

ARCABOUÇO DIDÁTICO DA MÁQUINA DE TURING

Diogo dos Santos

Universidade Federal da Paraíba – UFPB¹

Departamento de Ciências Exatas – DCX

Curso de Licenciatura em Ciências da Computação - LCC

diogo.santos@dce.ufpb.br

Abstract. *The Turing Machine (TM), developed in the first half of the last century, still maintains an important relevance for the courses of computer science, logic and mathematics, as a reference tool to establish the extent and limits of automatic resolution of problems. Its concept is attached to the renowned "Church-Turing Thesis", a collection of works on functions whose values are effectively calculable; that is, those functions that can be solved by (computable) algorithms. The MT remains the main model preferred by theorists who investigate issues concerning various problems in computer theory; in particular, the theories of decidability and computational complexity. In this context, a TM framework becomes a useful tool for learning such concepts, since it enables the learner, the construction of the most diverse TMs and the verification of the most varied programs built for them, completely independent of a supervisor.*

Keywords: *Turing machine, Framework, Reversible words.*

Resumo. *A Máquina de Turing (MT), desenvolvida na primeira metade do século passado, mantém ainda hoje uma importante relevância para os cursos de ciências da computação, lógica e matemática, por ser essa uma ferramenta referencial para estabelecer a extensão e os limites da resolução automática de problemas. Seu conceito está adstrito à renomada "Tese de Church-Turing", uma coletânea de trabalhos sobre as funções cujos valores são efetivamente calculáveis; ou seja, aquelas funções que podem ser resolvidas através de algoritmos (computáveis). A MT continua a ser o principal modelo preferido pelos teóricos que investigam questões relativas a vários problemas em teoria da computação; em particular, as teorias da decidibilidade e da complexidade computacional. Um arcabouço de uma MT torna-se, nesse contexto, uma ferramenta útil para a aprendizagem de tais conceitos, já que possibilita ao aprendiz, a construção das mais diversas MTs e a verificação dos mais variados programas para essas, de maneira totalmente independente de um supervisor.*

Palavras-chave: *Máquina de Turing, Arcabouço, Palavras reversíveis.*

¹ Trabalho de Conclusão de Curso (TCC) na modalidade Artigo apresentado como parte dos pré-requisitos para a obtenção do título de Licenciado em Ciência da Computação pelo curso de Licenciatura em Ciência da Computação do Centro de Ciências Aplicadas e Educação (CCAEE), Campus IV da Universidade Federal da Paraíba, sob a orientação do professor Dr. Joelson Nogueira de Carvalho.

1 Introdução

1.1 Importância da MT para a computação

A Máquina de Turing foi criada no ano de 1935 pelo matemático, lógico e cripto-analista inglês Alan Mathison Turing; foi uma ideia inovadora, que definia um sistema formal axiomático simples para manipular símbolos, que seria capaz de empregar operações efetivas para resolver problemas diversos, estabelecendo um formalismo para os algoritmos; a primeira menção da máquina de Turing ocorreu em um artigo do próprio Turing, publicada em 1936: “On computable numbers, with an application to the Entscheidungsproblem” (CASSOL et al. 2010) , onde o autor empregava seu uso como aplicação para um dos problemas de Hilbert. Este feito notável trouxe a Alan Turing a alcunha de “Pai da Computação”, pois proporcionou o desenvolvimento dos computadores modernos. O filósofo João Fernandes Teixeira cita:

[...] um estudante do King’s College, em Cambridge, na Inglaterra, teve uma ideia para tentar resolver o chamado Problema de Hilbert, uma famosa questão matemática. Na mesma época, um grande debate entre os matemáticos de Princeton, nos Estados Unidos, levava ao aparecimento de um novo tipo de cálculo lógico, criado para fornecer uma base matemática para a ideia de realizar uma computação. Estas duas iniciativas diferentes formaram as bases para o que mais tarde ficou conhecido como “ciência da computação”. Dez anos mais tarde, John Von Neumann decidiu usar essas ideias para, efetivamente, construir os primeiros computadores modernos (Teixeira, 1998 - p.20).

O autor reforça que, após milhares de anos de história da Matemática, não se sabia o que era um algoritmo e tampouco o que era uma computação; com essa máquina criada por Turing, é possível resolver qualquer problema decidível. Ainda para o autor, a máquina de Turing constitui a melhor formalização da noção de algoritmo de que se tem notícia na história da Matemática.

O programa de uma Máquina de Turing é conhecido como função programa; nele encontra-se toda a lógica para resolver um determinado problema decidível, e é o que faz dessa formalização de algoritmo como sendo a melhor para problemas com essa característica. Um dos mais consagrados autores na área de Teoria da Computação, Michael Sipser, observa que “Uma máquina de Turing pode fazer tudo que um computador real pode fazer” (SIPSER, 2006 – pg.137), reforçando a ideia de que qualquer computador digital é, em princípio, uma máquina de Turing. Assim, podemos afirmar que a Máquina de Turing é o princípio geral para a construção de computadores digitais, pois, por meio dela, podemos executar qualquer algoritmo.

1.2 Relevância do uso de simuladores

Hoje em dia, devido aos avanços tecnológicos que trouxeram novas ferramentas para a educação, existe a necessidade de inovar os processos de ensino, para promover o aumento no estímulo e a consequente melhoria no aproveitamento do aluno, aproveitando esses recursos.

O uso de simuladores já é bem disseminado na educação e em outras áreas; são inúmeras as vantagens do uso de tais ferramentas, pois elas permitem uma imersão no ambiente simulado, aproximando-se da realidade tanto quanto possível. Não é difícil imagi-

nar a relevância de simuladores para treinamento profissional em algumas áreas, como a medicina e a aviação civil e militar. Dentre outros benefícios, destacam-se a redução de custos, prática descentralizada e altamente disponível e em alguns casos, a distância de situações de risco.

O uso de um arcabouço da MT como ferramenta de ensino permitirá a construção de diversas MTs para resolver os mais diversos problemas, tanto como reconhecedor de linguagens como no modo transdutor. Os usuários terão acesso a um ambiente gráfico amigável, que simula a estrutura da máquina; simples, uniforme e comum a todas as possíveis implementações. Nesse ambiente será possível inserir todas as informações que permitirão o funcionamento da máquina, inclusive com a detecção de possíveis erros na construção de programas ou mesmo durante o fornecimento de dados para processamento. Isso possibilita a verificação de diversos exercícios, tanto em sala-de-aula, quanto fora dela, permitindo ao aluno usuário testar as suas respostas a exercícios solicitados, por exemplo.

Utilizar um simulador é de suma importância; segundo LOPES e OLIVEIRA (2013, p.16),

[...] os simuladores educativos são recursos que têm sido geralmente descuidados pelos educadores, formadores e docentes, assim como pelos órgãos de gestão dos estabelecimentos de ensino e formação. Apesar de não se tratar de uma panaceia para a educação do futuro, devem ser equacionados como mais uma ferramenta disponível a utilizar no processo de ensino-aprendizagem.

Ferramenta essa que poderá ser útil aos alunos da disciplina de Teoria de Computação, pois fará com que os alunos tenham uma visão mais realista das operações realizadas, já que os simuladores podem ser definidos como representações de uma situação real num programa de computador LOPES e OLIVERIA (2013, p.12). Com um simulador aplicado ao assunto Máquina de Turing, é possível fazer um acompanhamento mais concreto das operações e mudanças de estados realizados pela função programa, já que, sem um apoio, é um processo abstrato e se torna difícil sua visualização e acompanhamento da execução de uma função programa aplicado a este assunto. A utilização de computadores e recursos multimídia na aprendizagem pode favorecer o surgimento de ideias, emoções, atitudes e habilidades, as quais propiciam uma relação cognitiva e interativa dos estudantes com o objeto de conhecimento BALBINOT; TIMM; ZARO (2009, p.2). Habilidades de entender o assunto que podem ser estimuladas ainda mais com a utilização de um simulador que possa fazer uma representação mais realista.

1.3 Objetivos

Desenvolvimento de uma aplicação que possa receber funções programas de uma MT genérica e ser capaz de processar tais programas apresentando seus respectivos resultados. Assim, fazendo a simulação de execução de uma MT para melhor entendimento do assunto aos alunos da disciplina de Teoria da Computação.

2 Problemas Decidíveis

O Segundo Congresso Internacional de Matemática aconteceu em Paris no ano de 1898; nele, um dos mais proeminentes matemáticos da época, David Hilbert, nascido na Prússia em Königsberg (hoje Kalinigrado), apresentou uma lista de 23 problemas, supostamente elaborados para alavancar o desenvolvimento da matemática. Essa lista denominava-se *Entscheidungsproblem* - Termo alemão para "Problema de decisão". Um problema de decisão é uma questão sobre um sistema formal, cuja resposta é geralmente

binária, do tipo sim-ou-não; em outras palavras, se questiona se há ou não uma solução matemática para o mesmo.

É bem provável que boa parte dos trabalhos de Turing não tivessem sido realizados sem a instigação provocada pelos problemas de Hilbert. A Máquina de Turing foi criada como resposta a alguns desses problemas.

3 Arcabouços e Simuladores

Um simulador é um dispositivo que reproduz as condições específicas de uma atividade. Nesse contexto, a estrutura subjacente ao ambiente de simulação pode ser denominada “arcabouço”. Isso porque, em mecânica, esse termo designa um conjunto de peças que faz funcionar algo, e como será mostrado mais adiante, a Máquina de Turing não é dotada de nenhuma função lógica ou matemática para a resolução de problemas, pois essas fazem parte da entrada e são definidas pelo usuário. Em outras palavras, o arcabouço representa então a parte estrutural do simulador, sendo desprovido de funções lógicas e matemáticas para a resolução de problemas.

4 Máquinas de estados Finitos

Desprezando a enorme complexidade dos conceitos da cibernética que definem o termo “Máquina”, é comum abstrair a imagem de um mecanismo que recebe algo em sua entrada, transforma essa entrada em algo que é colocado em sua saída; é evidente que a atividade fundamental de tal mecanismo é a transformação. A relação entre uma máquina real e transformação que ela realiza é muito estreita; a transformação é a representação canônica da máquina, enquanto que a máquina incorpora a transformação (ASHBY, 1970 - pg. 34).

Máquinas de Estado Finito: É um modelo abstrato de uma máquina com memória interna primitiva; Uma Máquina de Estado Finito M , pode ser formalmente descrita por uma quintupla $M = [Q, X, Y, f_q, f_y]$, onde (Gill, 1962):

- a) X é um conjunto finito de símbolos de entrada.
- b) Y é um conjunto finito de símbolos de saída.
- c) Q é um conjunto finito de estados.
- d) Uma função de estado-próximo é definida por: f_q de $Q \times X \rightarrow Q$.
- e) Uma função de saída é definida por: f_y de $Q \rightarrow Y$.
- f) Um estado inicial, incluído em Q .

A função f_q é uma função do próximo estado. Ela leva pares (estados, entradas) em estados. Portanto, para o ciclo do relógio (ou sequência discreta) t_{i+1} , o estado $q(t_{i+1})$, é obtido pela aplicação da função do próximo estado ao estado no início t_i e à entrada de um ciclo de tempo t_i . ou seja:

$$q(t_{i+1}) = f_q(q(t_i), w(t_i))$$

onde w = entrada (input). Em outras palavras, cada estado é alcançado a partir do estado anterior, quando se reconhece um símbolo na entrada da máquina.

5 Máquina de Turing

A Máquina de Turing é um tipo bastante especial de máquina de estados finitos e possui uma estrutura bastante simples, com dois elementos principais: Uma fita (memória da máquina) e um cabeçote, que pode ser abstraída conforme mostra a figura 1:

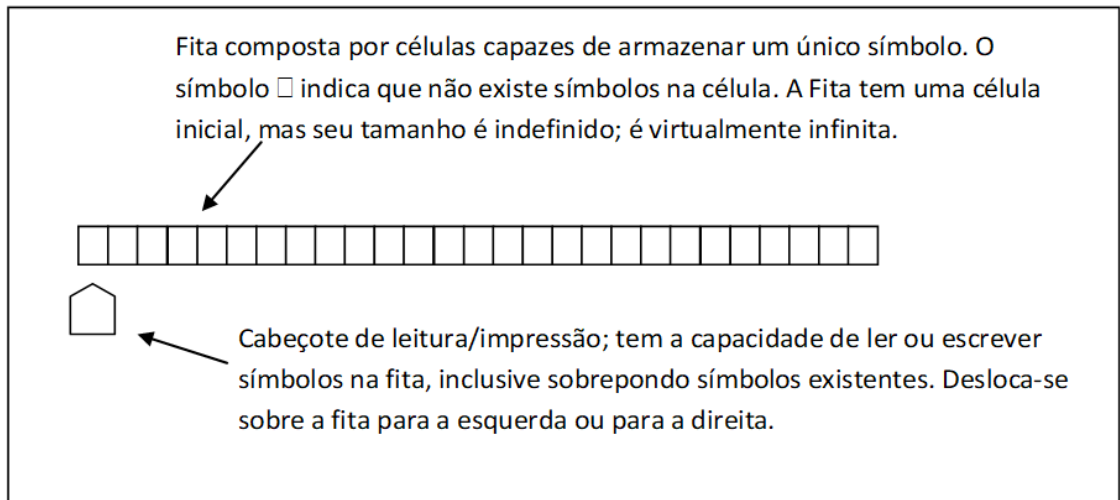


Figura 11 - Estrutura da Máquina de Turing

Define-se mais formalmente uma máquina de Turing, como uma sétupla, $MT = \{Q, \Sigma, \Gamma, q_0, q_{fai}, q_{fri}, \delta\}$ onde:

- Q é um conjunto finito de estados;
- Σ é um alfabeto finito de símbolos;
- Γ é o alfabeto da fita (conjunto finito de símbolos);
- q_0 é o estado inicial; $q_0 \in Q$;
- q_{fai} , com $i = \{1, \dots, n\}$ é o conjunto de estados finais de aceitação, $q_{fai} \in Q$;
- q_{fri} , com $i = \{1, \dots, n\}$ é o conjunto de estados finais de aceitação, $q_{fri} \in Q$;
- δ é uma função parcial chamada função de transição, onde Δ representa o movimento para a esquerda ($\leftarrow$) ou para a direita ($\rightarrow$); admite-se o emprego de um movimento neutro (-). Assim, temos: $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{\Delta\}$.

5.1 A Máquina de Turing como Reconhedora de Linguagens

Uma linguagem aceita ou reconhecida por uma Máquina de Turing é dada pela definição abaixo:

Seja uma Máquina de Turing $MT_{ab} = (Q, \Sigma, \Gamma, \delta, q_0, F)$, e δ' uma sequência de aplicações sucessivas da função programa δ . Então a linguagem reconhecida por M é:

$$L(M) = \{ w \mid w \in \Sigma^+, \delta'(q_0, w) \rightarrow^n q_{fa}, q_{fa} \in F \}$$

Onde Σ^+ , significa que a palavra w é formada por pelo menos um símbolo do alfa-

beto e “ $\rightarrow^n$ ” representa n produções de δ sobre w . $L(M)$ é formada por todas as palavras submetidas à MT que terminam em um estado final de aceitação q_{fa} . Obviamente, uma palavra w seria rejeitada se o processamento pela MT terminasse em um estado final de rejeição q_{fr} , ou seja,

$$\text{NOT}(L(M)) = \{ w \mid w \in \Sigma^+, \delta^*(q_0, w) \rightarrow^n q_{fr}, q_{fr} \in F \}$$

Considere o exemplo da seguinte situação-problema: Para $\Sigma = \{a,b\}$, a Máquina de Turing que aceita $L = \{a^n b^n, n \geq 1\}$ pode ser definida como: $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{a,b\}, \{a,b,x,y,\sim\}, \delta, q_0, \{q_4\})$ com a função programa definida como mostra a Tabela 1:

δ	$Q \times \Gamma$	$Q \times \Gamma \times \{\Delta\}$
δ^1	q_0, a	q_1, x, R
δ^2	q_1, a	q_1, a, R
δ^3	q_1, y	q_1, y, R
δ^4	q_1, b	q_2, y, L
δ^5	q_2, y	q_2, y, L
δ^6	q_2, a	q_2, a, L
δ^7	q_2, x	q_0, x, R
δ^8	q_0, y	q_3, y, R
δ^9	q_3, y	q_3, y, R
δ^{10}	$q_3, \sim$	$q_4, \sim, R$

Table 1- Função Programa da MTab

Descrição do funcionamento:

- A MT inicia seu processamento em q_0 , seu cabeçote aponta para o primeiro símbolo da palavra w .
- Caso o primeiro símbolo seja “a” (esperado), aplica-se a produção estabelecida para a combinação ($Q = q_0$ e $\Gamma = “a”$), marca a posição com um “x” ($\Gamma = a$), muda para o estado q_1 . E o cabeçote se desloca para a direita.
- No algoritmo proposto, o estado q_1 é “responsável” por encontrar um “b” em w ; caso consiga, marca o local da ocorrência de “b” com “y”, o estado muda para q_2 .
- A função do estado q_2 é retroceder na fita até encontrar um x, que representa a marcação da última ocorrência de “a”.
- Em q_2 , encontrando “x”, volta ao estado q_0 , reiniciando o processamento.

- Quando q_0 encontra o “y” significa que já terminou de marcar os símbolos “a”; nesse caso, se não houver mais nenhum “b” para serem marcados, a cadeia está correta; caso contrário, estará errada. Isso é verificado no estado q_3 , que percorre o restante da cadeia.

Essa aplicação da MT para reconhecimento de linguagens demonstra a grande capacidade desse dispositivo, pois a linguagem reconhecida ($L = \{a^n b^n, n \geq 1\}$) representa uma classe especial de linguagens, que não pode ser reconhecida por autômatos muito autômatos.

5.2 A Máquina de Turing como um Transdutor

Um transdutor é um dispositivo que realiza a tradução de uma cadeia de símbolos colocados em sua entrada, noutra cadeia de símbolos em sua saída. Considere uma função $f(x)=y$, onde $x \in X$ (X é o conjunto de símbolos de entrada) e $y \in Y$ (Y é o conjunto de símbolos de saída). Uma função f , com domínio D , é dita ser Turing-Computável ou simplesmente computável se existe alguma Máquina de Turing $M = (Q, \Sigma, \Gamma, q_0, q_{fai}, q_{fri}, \delta,)$ tal que para alguma entrada w ,

$$q_0 w \rightarrow^* q_{fai}$$

Ou seja, para alguma cadeia colocada na entrada da MT, após o seu processamento, ou seja, ao varrer todos os símbolos desta, terminará num estado final de aceitação!

6 Revisão Bibliográfica

Utilizando-se de um simulador para ensinar um determinado conteúdo, pode motivar alunos a se interessarem por um determinado conteúdo que se queira ensinar. LOPES e OLIVEIRA (2013, p.12) indagam que,

[...] nos Estados Unidos da América, mas também no Reino Unido e na Alemanha, os simuladores, e mais recentemente os serious games, têm sido utilizados frequentemente em três áreas de formação: na área militar, na saúde e nos negócios. Os serviços militares foram os primeiros a tirar proveito do uso de simuladores e videojogos, quer como recurso de formação, de terapia ou mesmo como forma para angariar novos recrutas.

E por que não utilizar um simulador para como recurso e/ou para atrair atenção dos alunos para educação? Sendo que são ferramentas poderosas que podem motivar o ensino aos alunos. Para isso, é necessário que os professores estejam abertos a novas propostas de ensino baseados em plataformas digitais, pois, ainda há uma recusa de alguns docentes que ainda insistem em utilizar apenas as metodologias tradicionais. Sobre o baixo uso e mal aceitação dessas ferramentas, PEREIRA(2007, citado LOPES E OLIVEIRA, 2013, p. 12-13) disse que, subsiste uma tendência por parte das editoras nacionais para que os videojogos educativos sejam principalmente vocacionados para os primeiros anos de escolaridade. As editoras consideram, na sua maioria, que a aceitação dos jogos educativos no ensino tem sido “baixa” e que o maior entrave à inclusão destes na sala de aula têm sido os professores. Fato esse que deve ser repensado pelos educadores, assim como disse LOPES e OLIVEIRA (2013, p.16), é fundamental desenvolver esforços para melhor compreender a percepção dos educadores, formadores e docentes sobre o uso destes recursos, quer em ambientes recreativos, quer no processo de ensino-

aprendizagem. PAZIN FILHO e SCARPELINI(2007, p.162) também reforçam precisamente em suas palavras quando dizem que, a simulação é uma técnica de ensino que se fundamenta em princípios do ensino baseado em tarefas e se utiliza da reprodução parcial ou total destas tarefas em um modelo artificial, conceituado como simulador. Sua aplicação é relacionada, em geral, às atividades práticas, que envolvam habilidades manuais ou decisões. Os alunos da disciplina de Teoria da Computação fazem funções programas para resolver determinado problema, mas é difícil ver na prática a execução das funções utilizando lápis e papel, é de fundamental importância ter uma noção mais clara de como está sendo executado, uma vez que um simulador pode dar esta visão aos alunos. É o que complementa FLATO e GUIMARÃES (2011, p. 361), dizendo que, [...] situações “próximas das reais”, o qual se correlaciona com retenção do conhecimento por um tempo mais prolongado e absorção do conteúdo de forma mais agradável e prazerosa do que o ensino usual. Forma agradável essa que pretende-se atingir em utilizar um simulador na disciplina de Teoria da Computação.

7 Desenvolvimento

7.1 Exemplo de aplicação

Como dito anteriormente, é necessário ter noção de como construir uma função programa para utilizar o *software*. Primeiramente deve-se preencher os campos de nome da função(I) e do usuário(II); de uma forma bem simples, é imprescindível ter em mente o que sua função deverá fazer, assim como também definir o alfabeto de entrada(III), conjunto de estados(IV), o estado inicial(VI) que deverá estar contido no conjunto de estados e os possíveis estados finais(V). Depois, é necessário clicar no botão “carregar dados iniciais”.

A interface do software ARCABOUÇO MT apresenta um formulário com os seguintes campos e elementos:

- Nome MT:** Campo de texto com o valor "Função AND" (rotulado I).
- Autor:** Campo de texto com o valor "Diogo" (rotulado II).
- Σ:** Campo de texto com o valor "01" (rotulado III).
- Q:** Campo de texto com o valor "q0q1q2q3q4q5q6qf" (rotulado IV).
- Estado Inicial:** Campo de texto com o valor "q0" (rotulado VI).
- Estados Finais:** Campo de texto com o valor "qf" (rotulado V).
- Botão:** "Carregar dados iniciais" (rotulado V).

Figura 2 – Inserção dos dados iniciais no *software*.

Feito isso, é necessário inserir na tabela(ver figura 3) os dados da função programa que deseja executar. Um exemplo prático e de fácil entendimento é uma função programa que faz uma operação binária de uma operação lógica AND. Deverá inserir linha a linha da função programa; onde primeiro coloca-se um possível estado de leitura(VII), um possível caractere(VIII) que deve estar contido no alfabeto, o estado que permanecerá ou mudará de acordo com sua lógica(IX), o caractere que resultará depois da conversão(X) e por fim, o delta que representa o movimento do cabeçote para a esquerda ou direita(XI).

6 Q Γ $\dashv$ Q Γ Δ

q0 0 $\dashv$ q1 0 > **Load**

Q	Γ	$\dashv$	Q	Γ	Δ

Figura 3 – Inserção na tabela dos dados da função programa.

Adicionado linha a linha e clicando no botão *load*(carregar), logo a tabela estará completa assim como mostra a figura 4. Ainda podendo excluir uma linha ou várias linhas, caso tenha errado algum dos dados inseridos, para isso, tem o botão *del*(deletar).

δ	Q	Γ	$\rightarrow$	Q	Γ	Δ
#			$\rightarrow$			

Q	Γ	$\rightarrow$	Q	Γ	Δ
q0	0	$\rightarrow$	q1	0	$\succ$
q0	1	$\rightarrow$	q2	1	$\succ$
q1	0	$\rightarrow$	q3	0	$\succ$
q1	1	$\rightarrow$	q4	0	$\succ$
q2	0	$\rightarrow$	q5	0	$\succ$
q2	1	$\rightarrow$	q6	1	$\succ$
q3	*	$\rightarrow$	qf	-	$\succ$
q4	*	$\rightarrow$	qf	-	$\succ$

Figura 4 – Tabela preenchida com a função programa.

Na figura 4 observa-se a presença de asteriscos e traços na coluna que contém a letra grega gama(Γ) de leitura e escrita; o asterisco representa um espaço vazio e o traço representa o “não faça nada”, ou seja, se vazio, não faça nada.

Tendo preenchido todos os dados da função programa, pode-se digitar na fita(XIII) abaixo da tabela, uma combinação de dois números que deverão estar contidos no alfabeto. Vejamos um exemplo na figura 5 em que são inseridos números binários definidos anteriormente para que possa fazer a operação. Sendo possível acompanhar quais estados foram percorridos(XII) até chegar ao estado final.

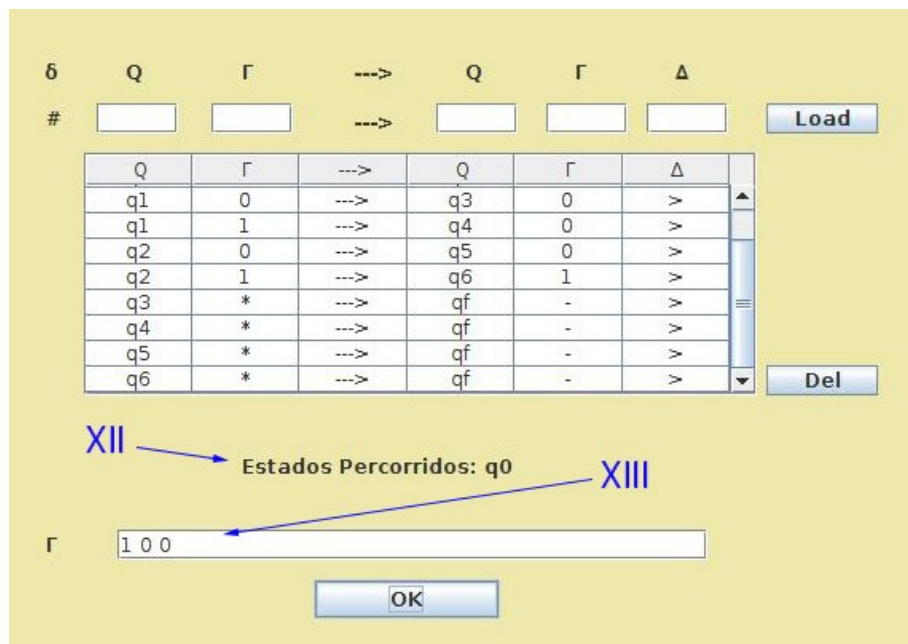


Figura 5 – Execução do simulador.

7.2 Implementação

Para implementação foram utilizados a linguagem de programação *Java*, o ambiente de desenvolvimento *Eclipse* e o *plugin Window Builder* para criação da interface gráfica. Além disso, mais detalhadamente, foram utilizadas expressões regulares para fazer um tratamento dos dados inseridos no campo do alfabeto e estados de uma forma mais rápida e prática; depois de tratados, foi utilizado a estrutura de dados *ArrayList* para armazenagem de tais dados. Já na tabela, cada linha representada no código como um objeto em que é composto por: um estado de leitura, um caractere de leitura, um estado de escrita, um caractere de escrita e um delta. Tais linhas objetos também estão sendo armazenados em uma estrutura de dados *ArrayList*. Na entrada abaixo da tabela que representa a fita, são verificados os caracteres se realmente estão contidos no alfabeto, já que só é possível manipular caracteres que já tenham sido definidos anteriormente no alfabeto.

7.3 Execução

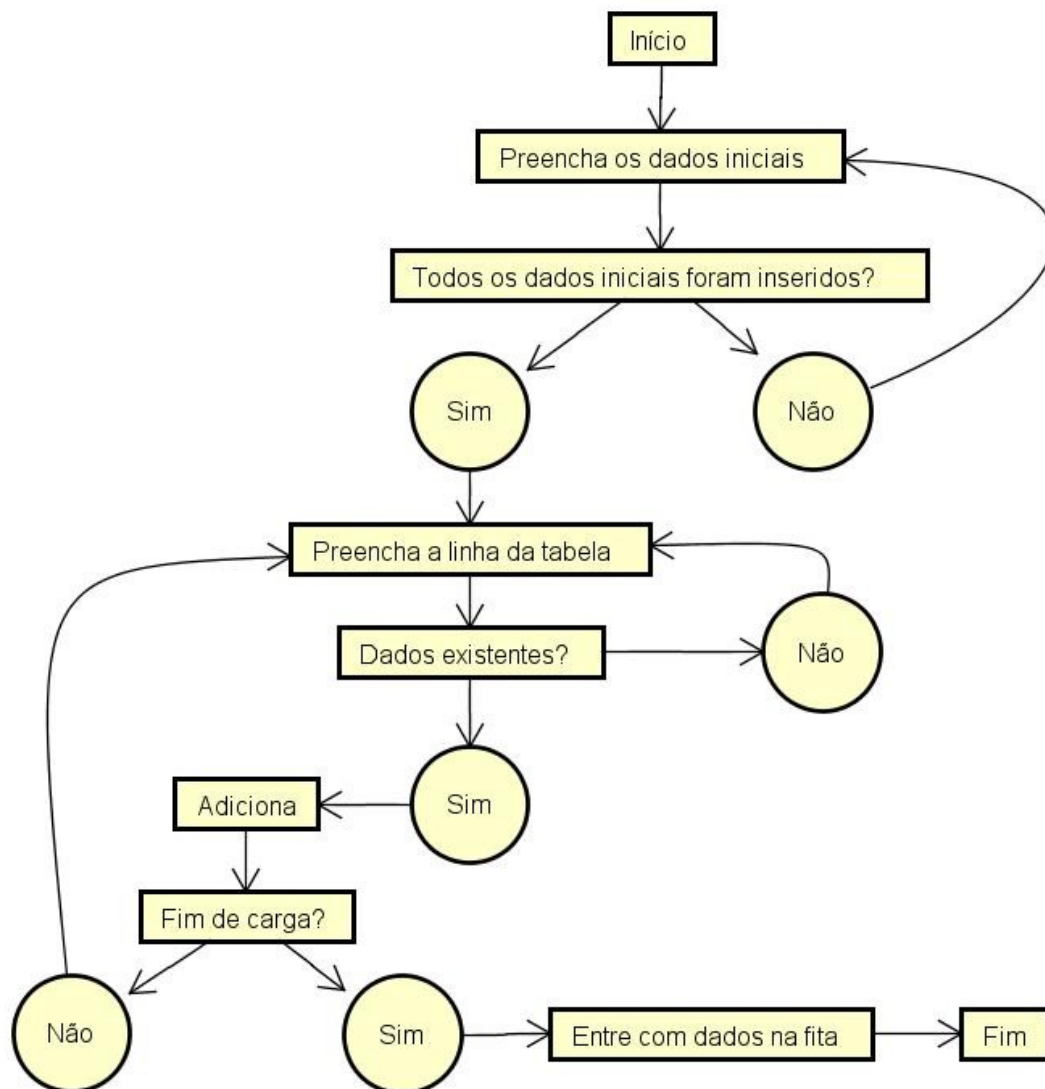


Figura 6 – Fluxo de execução.

8 Trabalhos Relacionados

Aqui serão apresentados trabalhos/projetos que também simulam a Máquina de Turing e têm algo em comum com este trabalho. Salientando que ainda é difícil de encontrar simuladores feitos ou traduzidos para a língua portuguesa, sendo maioria em língua inglesa. Abaixo será apresentado um trabalho relacionado no idioma português e outro inglês.

8.1 Simulador de máquina de turing : uma aplicação web para o aprendizado

Este trabalho também simula a execução da Máquina de Turing. Com o diferencial que é uma aplicação *WEB* (*World Wide Web*), ou seja, é possível executar a aplicação dentro de um navegador de internet. Também, é possível observar uma pequena diferença na sintaxe da construção da função programa, onde os estados são representados por apenas números. Uma coisa bem interessante é que pode-se entrar já com uma função programa que tenha sido feita em um determinado bloco de notas, daí é só colar a função no campo adequado e executar o simulador.

8.2 Turing Machine Simulator

Assim como o simulador que foi descrito anteriormente, este é uma aplicação *WEB* em que toda página é em inglês. Onde têm algumas funções programas já pronta como exemplo para que o usuário que não tenha noção de como construir uma função programa, comece a executar mesmo sem saber como funciona exatamente a execução. Além disso, é possível criar novas funções programas para serem executadas, mas o que dificulta é a sintaxe das funções que o simulador aceita. É uma sintaxe muito confusa, onde se utiliza símbolos como asterisco, exclamação, ponto e vírgula, dois pontos e mais algumas letras. Apesar de ter as instruções de como utilizar, mas acaba sendo complexo para quem esteja aprendendo sobre a criação de MTs.

9 Conclusão

O desenvolvimento do trabalho foi muito proveitoso, pois foi colocado em prática conhecimentos adquirido ao longo do curso de Lic. Ciência da Computação, e em especial, na disciplina de Teoria da Computação, pois o assunto trabalhado em questão é exclusivamente desta disciplina. Acredita-se também, que não seja apenas mais um trabalho de conclusão de curso, mas que possa ser continuado para experimentos, aplicações, adições de funcionalidades e assim gerando trabalhos futuros.

10 Trabalhos futuros

Para trabalhos futuros, pode-se fazer uma refatoração do código, adicionar mais funcionalidade como, por exemplo, um versionamento de funções programa que foi inserida, opção de salvar no computador os dados das funções e aplicar o software em algum grupo de alunos para que possa analisar os dados e verificar se foram satisfatórios ou não.

Referências Bibliográficas

ASHBY, R. A.: *Introdução à Cibernética*. Editora Perspectiva – São Paulo, 1970.

- BALBINOT, Amanda; TIMM, Maria Isabel; ZARO, Milton Antônio. Aplicação de jogos e simuladores como instrumentos para educação e segurança no trânsito. **RENOTE**, v. 7, n. 1, 2009.
- CASSOL, V. J.; LOPES, L.; RIVA, A. D.: **Utilização de Máquina de Turing aplicada a um problema de comparação de Listas de Palavras** - Revista Brasileira de Computação Aplicada (ISSN 2176-6649), Passo Fundo, v. 2, n. 2, p. 66-80, set. 2010
- GILL, A., : **Introduction to the Theory of Finite-state Machines**. McGraw-Hill, 1962.
- LOPES, Nuno; OLIVEIRA, Isolina. Videojogos, Serious Games e Simuladores na Educação: usar, criar e modificar. **Educação, Formação & Tecnologias-ISSN 1646-933X**, v. 6, n. 1, p. 4-20, 2013.
- PAZIN FILHO, Antonio; SCARPELINI, Sandro. Simulação: definição. **Medicina (Ribeirão Preto. Online)**, v. 40, n. 2, p. 162-166, 2007.
- SIPSER, M.: **Introduction to the Theory of Computation**, Thomson Course Technology Ed., Boston, MS, 2006.
- TEIXEIRA, João F.:**Mentes e máquinas** – Uma Introdução à Ciência Cognitiva. Ed. Artes Médicas, São Paulo, 1998.
- FLATO, Uri Adrian Prync; GUIMARÃES, Helio Penna. Educação baseada em simulação em medicina de urgência e emergência: a arte imita a vida. **Rev Bras Clin Med**, v. 9, n. 5, p. 360-4, 2011.
- PINTO, Wellmmer Lucas de Oliveira. **Simulador de máquina de turing**: uma aplicação web para o aprendizado. 2016. 92 f. São Paulo.
- MORPHETT, Anthony. **Turing machine simulator**. Disponível em: <<http://morphett.info/turing/turing.html>> Acesso em 14 maio de 2018.