

Uma Ferramenta Web para Criação e Simulação de Máquinas de Turing

David Alves¹, Paulo T. Guerra¹

¹Universidade Federal do Ceará (UFC)
Campus de Quixadá – Quixadá, CE – Brasil

david.soares@alu.ufc.br, paulodetarso@ufc.br

Abstract. *Turing Machines are abstract computational devices used in the study of Computer Science Theory. Tools like JFLAP, FLApp, and the Teocomp library assist in teaching this concept but suffer from limitations such as outdated interfaces or requirement for programming familiarity. This paper proposes a web-based tool developed with P5.js for creating and simulating Turing Machines, featuring an intuitive and cross-device interface. The tool includes a visual editor for creating machines and a simulator for step-by-step execution, enhancing practical comprehension of the concept.*

Resumo. *Máquinas de Turing são dispositivos computacionais abstratos utilizados no estudo da Teoria da Computação. Ferramentas como JFLAP, FLApp e a biblioteca Teocomp, auxiliam no ensino desse conceito, mas apresentam limitações como interfaces desatualizadas ou a necessidade de alguma familiaridade com programação. Este trabalho propõe uma ferramenta web, desenvolvida com P5.js, para criação e simulação de Máquinas de Turing, utilizando uma interface intuitiva e portátil para diferentes dispositivos. A ferramenta possui um editor que permite criar máquinas de forma visual e um simulador que as executa passo a passo, facilitando a compreensão prática do conceito.*

1. Introdução

Máquinas de Turing (MT) são dispositivos computacionais abstratos que possuem uma fita infinita e uma cabeça de fita que pode ler, escrever, e mover-se sobre esta fita. Essa máquina faz computações, escrevendo ou apagando símbolos na fita utilizando sua cabeça, e produz uma saída quando entra em algum estado de aceitação ou rejeição. A Figura 1 ilustra um exemplo de MT com uma fita contendo os símbolos “abba” e a cabeça de fita localizada sobre o segundo símbolo mais à esquerda da fita.

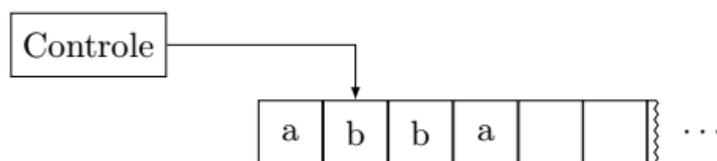


Figura 1. Exemplo de Máquina de Turing.

MTs são essenciais para o estudo de Teoria da Computação, onde permitem a formalização do conceito de algoritmo e por consequência a compreensão dos limites do que pode ou não ser computado. MTs servem como base para muitos conceitos avançados no estudo de algoritmos e são um componente crucial no currículo de cursos superiores em Ciência da Computação e áreas afins.

Tradicionalmente, MTs são ensinadas por professores utilizando exposição oral e quadro branco, apoiado por referências a livros didáticos e atividades-exercício em papel. Isso por vezes pode ser abstrato e difícil de compreender para muitos estudantes. Esse método tradicional, embora efetivo em muitos casos, frequentemente não proporciona uma compreensão prática e profunda do conceito apresentado.

Para [Haleem et al. 2022], as metodologias tradicionais de ensino falham em criar um ambiente de aprendizagem que gere mais envolvimento, dinamicidade e que tenha avaliações mais imediatas. Em contraste a isso está o uso de ferramentas digitais de ensino, que podem promover muitos benefícios para os professores e alunos, principalmente quando utilizadas em conjunto com outras metodologias já existentes.

Na literatura temos algumas ferramentas digitais desenvolvidas para auxiliar no ensino de MTs. A mais amplamente conhecida é o JFLAP¹ [Rodger and Finley 2006], desenvolvida para dispositivos *desktops*, conta com diversos recursos que podem ajudar no estudo e no ensino de Teoria da Computação. [Pereira and Terra 2018] por sua vez apresenta o aplicativo *Formal Languages and Automata Application* (FLApp) para estudo Teoria da Computação em dispositivos móveis. [Vasconcelos and Guerra 2023] apresentam cadernos computacionais para estudos de Teoria da Computação que executa através de *Jupyter Notebooks* e permite a criação, visualização e execução de MTs, que podem ser executados localmente ou por meio de ferramentas online, como Google Colab.

No entanto, muitas dessas ferramentas possuem interfaces desatualizadas e limitações que podem dificultar a experiência de aprendizagem. Por exemplo, para utilizar a abordagem de [Vasconcelos and Guerra 2023] o usuário precisa estar em certo grau familiarizado com a linguagem de programação ou com o ambiente de virtualização. O próprio JFLAP, ferramenta mais amplamente utilizada para esse fim, possui limitações de plataforma e diversos problemas relacionados a interface e experiência do usuário, como mostrado em uma avaliação de experiência feita por [da Silva et al. 2023].

Este trabalho propõe uma ferramenta completamente baseada em Web, utilizando tecnologias modernas para oferecer uma interface mais intuitiva e interativa. Esta ferramenta visa tornar o ensino e a aprendizagem de MTs mais acessíveis, envolventes e eficazes para estudantes de Teoria da Computação. Para tanto, avaliamos as ferramentas disponíveis, com o objetivo de combinar as melhores funcionalidades e aspectos de usabilidade. Adicionalmente, atentamos a requisitos de disponibilidade e acessibilidade em nossa proposta, garantindo que os usuários possam acessar a ferramenta a partir de diferentes dispositivos e locais, sem a necessidade de instalações complicadas, assegurando assim uma experiência inclusiva para todos os estudantes.

¹Disponível em <https://www.jflap.org/jflaptmp/>

2. Máquinas de Turing

Uma Máquina de Turing (MT) é um modelo matemático de computação que possui uma fita infinita dividida em posições, uma cabeça de leitura/escrita que se move sobre as posições e uma unidade de controle que segue um conjunto finito de regras, sendo formalmente descrita como segue.

Definição 1. Uma Máquina de Turing M é uma 7-tupla $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$, onde:

- Q é um conjunto finito de estados;
- Σ é o alfabeto de entrada tal que $\square \notin \Sigma$;
- Γ é o alfabeto da fita, onde $\Sigma \subseteq \Gamma$ e $\square \in \Gamma$;
- $\delta : Q' \times \Gamma \rightarrow Q \times \Gamma \times \{E, D\}$ é a função de transição, onde $Q' \subseteq Q - F$;
- $q_0 \in Q$ é o estado inicial;
- \square é o símbolo que representa uma célula vazia;
- $F \subseteq Q$ é o conjunto de estados de aceitação.

Uma MT pode ser ilustrada por meio de um diagrama de estados, onde os estados são representados por círculos e as transições por setas rotuladas. Cada rótulo $A \rightarrow B, C$ em uma seta indica que a leitura do símbolo A desencadeia a escrita do símbolo B e um movimento na direção C . O estado inicial é indicado por uma seta sem origem em outro estado. Os estados de aceitação são representados por círculos duplos. A Figura 2 ilustra a MT formalmente definida como $M = (\{q_0, q_1, q_2\}, \{a, b\}, \{a, b, \square\}, \delta, q_0, \square, \{q_1\})$ onde $\delta(q_0, a) = (q_1, a, D)$, $\delta(q_0, b) = (q_2, b, E)$, $\delta(q_0, \square) = (q_0, \square, D)$.

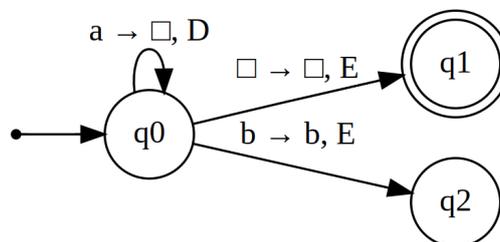


Figura 2. Exemplo de diagrama de estados de Máquina de Turing.

A computação de uma MT é o processo pelo qual a máquina manipula o conteúdo da fita de acordo com as regras definidas pela função de transição. Nesse processo, a máquina altera diferentes aspectos, como estado de controle, conteúdo da fita e posição da cabeça de leitura/escrita na fita. A combinação momentânea desses fatores é chamado de configuração.

Definição 2. Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$, o conjunto de possíveis configurações de M é dado pelo conjunto $\{uqv \mid q \in Q \text{ e } u, v \in \Gamma^*\}$, onde uqv indica que M está no estado de controle q , que sua fita tem conteúdo uv e que sua cabeça de leitura/escrita está na $|u|$ -ésima posição.

Uma computação de uma MT M é uma sequência de configurações que segue uma regra de formação baseada na função de transição da máquina.

Definição 3. Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$, computação de M é uma sequência de configurações C_0, C_1, \dots, C_n tal que para todo C_i e C_{i+1} : $C_{i+1} = rcv$ se $C_i = qbv$ e $\delta(q, b) = (r, c, E)$; $C_{i+1} = ucrv$ se $C_i = uqbv$ e $\delta(q, b) = (r, c, D)$; ou $C_{i+1} = uracv$ se $C_i = uaqbv$ e $\delta(q, b) = (r, c, E)$, para $a \in \Gamma$.

Dizemos que $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ computa uma cadeia $w \in \Sigma^*$ se existe uma computação que inicia na configuração q_0w e continua até que a máquina entre em uma configuração uqw para a qual δ não está definido, o que denominamos de configuração de parada. Se $q \in F$ dizemos que M aceita w . Caso contrário, dizemos que M rejeita w . A sequência $q_0aa, \square q_0a, \square\square q_0, \square q_1\square$ indica, por exemplo, que a máquina da Figura 2 aceita a cadeia aa .

3. Ferramentas para o Ensino de Máquinas de Turing

No trabalho de [Procopiuc et al. 1996], foi desenvolvida a ferramenta JFLAP, um *software desktop* desenvolvido utilizando a linguagem de programação Java que permite a simulação de diversas estruturas abstratas da Teoria da Computação. O JFLAP permite a criação interativa das MTs por meio do movimento de arrasta e solta ao selecionar os botões correspondentes as ações no menu superior. Além disso, o JFLAP permite que seja possível executar e visualizar a computação das MTs tanto em formato de diagrama de estados como em formato de passo a passo na fita.

Em [Pereira and Terra 2018] foi desenvolvido e avaliado um aplicativo para dispositivos móveis FLApp, que permite o estudo de vários conteúdos de Linguagens Formais e Teoria dos Autômatos. O FLApp permite a criação e a simulação de diversos tipos de máquinas, além de permitir a conversão entre autômatos finitos determinísticos e não determinísticos, checagem de ambiguidades, remoção de recursões, minimização de autômatos, e muito mais. Quando se trata de MTs, o aplicativo permite a criação e simulação de MTs determinísticas. O usuário necessita digitar as regras, os estados e a palavra a ser analisada. O resultado é mostrado em formato de diagrama de estados juntamente com um passo a passo dos resultados sobre a sua fita.

Em [Vasconcelos and Guerra 2023], os autores desenvolveram uma biblioteca para auxílio no ensino de Teoria da Computação denominado Teocomp. Ela foi construída utilizando a linguagem de programação Python e disponibilizada utilizando a ferramenta *Jupyter Notebook*, um ambiente para desenvolvimento interativo de código. A biblioteca construída pode ser utilizada para o aprendizado e ensino de MTs, para isso, os estudantes descrevem as máquinas conforme definido na biblioteca, e recebem como retorno a visualização da máquina desejada.

Realizamos a análise de cada uma dessas ferramentas de modo a identificar características e funcionalidades que possam ser incorporadas a ferramenta desenvolvida neste trabalho. Além disso, baseado em uma análise empírica, foi avaliado também aspectos de Experiência do Usuário (*User Experience*, ou UX) e Interface do Usuário (*User Interface*, ou UI) que foram mantidas ou evitadas em nossa implementação.

3.1. Análise das Ferramentas de Ensino

Nossa análise foi fundamentada em três principais levantamentos: principais características; principais funcionalidades com foco em MTs; e aspectos de UX e UI. Quanto as principais características, descrevemos brevemente o escopo global da ferramenta, como plataforma computacional na qual pode-se usar a ferramenta, modo de uso, etc. Já quanto as principais funcionalidades com foco em MTs, resumimos as informações relativas às MTs, visto que algumas ferramentas que abrangem uma série de outros assuntos relacionados a Teoria da Computação. Quanto a UX e UI, destacamos aspectos de utilização

da ferramenta, tanto positivos como negativos para a usabilidade do usuário em geral. A Tabela 1 compila as principais informações.

As três ferramentas foram construídas para serem executadas em plataformas computacionais diferentes, e suas formas de interação também variam, algumas possuindo interface gráfica, enquanto outras não. Essas características influenciam a quantidade de usuários que cada ferramenta pode atingir, pois a utilização é muitas vezes limitada a um tipo específico de plataforma, com exceção à ferramenta de [Vasconcelos and Guerra 2023]. Além disso, algumas ferramentas requerem a instalação de dependências adicionais, o que pode dificultar ainda mais sua utilização.

Quando se trata das principais funcionalidades relacionadas ao estudo de MTs, o JFLAP se destaca pelas diversas funções para criação, edição e simulação de MTs oferecidas de forma interativa via interface gráfica. Ela também permite a importação e exportação de arquivos e a integração de MTs com sub-rotinas. A biblioteca de [Vasconcelos and Guerra 2023] oferece funcionalidades similares ao JFLAP, mas é limitada pela ausência de meios de definir graficamente a estrutura de MTs, bem como não possuindo funções de importação e exportação.

Em relação ao aplicativo FLApp, no momento da análise, ele não estava disponível na loja oficial do sistema Android. Sua utilização só foi possível por meio da instalação manual do APK, utilizando o repositório de códigos que os autores disponibilizaram. Nesta análise, observou-se que, em relação ao estudo de MTs, o FLApp possuía apenas funcionalidades básicas, não estando presente a importação e exportação de arquivos, simulação visual passo-a-passo, integração de sub-rotinas ou multiteste. Esses fatores tornam sua utilização limitada para fins educacionais, pois essas funcionalidades são importantes para o aprendizado eficaz de MTs.

Em relação aos aspectos de UX, foi possível identificar diversos pontos positivos e negativos nas ferramentas analisadas. Algumas delas possuem menus para navegação que organizam as funções da ferramenta, como o JFLAP. Contudo, todas as ferramentas testadas apresentam algum tipo de problemas de estrutura e desordem visual, dificultando a localização de informações importante. A responsividade e falta de compatibilidade com dispositivos móveis são um grande problema para o JFLAP e Teocomp. Além disso, algumas respostas das ferramentas às ações dos usuários não são claras ou ficam ocultas, dificultando a detecção de erros. Problemas adicionais de UX incluem áreas de suporte e ajuda desatualizadas ou mal organizadas, ausência de atalhos de teclado e grande curva de aprendizado para utilização dessas ferramentas.

Em termos de UI, também é possível notar diversos pontos a serem melhorados, como interfaces antiga e ícones não tão intuitivos. Além disso, devido ao uso de modelos de interface mais antigos, as cores das telas do JFLAP e do FLApp não oferecem um bom contraste. A falta de opção para mudança de tema de cores pode causar, por exemplo, fadiga visual em usuários sensíveis a exposição a luz forte da tela por longos períodos.

4. Aplicação Web para Criação e Simulação de Máquinas de Turing

Com base na análise realizada, implementamos uma ferramenta para criação e simulação de MTs utilizando P5.js [McCarthy et al. 2015], uma biblioteca² JavaScript de código

²Disponível em: <https://p5js.org/>

Tabela 1. Análise das ferramentas para ensino de Máquina de Turing.

Análise	JFLAP	Teocomp	FLApp
Principais características	Aplicação <i>desktop</i> ; possui interface gráfica interativa para criação de MTs.	Aplicação <i>desktop</i> e móvel; MTs são criadas por meio de códigos escritos.	Aplicação móvel; possui interface gráfica interativa para criação de MTs.
Principais funcionalidades com foco em MTs	Criação, visualização e testes de MT e variantes; permite importar e exportar MT; possui controle de simulação de MTs em diagrama de estados e/ou fita; integração de sub-rotinas.	Criação, visualização e testes de MT e variantes; possui controle de simulação de MTs em diagrama de estados e/ou fita; integração de sub-rotinas.	Criação, visualização e testes de MT e variantes; possui visualização do resultado da simulação de MTs em diagrama de estados e/ou fita.
Experiência de usuário	Ajuda, suporte e documentação externa; não há ajuda interna como tutoriais, alertas e/ou <i>tooltips</i> ; possui menus para navegação; navegação parcialmente confusa e com desordem visual; algumas funções não mostram se foram executadas ou não de maneira clara; faltam atalhos comuns como <i>ctrl+z</i> (desfazer) e <i>ctrl+y</i> (refazer); curva de aprendizado alta; alta performance.	Ajuda e documentação interna; possui suporte e ajuda parcial, sem abranger todas as dúvidas para navegação e utilização da biblioteca; navegação linear com eventual desordem visual; gestos compatíveis com dispositivos móveis; possui apenas atalhos de teclado de editores de texto padrão; curva de aprendizado alta; alta performance.	Não há ajuda e/ou suporte; possui menus para navegação; navegação parcialmente confusa e com desordem visual; algumas funções não mostram se foram executadas ou não de maneira clara; gestos compatíveis com dispositivos móveis; curva de aprendizado alta; alta performance.
Interface de usuário	Não responsivo; documentação antiga; interface antiga, com pouca diferenciação por cores e ícones não intuitivos.	Não responsivo para os editores de texto testados; cores e ícones dependem da ferramenta que o usuário utiliza para executar os notebooks.	Interface responsiva; interface antiga, com pouca diferenciação por cores e ícones pouco intuitivos.

aberto que busca tornar programação visual mais acessível e intuitiva. Esta escolha foi motivada pela necessidade de desenvolver uma ferramenta moderna, capaz de melhorar aspectos e limitações das ferramentas existentes. A Figura 3 apresenta a tela principal da aplicação.

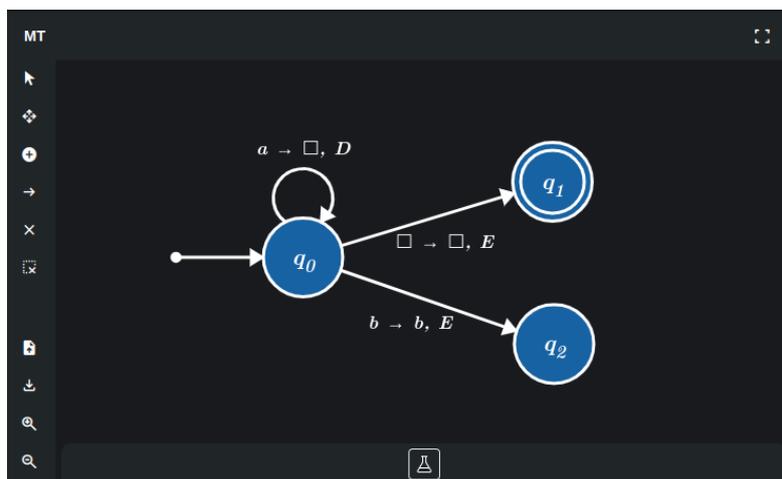


Figura 3. Tela principal do editor.

Um dos principais objetivos ao desenvolver esta ferramenta foi garantir que ela fosse leve e responsiva. Graças ao uso de P5.js, a aplicação é leve em termos de desempenho e pode ser executada diretamente em navegadores web sem a necessidade de instalações adicionais. Devido a sua interface responsiva, ela pode ser utilizada em uma variedade de dispositivos, incluindo *smartphones* e *tablets*. Isso amplia significativamente o acesso, permitindo que estudantes utilizem a ferramenta em sala de aula por meio de seus próprios dispositivos.

A ferramenta possui um editor de MTs, que permite aos usuários criar e editar máquinas de maneira intuitiva e visual. Utilizando os controles laterais, é possível criar novos estados e definir as transições entre esses estados de forma visual, especificando os símbolos de leitura e escrita, bem como a direção do movimento da cabeça de leitura/escrita. Por fim, os usuários podem configurar o estado inicial e os estados de aceitação.

Diversos elementos do editor possuem *tooltips*, que são pequenas mensagens de texto que aparecem quando o usuário passa o cursor sobre eles. Esses *tooltips* fornecem informações adicionais e dicas sobre a funcionalidade de cada item, tornando a ferramenta mais intuitiva e fácil de usar. Essa informação é importante especialmente para usuários iniciantes, pois oferece orientações imediatas, facilitando o aprendizado e a utilização eficiente do editor.

A ferramenta também traz funcionalidades importantes para a experiência do usuário, como as ações de desfazer e refazer a criação, remoção ou edição de elementos da MT. A ferramenta oferece ainda a possibilidade de importar e exportar arquivos, facilitando a continuidade de projetos e o compartilhamento de MTs entre diferentes usuários.

Além das capacidades de edição, a ferramenta também inclui um simulador de

MTs. Com o simulador, os usuários podem realizar a execução de uma máquina passo a passo, acompanhando visualmente como a fita é modificada e como a cabeça de leitura/escrita se move nela. Durante a simulação, o estado atual é destacado e a posição da cabeça de leitura/escrita e o conteúdo da fita são dinamicamente atualizados, proporcionando uma compreensão clara do processo de computação realizado pela máquina para uma dada cadeia de entrada. A Figura 4 apresenta a tela do ambiente de simulação.

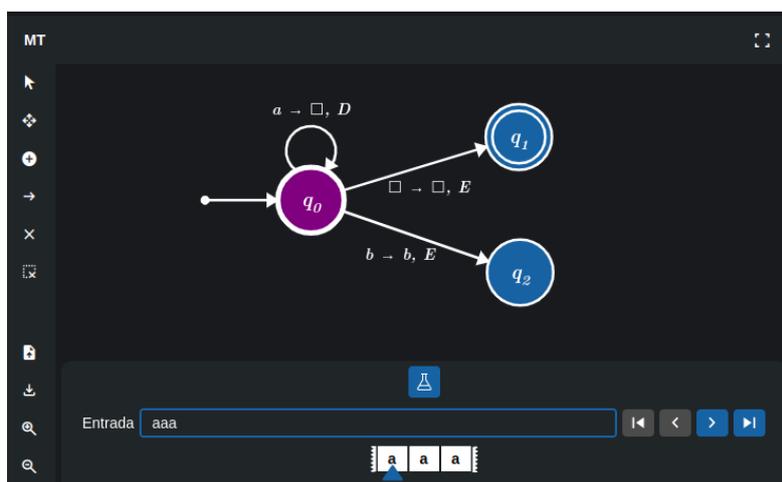


Figura 4. Tela do ambiente de simulação.

O simulador possui um campo onde os usuários podem inserir cadeias que serão depois computadas pela MT. Com isso, os usuários podem testar diversos casos, facilitando entendimento do funcionamento da MT e a identificação de possíveis erros. Essa funcionalidade de simulação da autonomia para os estudantes, permitindo que estes vejam em tempo real como suas máquinas processam diferentes entradas e em que pontos podem melhorar.

Os controles de simulação são compostos por quatro botões principais que permitem executar um passo, retroceder um passo, retornar para o início da computação e ir para o fim da computação. A capacidade combinada de avançar ou retroceder um passo é particularmente útil para depuração, permitindo que os estudantes entendam e corrijam erros. A função de ir para o fim oferece uma visão geral rápida do resultado da computação, dando agilidade para o teste sobre linguagens. A função de ir para o início facilita o processo de revisão e compreensão do funcionamento da máquina. Esses controles permitem que os usuários executem suas MTs de forma detalhada, tornando o processo de aprendizado mais interativo e eficiente.

Os usuários também dispõem de diversos *feedbacks* na ferramenta, indicando, por exemplo, a existência de erros na definição da MT construída (Figura 5). Esses *feedbacks* são importantes para orientar os estudantes sobre possíveis inconsistências na definição formal das máquinas e outras divergências com os conceitos estudados, proporcionando uma experiência formativa durante o processo de criação e edição de MTs.

Além disso, a ferramenta oferece informações visuais detalhadas quanto ao sucesso das simulações realizadas (Figura 6), fornecendo aos usuários uma representação clara e imediata dos resultados obtidos, elemento importantes para o entendimento do

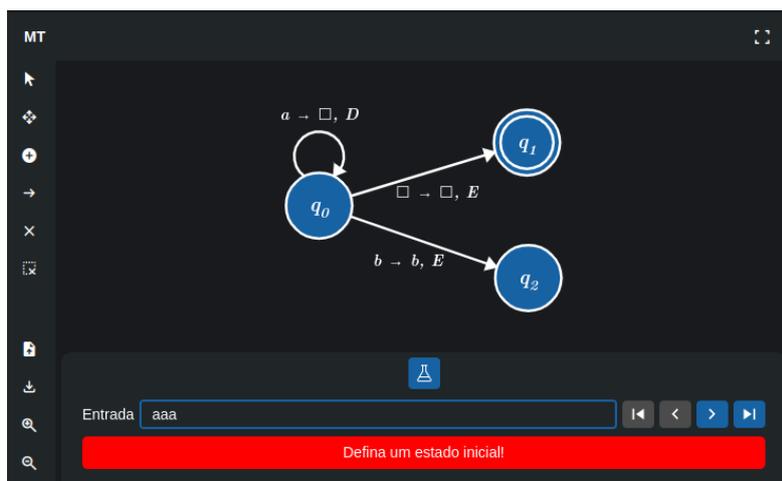


Figura 5. Tela com mensagem de erro.

comportamento das MTs em diferentes cenários de entrada e para a validação do funcionamento correto das implementações desenvolvidas.

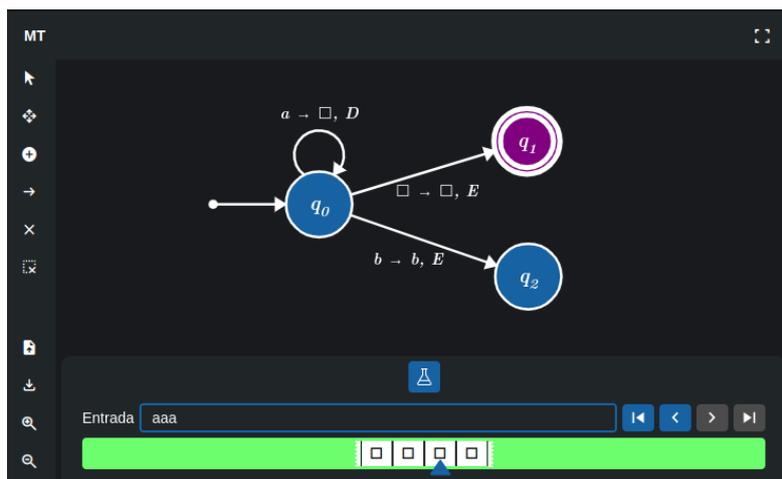


Figura 6. Tela com indicação de sucesso.

Como ferramenta didática, a aplicação facilita e amplia possibilidades de estudo sobre MTs. Ela pode ser combinada com outros recursos didáticos, como livros e listas de exercício, proporcionando uma experiência mais interativa e dinâmica ao método tradicional. A ferramenta permite assim os estudantes explorarem conceitos teóricos de maneira prática, promovendo sobretudo autonomia ao oferecer um ambiente onde estes podem experimentar, testar e ajustar suas próprias criações de máquinas.

4.1. Avaliação com usuários

A avaliação inicial da ferramenta foi feita por meio de uma avaliação prática com a participação de cinco alunos do curso de Ciência da Computação, todos com experiência prévia na disciplina de Teoria da Computação, não sendo necessária familiaridade com ferramentas de construção de MTs. O objetivo foi verificar a usabilidade, a eficiência e a experiência geral de utilização das ferramentas.

Inicialmente, os alunos foram solicitados a criar uma MT que decide a linguagem $\{w \in 0^*1^* \mid w \text{ tem o mesmo número de } 0 \text{ e } 1\}$ utilizando as ferramentas JFLAP, Teocomp e a proposta nesse trabalho. Links para as páginas principais dessas ferramentas foram fornecidos. Os alunos podiam ou não ter as ferramentas instaladas em seus dispositivos. Esta etapa prática busca induzir que os alunos explorem as funcionalidades de cada ferramenta, como em um cenário de uso real.

Após completar a tarefa, os alunos foram convidados a preencher um formulário de avaliação. No formulário, eles indicaram a ordem de preferência entre as ferramentas utilizadas, considerando critérios como facilidade de uso, clareza das interfaces, recursos disponíveis e experiência geral de utilização. Além disso, foi solicitado um *feedback* livre sobre a experiência comparativa entre as ferramentas, permitindo que os alunos expressassem suas opiniões e sugestões de melhorias.

Segundo as respostas ao formulário, quatro participantes conheciam previamente a biblioteca Teocomp. Nenhum informou conhecer o JFLAP ou a ferramenta proposta neste trabalho. Ao avaliarem a experiência de uso, quatro participantes indicaram como médio ou difícil a experiência de uso do JFLAP e do Teocomp, enquanto todos os participantes indicaram como fácil ou muito fácil a experiência com a ferramenta proposta.

Nos *feedbacks* livres, os participantes indicaram que a ferramenta deste trabalho proporcionou uma experiência positiva no processo de criação de MTs. Um participante mencionou ter inicialmente encontrado dificuldades ao relembrar os conceitos básicos, mas que gostou da ferramenta. Outro participante relatou que a ferramenta deste trabalho apresenta uma interface significativamente mais interativa em comparação às outras plataformas testadas e que sua abordagem visual na criação de MTs reduz consideravelmente a complexidade em comparação à metodologia de escrita direta, como observado na Teocomp.

Ao solicitar a indicação de ordem de preferências, quatro dos participantes escolheram como preferida a ferramenta proposta neste trabalho, um participante indicou o JFLAP. Teocomp foi a segunda escolha de todos os participantes. Os resultados mostraram que a ferramenta proposta foi bem recebida pelos alunos, sendo positivamente avaliada por sua interface intuitiva e amigável. A facilidade na criação e edição visual das MTs foi destacada como uma vantagem.

Em geral, a avaliação refletiu uma recepção positiva à ferramenta proposta, destacando seus pontos fortes e identificando áreas para futuras melhorias e desenvolvimentos. Os *feedbacks* recebidos serão utilizados para orientar as próximas etapas de desenvolvimento da ferramenta, visando torná-la ainda mais robusta e útil para o ensino e a aprendizagem de MTs.

5. Conclusões

Este trabalho apresenta uma ferramenta completamente baseada na web para ensino de Máquinas de Turing, utilizando tecnologias atuais para oferecer uma interface mais intuitiva e interativa. A ferramenta foi desenvolvida utilizando P5.js e sua construção foi guiada a partir da análise das limitações e das melhores práticas encontradas em ferramentas existentes, como o JFLAP, Teocomp e FLApp.

Nosso objetivo foi criar uma ferramenta que fosse leve e responsiva, adaptável a diferentes dispositivos, como *smartphones* e *tablets*. Essa abordagem visa não apenas facilitar a aprendizagem de MTs em diversos ambientes educacionais, mas também promover uma experiência de ensino mais envolvente para o estudo do tema.

A ferramenta apresentada nesse trabalho possui um editor de MTs que permite aos usuários criar e editar máquinas de maneira visual. Nessa interface, é possível criar novos estados, representados por círculos, e definir as transições, representadas por setas, entre esses estados, especificando rótulos com os símbolos de leitura e escrita, bem como a direção do movimento da cabeça de leitura/escrita. O editor inclui funções como desfazer e refazer, bem como a possibilidade de importar e exportar MTs através de arquivos.

A ferramenta também inclui um simulador de MTs, onde os usuários podem visualizar a execução de uma máquina passo a passo, acompanhando como a fita é modificada e como a cabeça de leitura/escrita se move entre posições da fita. Durante a simulação, o estado, posição da cabeça de leitura/escrita e o conteúdo da fita são dinamicamente atualizados, proporcionando uma compreensão clara do processo de computação realizado pela máquina para uma dada cadeia de entrada.

A avaliação da ferramenta foi realizada por um grupo de alunos de Ciência da Computação que já cursaram a disciplina de Teoria da Computação. Os alunos resolveram uma questão utilizando diferentes ferramentas. O *feedback* quanto ao uso da ferramenta proposta neste trabalho foi positivo, especialmente quanto a sua interface.

Como ferramenta didática, nossa proposta amplia possibilidades de estudo sobre MTs. Ela pode ser combinada a outros recursos didáticos, como livros e listas de exercício, proporcionando uma experiência mais interativa e dinâmica ao método tradicional. A ferramenta permite assim que os estudantes explorem conceitos teóricos de maneira prática, promovendo sobretudo autonomia ao oferecer um ambiente onde estes podem experimentar, testar e ajustar suas próprias criações de máquinas.

Como trabalhos futuros, planejamos expandir a plataforma para incluir suporte a diferentes variantes de MTs, como MTs não-determinísticas e máquinas multi-fitas, bem como outros modelos mais simples, como autômatos finitos determinísticos, não-determinísticos e autômatos de pilha. Desse modo a ferramenta pode ser usada para outras disciplinas, como de linguagens formais e autômatos.

Além disso, pretendemos integrar recursos didáticos adicionais, como exercícios interativos, tutoriais, videoaulas e outros vídeos explicativos. Nosso objetivo é transformar a ferramenta em uma plataforma educacional completa, voltada ao ensino de Teoria da Computação, facilitando tanto o aprendizado autônomo quanto o ensino assistido por um professor.

Agradecimentos

Os autores deste trabalho declaram a utilização de ferramentas e tecnologias de IA Generativa no processo de escrita desse trabalho. Foram utilizados os modelos GPT-3.5 e GPT-4.0, sendo estes aplicados principalmente na revisão da redação final do texto, contribuindo para a clareza, dimensionamento e coesão do texto apresentado. Ressaltamos que, apesar do auxílio dessas ferramentas, os autores são inteiramente responsáveis pelo conteúdo deste trabalho.

Referências

- da Silva, L. G., de Abreu Dias, B., Finger, A. F., and Silva, W. (2023). Avaliação da experiência de uso do jflap como recurso pedagógico no ensino de linguagens formais. In *Anais do XXXIV Simpósio Brasileiro de Informática na Educação*, pages 995–1006. SBC.
- Haleem, A., Javaid, M., Qadri, M. A., and Suman, R. (2022). Understanding the role of digital technologies in education: A review. *Sustainable Operations and Computers*, 3:275–285.
- McCarthy, L., Reas, C., and Fry, B. (2015). *Getting started with P5.js: Making interactive graphics in JavaScript and processing*. Maker Media, Inc.
- Pereira, C. H. and Terra, R. (2018). A mobile app for teaching formal languages and automata. *Computer Applications in Engineering Education*, 26(5):1742–1752.
- Procopiuc, M., Procopiuc, O., and Rodger, S. H. (1996). Visualization and interaction in the computer science formal languages course with jflap. In *Technology-Based Re-Engineering Engineering Education Proceedings of Frontiers in Education FIE'96 26th Annual Conference*, volume 1, pages 121–125. IEEE.
- Rodger, S. H. and Finley, T. W. (2006). *JFLAP: an interactive formal languages and automata package*. Jones & Bartlett Learning.
- Vasconcelos, D. R. and Guerra, P. T. (2023). Ensinando teoria da computação com jupyter notebook. In *Anais do XXXI Workshop sobre Educação em Computação*, pages 9–19. SBC.