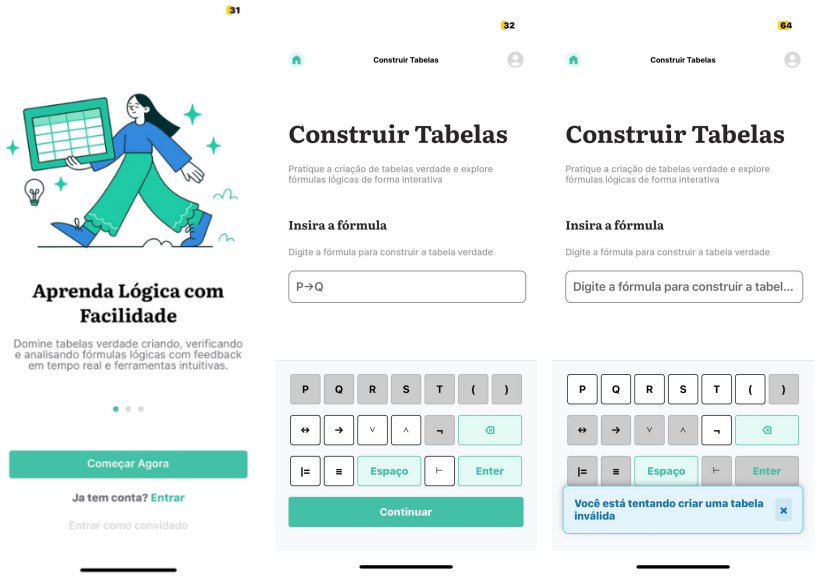
Na aplicação Trueble, ao acessar a ferramenta *Construir Tabelas*, o usuário insere a fórmula lógica para a qual deseja construir a tabela verdade. O teclado personalizado segue regras de entrada sintática, conforme resumido na Tabela 1. A primeira coluna da tabela indica o último caractere inserido, enquanto a segunda apresenta os próximos caracteres válidos disponíveis ao usuário, garantindo que apenas expressões logicamente bem formadas possam ser construídas.

Na Figura 1(b) pode ser visualizada a implementação do teclado personalizado, cujas regras foram apresentadas na Tabela 1. Ao inserir a fórmula, é possível perceber que o teclado se adapta ao que consta na caixa de texto disponível para digitação, em que

Tabela 1. Regras aplicadas ao teclado personalizado.

Último caractere	**Próximos permitidos**
Início	(, ¬, variável
¬	(, variável
variável	$\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\vdash$, $\models$, $\equiv$,) (se existir '(' aberto)
(	(, ¬, variável
$\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\vdash$, $\models$, $\equiv$	(, ¬, variável
)	$\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\vdash$, $\models$, $\equiv$
Obs. 1: teclas de apagar caractere e espaço são sempre permitidos.	
Obs. 2: $\vdash$, $\models$ *e* $\equiv$ *não podem se repetir.*	

as teclas desativadas permanecem na cor cinza, enfatizando que não podem ser utilizadas naquele momento. Se o usuário mesmo assim tentar inserir uma tecla inválida, então ele é notificado e surge uma mensagem indicando que a ação não é permitida, o que pode ser visualizado na Figura 1(c).



(a) Tela inicial da aplicação (b) Tela para inserir fórmula (c) Resposta de fórmula inválida

Figura 1. Telas iniciais da ferramenta *Construir Tabelas*.

4.3. Automatização da Tabela

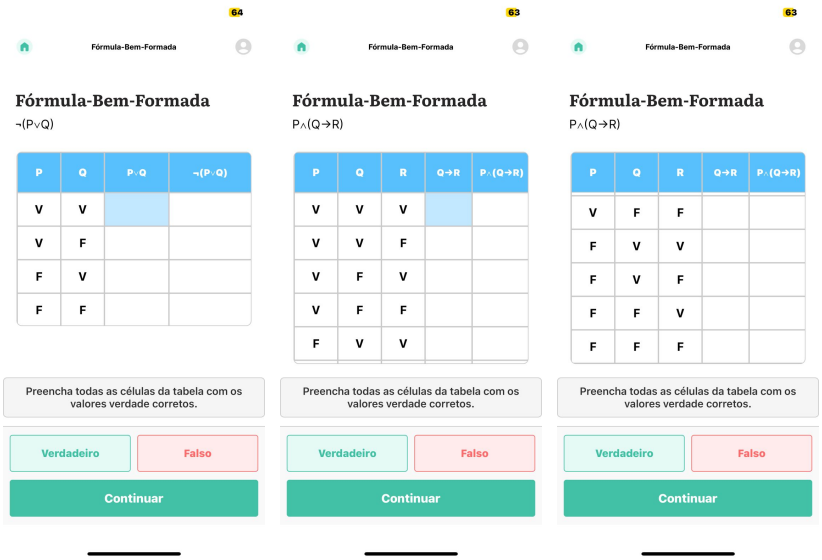
A partir da fórmula inserida pelo usuário, a ferramenta inicia a geração da tabela verdade com a identificação das variáveis proposicionais presentes na expressão. Com base nessa identificação, são construídas todas as combinações possíveis de valores verdade, totalizando 2^n linhas, onde n é o número de variáveis. Por exemplo, na Figura 2(a), o usuário utilizou a fórmula $\neg(P \vee Q)$, que envolve duas variáveis: P e Q. Assim, a tabela verdade gerada contempla as quatro possíveis combinações de valores (i.e., $2^2 = 4$).

Em seguida, a ferramenta realiza a detecção de subfórmulas. Esse processo começa pela análise das expressões mais internas entre parênteses, evitando repetições. São identificadas negações simples (como $\neg P$) e compostas (por exemplo, $\neg(P \vee Q)$). Depois, são localizadas, em cada nível livre de parênteses, as operações binárias seguindo a ordem de precedência: conjunção ($\wedge$), disjunção ($\vee$), condicional ($\rightarrow$) e bicondicional ($\leftrightarrow$). Ao

final, garante-se que a própria fórmula principal esteja incluída na lista de subfórmulas, caso ainda não tenha sido detectada. Um exemplo visual desse processo é apresentado na Figura 2(b), onde o usuário seleciona a fórmula $P \wedge (Q \rightarrow R)$. Nesse caso, a subfórmula $(Q \rightarrow R)$ é processada primeiro devido aos parênteses, e posteriormente reutilizada na construção da fórmula principal.

Para otimizar a avaliação das expressões, é utilizada uma estratégia de cache interno. Cada subfórmula é avaliada para todas as linhas da tabela por meio de uma rotina recursiva que armazena os resultados em um mapa interno. Essa abordagem evita cálculos redundantes, especialmente em fórmulas com estrutura aninhada. Na Figura 2(a), observa-se a reutilização da subfórmula $P \vee Q$, sem recriação da estrutura, aproveitando-se da equivalência lógica.

Por fim, a apresentação da tabela utiliza um layout com colunas e cabeçalhos fixos. As colunas referentes às variáveis permanecem fixas à esquerda, enquanto o cabeçalho da tabela se mantém visível no topo. Essa configuração facilita a leitura e o preenchimento, mesmo quando o conteúdo exige rolagem. Um exemplo dessa funcionalidade pode ser observado nas Figuras 2(b) e 2(c), que mostram uma tabela com oito linhas — das quais apenas cinco são inicialmente visíveis. Ao rolar a tela, como ilustrado na Figura 2(c), as subfórmulas permanecem fixas, permitindo ao usuário acompanhar com clareza a estrutura lógica ao longo de toda a tabela.



(a) Exemplo de Fórmula-Bem-Formada

(b) Tabela verdade com oito linhas (usar scroll)

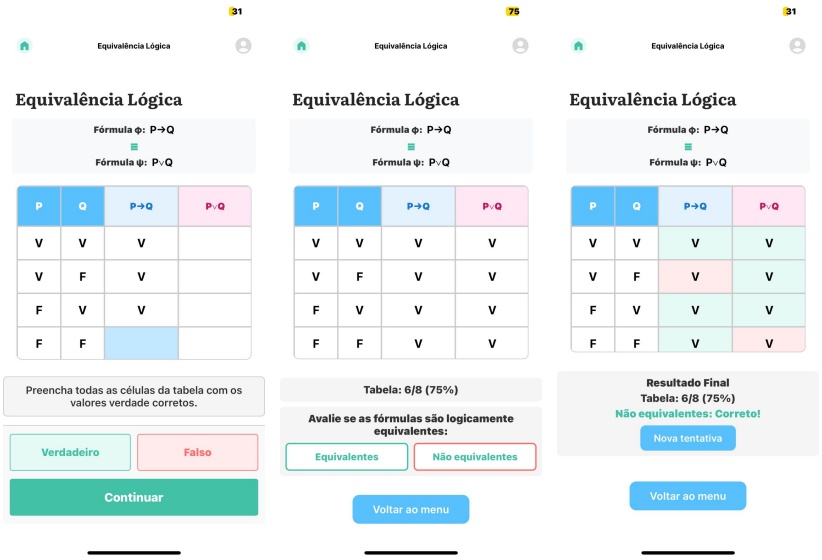
(c) Primeira linha fixa e scroll sendo rolado para baixo

Figura 2. Tabela gerada automaticamente pela ferramenta.

Após concluídas as etapas de geração da tabela verdade, o usuário é direcionado para uma tela contendo a tabela ainda em branco, que deve ser preenchida manualmente. O processo segue duas etapas principais. A primeira é o preenchimento manual, no qual, ao tocar em uma célula, abre-se um teclado simplificado com as opções “Verdadeiro” ou “Falso”. O valor escolhido é armazenado e, automaticamente, o foco avança para a próxima célula, destacada em azul. Apenas após o preenchimento de todas as células, o

botão “Continuar” é habilitado. Essa etapa pode ser visualizada na Figura 3(a).

Na segunda etapa, ocorre a verificação e classificação. Para exercícios do tipo argumento, equivalência ou consequência lógica, é exibida uma interface de classificação, com as opções “Válido/Inválido”, “Equivalentes/Não equivalentes” ou “É consequência/Não é consequência”, respectivamente. Ao realizar sua escolha, o sistema compara a resposta com os cálculos internos e destaca as células da tabela: as corretas recebem fundo verde, enquanto as incorretas são marcadas com fundo vermelho. Também é exibido se a classificação geral realizada pelo usuário foi correta. Um exemplo desse processo é apresentado na Figura 3, que mostra a resolução de um exercício de equivalência lógica.



(a) Tabela de Equivalência Lógica (b) Classificação da Equivalência Lógica (c) Verificação e Classificação

Figura 3. Telas de Equivalência Lógica.

Já para fórmulas bem-formadas, como ilustrado na Figura 4, após o preenchimento das células, solicita-se ao usuário que classifique a fórmula como “Tautologia”, “Contradição” ou “Contingência” (Figura 4(a)). Em seguida, é necessário selecionar uma opção de satisfazibilidade entre “Válida”, “Satisfazível”, “Falsificável” ou “Insatisfazível” (Figura 4(b)). Somente após essas escolhas, o sistema apresenta o resultado final, indicando as células corretas (em verde), as incorretas (em vermelho) e fornecendo o feedback correspondente à classificação e à satisfazibilidade (Figura 4(c)).

Por fim, destaca-se que a ferramenta impõe um limite de cinco variáveis proposicionais por fórmula, uma decisão motivada pela complexidade crescente da tabela verdade, cujo número de linhas aumenta exponencialmente (2^n). Esse cuidado visa preservar a clareza visual e a navegabilidade, especialmente em dispositivos móveis. Para mitigar esse desafio, o sistema oferece suporte à rotação de tela, permitindo ao usuário alternar entre os modos retrato e paisagem conforme sua preferência. Além disso, o cabeçalho e as colunas de variáveis permanecem fixos, o que contribui para manter o contexto da fórmula durante o preenchimento e a análise. Apesar das soluções adotadas para ampliar a usabilidade, a restrição quanto ao número de variáveis foi mantida como uma decisão de

(a) Classificação de Fórmula-Bem-Formada (b) Classificação da satisfazibilidade (c) Verificação final (células + classificação)

Figura 4. Telas da opção Fórmula-Bem-Formada da ferramenta.

projeto estratégica, visando assegurar uma experiência estável, acessível e alinhada aos objetivos pedagógicos da ferramenta.

5. Avaliação Heurística

Com o intuito de inspecionar e avaliar a usabilidade da ferramenta Trueble, foi realizada uma Avaliação Heurística com base nas "10 Heurísticas de Usabilidade" propostas por Nielsen [Nielsen 1994]. O objetivo foi identificar problemas de interface e interação, funcionando como um diagnóstico preliminar da qualidade da experiência do usuário antes da aplicação de testes em larga escala, além de orientar possíveis melhorias no design da aplicação.

A avaliação ocorreu entre os meses de maio e junho de 2025, contando com quatro voluntários convidados, todos com formação em Computação e familiaridade prévia com princípios de usabilidade. Como critério de seleção, os participantes deveriam ter cursado a disciplina "Interação Humano-Computador" ou possuir experiência prévia com esse tipo de avaliação. O convite foi enviado por e-mail e, após a confirmação, cada avaliador recebeu um tutorial sobre a ferramenta, um formulário de Termo de Consentimento Livre e Esclarecido, além de uma planilha para o registro dos problemas encontrados e do grau de severidade atribuído.

Para a execução da avaliação, foram fornecidas instruções sobre instalação, acesso e uso da ferramenta, por meio de um guia online. Após a instalação do Trueble em seus dispositivos (via Expo Go para iOS ou arquivo `.aab` para Android), os participantes acessaram a ferramenta como convidados e exploraram livremente a funcionalidade *Construir Tabelas*, utilizando tanto os exemplos sugeridos quanto fórmulas próprias nas opções Fórmula-Bem-Formada, Argumento, Consequência Lógica e Equivalência Lógica. Durante essa exploração, registraram na planilha os problemas de usabilidade identificados,

com suas respectivas descrições e graus de severidade. Todos os documentos mencionados estão disponíveis em `https://zenodo.org/records/16972051`.

Com relação ao perfil dos participantes, todos os quatro avaliadores são egressos do curso de Engenharia de Software. Três relataram ter conhecimento mediano sobre Lógica Matemática, adquirido durante a graduação, enquanto um indicou possuir conhecimento avançado na área. Todos afirmaram ter familiaridade com aplicações educacionais, embora as utilizem apenas ocasionalmente. Quanto à faixa etária, três avaliadores têm mais de 27 anos e um está entre 24 e 26 anos.

5.1. Análise dos Resultados

A partir da análise dos problemas reportados nas planilhas, foi possível identificar quais heurísticas de Nielsen foram violadas com maior frequência. A Figura 5 apresenta a distribuição dos 42 problemas relatados, organizados por heurística e empilhados de acordo com o grau de severidade atribuído (variando de 0 a 4).

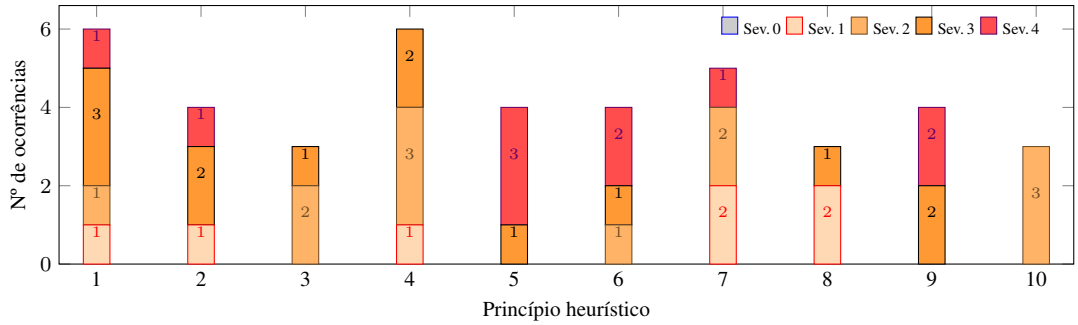


Figura 5. Distribuição dos problemas encontrados por heurística e grau de severidade.

A análise evidencia que os problemas mais críticos (severidade 4) concentram-se em cinco heurísticas: H5 (prevenção de erros) lidera com três ocorrências, seguida por H6 (reconhecimento em vez de memorização) e H9 (recuperação de erros), ambas com duas, e por H1 (visibilidade do status) e H2 (correspondência sistema–mundo), cada uma com uma ocorrência. Já os problemas de severidade 3 distribuem-se principalmente em H1 (três casos) e H2 (dois casos), indicando que essas duas heurísticas reúnem o maior volume absoluto de falhas graves. As demais heurísticas (H3, H4, H7, H8 e H10) apresentaram apenas falhas leves ou moderadas (severidade 1–2), associadas a inconsistências visuais, nomenclaturas pouco claras e ausência de ajuda contextual. Esses resultados sugerem que as correções prioritárias devem atacar os erros críticos de H5, H6 e H9, ao mesmo tempo em que se reforçam feedbacks e rótulos para reduzir o volume de falhas graves ainda presente em H1 e H2.

A análise dos problemas encontrados permitiu identificar que as fragilidades do sistema Trueble podem ser agrupadas em cinco categorias. A primeira diz respeito à ausência de feedback de status adequado. Esse tipo de falha compromete diretamente as heurísticas H1 (Visibilidade do status do sistema) e H9 (Ajudar os usuários a reconhecer, diagnosticar e recuperar de erros), correspondendo a nove das 10 ocorrências classificadas com severidade 4. Esses resultados estão em consonância com as percepções do Avaliador 1, que afirma: *"A mensagem que aparece quando a tabela está completa possui pouco*

destaque e é quase imperceptível", e do Avaliador 2, que destaca: *"A tabela verdade não indica que ainda há fórmulas para preencher em outras colunas"*.

Já a segunda categoria refere-se à consistência visual, na qual foram identificadas inconsistências como botões com cores e larguras distintas, ícones sem uma convenção clara e o uso de tons próximos ao vermelho em células que deveriam ser neutras, o que pode induzir a interpretações equivocadas. Essas falhas impactam principalmente as heurísticas H2 (Correspondência entre o sistema e o mundo real), H4 (Consistência e padrões) e H6 (Reconhecimento em vez de memorização). Houve unanimidade entre os avaliadores quanto à necessidade de uma melhor padronização quanto ao design do sistema, conforme evidenciado pelas observações específicas de cada um: conforme o Avaliador 1, *"os botões que ficam na tela da tabela não estão padronizados, nem em termos de cores (um azul e outro verde), nem em termos de tamanho"*; segundo o Avaliador 2, *"as células para diferir conclusão/premissa estão em um tom próximo de vermelho e parece que está com erro"*; e o Avaliador 3 ressaltou que *"o texto 'Entrar como convidado' tem um contraste muito abaixo do necessário para uma ação clicável"*.

Outra questão relevante, categorizada como terceira, destaca-se o risco de perda de progresso, pois a ação de clicar em Voltar ao menu resulta na exclusão da tabela sem qualquer solicitação de confirmação por parte do sistema. Essa ocorrência, classificada com severidade 4, está relacionada às heurísticas H3 (Controle e liberdade do usuário) e H5 (Prevenção de erros), comprometendo a percepção de controle do usuário sobre a tarefa. Conforme apontado pelo Avaliador 3: *"Ao apertar no botão 'Voltar ao menu' não é oferecido ao usuário nenhuma confirmação. Preenchi toda a tabela e apertei sem querer no botão, perdi todo o preenchimento feito"*.

Completando essas categorias, a quarta está relacionada à flexibilidade e eficiência. O teclado não dispõe de vírgula nem permite a reordenação dos símbolos, e o sistema não oferece uma tela de revisão antes da submissão. Tais limitações, associadas à heurística H7 (Flexibilidade e eficiência de uso), afetam especialmente os usuários mais experientes, que buscam atalhos para concluir as atividades. Os resultados encontrados corroboram a opinião do Avaliador 2, que ressaltou problemas, como: *Antes de clicar em 'Finalizar', poderia mostrar todas as opções que marquei"*; *O teclado não tem vírgula"*; *"Se eu errar uma letra no início da fórmula, preciso apagar tudo para corrigir."*.

Por fim, observa-se, na quinta categoria, a ausência de ajuda contextual. Conceitos como tautologia ou contingência não recebem explicação no momento da escolha, obrigando o discente a recorrer a fontes externas — lacuna que recai sobre as heurísticas H8 (Ajuda e documentação) e H10 (Ajuda no reconhecimento, diagnóstico e recuperação de erros). Essas conclusões estão em sintonia com as percepções do avaliador 3, que evidenciou: *"Não há nenhum local com informações para o usuário. Sendo um app educacional, poderiam haver explicações ou algo como 'relembrar conceito' "* e *"Não sei o que errei ou onde atacar soluções. O único guia é o percentual"*.

Considerando o total de problemas identificados, 23 (55%) foram classificados como graves (severidade 3 ou 4): 10 deles de severidade 4 e 13 de severidade 3. Esses achados, encontram correspondência direta nas cinco categorias de fragilidades identificadas. Portanto, um plano de ação prioritário deve começar por: (i) resolver a ausência de feedback de status, destacando mensagens com alto contraste e revisando a lógica de

validação; (ii) mitigar o risco de perda de progresso, implementando confirmações para ações destrutivas; e (iii) corrigir as falhas de consistência visual por meio da unificação de componentes. Tais ajustes atacariam diretamente a maior parte das ocorrências de severidade elevada, abrindo caminho para refinamentos posteriores, como a melhoria da flexibilidade e eficiência da interface e a inclusão de ajuda contextual.

6. Considerações Finais

Este artigo apresentou a ferramenta *Construir Tabelas*, que integra o ambiente virtual de aprendizagem Trueble, voltado ao ensino de lógica proposicional por meio da construção de tabelas verdade. Diferentemente das soluções existentes, as quais somente permitem a geração automática da tabela, a proposta aqui apresentada permite que o discente construa a tabela verdade, receba validação célula a célula e reflita sobre seus próprios erros. Até o momento, não foram identificadas na literatura ou em repositórios de aplicativos educacionais ferramentas que ofereçam essa combinação de interatividade e feedback formativo, o que confere caráter inédito e potencial inovador à solução apresentada.

Do ponto de vista de usabilidade, a avaliação heurística revelou 42 problemas distribuídos pelas dez heurísticas de Nielsen. A análise mostrou maior concentração de falhas graves nas heurísticas H5, H6 e H9, destacando cinco eixos críticos: feedback insuficiente, inconsistência visual, risco de perda de progresso, baixa flexibilidade e ausência de ajuda contextual. Dez dos problemas identificados receberam grau de severidade 4, comprometendo a execução das tarefas. Com base nesses achados, definimos um conjunto de correções prioritárias: destacar mensagens com alto contraste e revisar a lógica de validação, implementar confirmações para ações destrutivas e unificar componentes. Essas correções eliminariam, de imediato, a maior parte das ocorrências críticas. Foram definidas também melhorias para um refinamento posterior, baseadas nas duas últimas categorias selecionadas: flexibilidade e eficiência e ajuda contextual.

Como trabalho futuro, pretendemos implementar essas correções e realizar avaliações com estudantes de cursos de Computação, de modo a mensurar o impacto da ferramenta no desempenho acadêmico. A avaliação da eficácia pedagógica e do impacto no aprendizado dos alunos permanece como um trabalho futuro essencial. Em 2024, apresentamos um protótipo do *Construir Tabelas* e o avaliamos pelo modelo TAM com 57 estudantes [Moura et al. 2024]. Pretendemos conduzir estudo semelhante no software final, já com as correções de usabilidade implementadas. Por fim, a replicação do estudo em outros contextos educacionais poderá contribuir para a consolidação de um guia de boas práticas no desenvolvimento de aplicativos voltados ao ensino de lógica.

Agradecimentos

Os autores agradecem pelo apoio financeiro da FAPERGS (Projeto ARD/ARC - processo 23/2551-0000761-4).

Referências

Avigad, J. (2022). *Mathematical logic and computation*. Cambridge University Press.

De Troyer, O., Lindberg, R., and Sajjadi, P. (2019). Truebiters: Development, evaluation, and lessons learned. *Smart Learning Environments*, 6:27.

Dias, B. and Finger, A. (2020). Aplicativos para o ensino-aprendizagem de lógica matemática: qual a melhor escolha? In *Anais do XXVI Workshop de Informática na Escola*, pages 111–120, Porto Alegre, RS, Brasil. SBC.

Huth, M. and Ryan, M. (2004). *Logic in Computer Science: Modelling and reasoning about systems*. Cambridge university press.

Lewis, H. and Papadimitriou, C. H. (1997). *Elements of the Theory of Computation*. Pearson.

Lugli, L. C. (2018). Prototipagem de soluções tecnológicas, alfabetização matemática na educação infantil e deficiência sensorial-parametrização de características assistivas.

Moura, G., Poltronieri, I., and Finger, A. (2024). Trueble: Prototipação e avaliação do aplicativo para ensino de tabelas verdade. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 1417–1429. SBC.

Nielsen, J. (1994). *Usability engineering*. Morgan Kaufmann.

Nielsen, J. (2005). 10 usability heuristics for user interface design. Acessado em maio de 2025.

Prietch, S. S. and Pazeto, T. A. (2010). Estudo sobre a evasão em um curso de licenciatura em informática e considerações para melhorias. *WEIBASE, Maceió/AL*.

Sebesta, R. W. (2018). *Conceitos de Linguagens de Programação-11*. Bookman Editora.

Silva, F. S. C. d., Finger, M., and Melo, A. C. V. d. (2006). *Lógica para Computação*. Cengage Learning, São Paulo, 2 edition.

Valente, M. T. (2020). Engenharia de software moderna. *Princípios e Práticas para Desenvolvimento de Software com Produtividade*, 1(24).

Vasconcelos, V. and Andrade, E. (2018). Análise da evasão de alunos na licenciatura em computação. In *Workshop sobre Educação em Computação (WEI)*. SBC.