

# Um *merge* entre Máquina de Turing e Operações Matemáticas em Binário no Ensino de Linguagens Formais e Autômatos

**Maurilio Martins Campano Junior**

Programa de Pós-Graduação em  
Ciência da Computação da  
Universidade Estadual de  
Londrina  
Londrina-Paraná, Brasil  
maurilio.campanojr@gmail.com

**Carolinne Roque e Faria**

Programa de Pós-Graduação em  
Ciência da Computação da  
Universidade Estadual de  
Londrina  
Londrina-Paraná, Brasil  
carolinne.rf@outlook.com

**Cinthyana Renata Sachs Camerlengo de Barbosa**

Programa de Pós-Graduação em  
Ciência da Computação da  
Universidade Estadual de  
Londrina  
Londrina-Paraná, Brasil  
cinthyana@uel.br

**Alan Salvany Felinto**

Programa de Pós-Graduação em  
Ciência da Computação da  
Universidade Estadual de  
Londrina  
Londrina-Paraná, Brasil  
alan@uel.br

## RESUMO

Este trabalho propõe a utilização de operações da aritmética computacional no ensino de conceitos de Máquina de Turing. A máquina que foi desenvolvida recebe como entrada dois números em binário e gera como saída o resultado da adição desses números, permitindo assim por meio de um recurso didático multidisciplinar auxiliar no aprendizado das operações matemáticas em binário. Foi feita uma avaliação do *software* para operações de números binários utilizando uma Máquina de Turing por 29 alunos na disciplina de Linguagens Formais, Autômatos e Computabilidade para o segundo ano do curso de Ciência da Computação da Universidade Estadual de Maringá no mesmo momento que as operações com números binários foram utilizadas na matéria de Arquitetura e Organização de Computadores para a mesma turma e curso.

## Palavras-Chave

Máquina de Turing; Linguagens Formais e Autômatos;

Paste the appropriate copyright/license statement here. ACM now supports three different publication options:

- ACM copyright: ACM holds the copyright on the work. This is the historical approach.
- License: The author(s) retain copyright, but ACM receives an exclusive publication license.
- Open Access: The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.

This text field is large enough to hold the appropriate release statement assuming it is single-spaced in Times New Roman 8-point font. Please do not change or modify the size of this text box.

Each submission will be assigned a DOI string to be included here.

Números Binários.

## ABSTRACT

This paper proposes the use of computational arithmetic operations in the teaching of Turing Machine concepts. The developed machine receives two numbers in binary for entry and generates the result of adding these numbers, allowing by a didactic multidisciplinary resource helping in learning process of mathematical operations in binary. The software was evaluated by 29 students in the Formal Language, Automata and Computability discipline for a second year of the Computer Science course at Maringá State University at the same time that binary number operations were used in Architecture and Organization of Computers for the same class and course.

## Author Keywords

Turing Machine; Formal Languages and Automata; Binary Numbers.

## ACM Classification Keywords

CCS → Theory of computation → Formal languages and automata theory → Formalisms → Rewrite systems.

## INTRODUÇÃO

O “pai da Computação”, Alan Turing, propôs em seu artigo “*Computing Machinery and Intelligence*” [1] um modelo de máquina digital capaz de realizar qualquer operação. Esse modelo, a Máquina de Turing, seria reconhecido mais tarde como a base para o que temos hoje no mundo da computação [2] [3].

Outros modelos de máquinas chegaram a ser desenvolvidos, como por exemplo, a Máquina Norma e Máquina Post [4], porém tais modelos provaram-se ser no máximo equivalente a Máquina de Turing e nunca superior. Essa é a então conhecida Tese de Church-Turing [4].

Na literatura relacionada às Linguagens Formais e Autômatos [5], [6], [7], [4] os conceitos de Máquinas de Turing são apresentados normalmente usando exemplos de linguagens com “a” e “b”, ou com números binários, porém são raros exemplos de linguagens mais complexas.

O presente trabalho tem como objetivo apresentar uma Máquina de Turing Transdutora que tenha como entrada dois números binários e gere como saída a soma desses dois números. Tal máquina apresenta uma alta complexidade devido à quantidade de estados e transições, porém seu uso no ensino tanto de conceitos de Máquina de Turing, quanto das operações matemáticas em binário pode ser um elemento extra em aulas práticas.

**FUNDAMENTOS TEÓRICOS**

Esta seção é descrita em dois subtópicos, sendo a 2.1 para falar sobre a representação de números binários e a operação de adição de dois números binários e a seção 2.2 sobre a representação de Máquinas de Turing.

**Números binários e adição em números binários**

Os números binários são utilizados para representar dados em um meio digital, como por exemplo, a representação no meio analógico com presença ou ausência de carga elétrica e no meio digital por meio de zeros e uns. Essa representação com dois símbolos utiliza-se da mesma técnica do telégrafo, que transmitia mensagens por código Morse, sendo os símbolos curto e longo análogos ao zero e um [8].

Utilizando-se a notação binária é possível representar uma faixa de valores diferentes de acordo com a quantidade de bits. Por exemplo, com dois bits pode-se representar quatro valores distintos, sendo eles 00, 01, 10 e 11. Ou seja, com n bits, podemos representar 2<sup>n</sup> valores distintos.

Para a notação de números inteiros usando a base binária de zeros e uns, podemos representar os números utilizando as seguintes representações: de binário puro, de binários em sinal magnitude e a representação em complemento de 2 [8].

Tomando como base a representação de números inteiros na base binária pura, que também é a representação utilizada neste trabalho, pode-se observar na Tabela 1, que com quatro bits temos as seguintes possibilidades para números inteiros.

Binário	Decimal	Binário	Decimal
0000	0	1000	8
0001	1	1001	9
0010	2	1010	10
0011	3	1011	11
0100	4	1100	12
0101	5	1101	13
0110	6	1110	14
0111	7	1111	15

**Tabela 1. Comparação da base binário de 4 bits com base decimal**

As operações matemáticas de adição e subtração feitas na base binária seguem as mesmas regras da base decimal, contando com a diferença que temos apenas dois dígitos. Para a adição de dois números, temos quatro possibilidades de valores, sendo elas: 0 + 0, 0 + 1, 1 + 0, e 1 + 1. As três primeiras têm os mesmos resultados de uma operação em decimal, já para a operação de 1 + 1 temos como resultado zero, gerando um “vai um” para a coluna da esquerda [8].

**Máquina de Turing**

Turing descreve um computador digital como sendo formado por: uma unidade de armazenamento, uma unidade de execução e uma unidade de controle. A unidade de armazenamento é formada por uma fita, dividida em células, com um cabeçote apontando para a célula atual, a qual pode ser lida/escrita de acordo com a unidade de execução. Por sua vez, a unidade de execução tem como objetivo fazer a leitura do caractere representado na célula atual, analisar o que deve ser feito e alterar quando necessário. Já a unidade de controle faz as movimentações do cabeçote de acordo com o que a unidade de execução deseja, movendo o cabeçote para esquerda ou direita [1].

Em uma representação mais formal ou mais matemática, temos que uma Máquina de Turing (MT) é representada por uma óptupla.  $MT = \{\Sigma, E, i, F, \Gamma, \square, <, \delta\}$ , sendo  $\Sigma$  o conjunto de símbolos do alfabeto de entrada da fita,  $E$  o conjunto de estados da máquina,  $i$  é o estado inicial,  $F$  o conjunto de estados finais ou estados de aceitação,  $\Gamma$  é o conjunto de símbolos auxiliares da fita,  $\square$  é o símbolo de branco,  $<$  é o delimitador de início de fita e o  $\delta$  representa a função de transição, a qual indica que a partir de um estado, com um símbolo dos alfabetos da fita ou auxiliar, muda-se de estado, escreve algum símbolo do alfabeto da fita ou auxiliar e move-se o cabeçote para direita ou esquerda, representada assim por  $\delta: \{E \times (\Sigma \cup \Gamma) \rightarrow E \times (\Sigma \cup \Gamma) \times (\text{direita} \mid \text{esquerda})\}$  [5].

As Máquinas de Turing ainda podem ser divididas em dois tipos: as reconhecedoras e as transdutoras. As máquinas reconhecedoras respondem sim ou não para a palavra de entrada, de acordo com a linguagem que a MT reconhece, ou seja, a saída de uma MT reconhecedora é dizer se a palavra de entrada pertence ou não à linguagem que a MT reconhece. Já as MT transdutoras recebem uma palavra de entrada e a saída pode ser a palavra alterada de acordo com algum critério, ou seja, a saída de uma MT transdutora está na própria fita da máquina [4], [5], [9].

Um exemplo de MT transdutora pode ser visualizado na seção 4. Já um exemplo de MT reconhecedora pode ser visto na Figura 1. A MT reconhece palavras sobre a linguagem  $L = \{a^n b^n / n > 0\}$ , ou seja, as quais a primeira metade contém somente símbolos a's e a segunda metade contém exatamente a mesma quantidade de símbolos b's. Esse tipo de linguagem é chamado de duplo-bal, relacionando com o duplo balanceamento de símbolos, mesma quantidade de a's e b's. Esse conceito muito utilizado na construção de compiladores [4], [5], [7], [9].

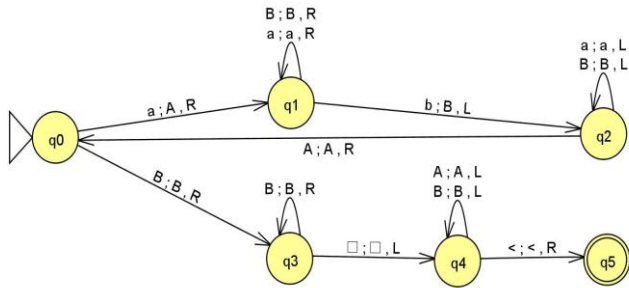


Figura 1. Exemplo de MT reconhecedora para a linguagem  $L = \{a^n b^n / n > 0\}$

A descrição formal da MT pode ser representada pelo alfabeto  $\Sigma = \{a,b\}$ , o conjunto de estados  $E = \{q_0, q_1, q_2, q_3, q_4, q_5\}$ , o estado inicial  $q_0$ , o conjunto de estados finais  $F = \{q_5\}$ , o alfabeto auxiliar  $\Gamma = \{<, \square\}$ , sendo  $<$  o marcador de início de fita,  $\square$  o símbolo de branco, e por fim a tabela de transição, representada pela Tabela 2.

	a	b	A	B	<	$\square$
q0	q1,A,R			q3,B,R		
q1	q1,a,R	q2,B,L		q1,B,R		
q2	q2,a,L		q0,A,R	q2,B,L		
q3				q3,B,R		q4, $\square$ ,L
q4			q4,A,L	q4,B,L	q5,<,R	
q5						

Tabela 2. Função de transição para a Figura 1

### TRABALHOS RELACIONADOS

Em livros didáticos para o ensino de linguagens formais e autômatos [4], [5], [6], [7], [9] nos tópicos de construção de

autômatos, vários exemplos de operações com números binários podem ser visualizados. Em [5] demonstra-se como reconhecer com um autômato finito determinístico todos os números binários de tamanho par, bem como reconhecer qualquer número binário na qual a quantidade de símbolos zeros é par e a quantidade de símbolos uns também é zero, exemplo no qual pode ser visualizado na Figura 2, sendo o estado “pp” o estado final, para as palavras da linguagem descrita acima.

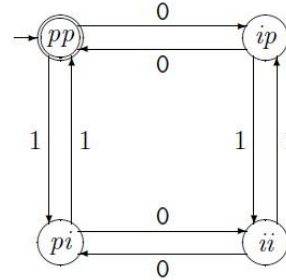


Figura 1. Exemplo de AFD de Vieira [Vieira 2006]

Ainda em [5] é proposto um exercício interessante com relação às Máquinas de Turing, conforme descrito abaixo:

“Construa uma MT padrão que gere todos os números naturais em notação binária. A MT não deve parar nunca. Ela deve gerar 0, depois 1, depois 10, etc., separados por branco, indefinidamente. A fita deverá ficar assim:

$\langle 0 \square 1 \square 10 \square 11 \square 100 \square 101 \square 110 \square \dots \rangle$

A resolução do exercício pode ser visualizada na Figura 5 abaixo.

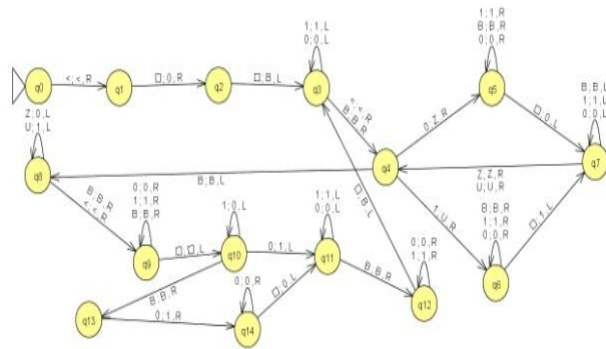


Figura 2. Resolução de exercício proposto por Vieira (2006)

Já Sipser [6] não apresenta exercícios de MT sobre operações matemáticas com números binários, porém faz exemplos de MT que decidem se a entrada contém números binários diferentes, sendo a entrada do tipo “ $\langle x_1 \# x_2 \# x_3 \dots \rangle$ ”, na qual  $x_1, x_2, x_3$  representa os números binários.

Hopcroft, Ullman e Motwani [7] detalha a construção de MT que aceitam linguagens sobre o alfabeto binário, porém nenhum exemplo relacionando operações matemáticas com a linguagem de números binários.

Menezes [4, 9] propõe uma série de exercícios de desenvolvimento de MT, porém utilizando em sua maioria o alfabeto “a” e “b” em vez do alfabeto binário, podendo ser modificado para atender ao alfabeto binário, contudo não mostrando assim operações matemáticas com o alfabeto.

Os trabalhos científicos que envolvem o uso dos mecanismos reconhecedores, como autômatos finitos determinísticos e não determinísticos, autômatos de pilha e máquinas de Turing, normalmente envolvem a construção de simuladores para o ensino. Como exemplo podemos citar os trabalhos de [10], [11], [12], [13], [14], [15], [16], [17], tendo cada simulador suas vantagens e desvantagens, porém todos sendo úteis para o ensino e aprendizagem de Linguagens Formais e Autômatos e seus mecanismos reconhecedores.

Além disso, vários trabalhos discutem sobre a importância do ensino de Linguagens Formais e Autômatos, bem como de metodologias voltadas para o ensino dos tópicos da matéria no ensino superior [18], [19], [20], [21], [22], [23] e todos esses ressaltam a importância da visualização gráfica das transições de um autômato no aprendizado.

Os exemplos descritos neste trabalho foram implementados utilizando o simulador JFLAP [12], o qual apresenta uma interface onde são inseridos os estados e as transições do autômato e também podem ser testadas palavras para verificar se a saída do autômato está correta. O JFLAP possui como vantagem a sua fácil interface de desenvolvimento dos autômatos, com diversos recursos adicionais, porém para usuários com pouco conhecimento nos autômatos, o simulador pode não ser o mais intuitivo se comparado com outros simuladores como LFapp ou Automata Tutor.

### MÁQUINA DE TURING PARA SOMA DE NÚMEROS BINÁRIOS

O exemplo de MT faz a soma de dois números binários quaisquer (Figura 4) e funciona de modo que a máquina recebe como entrada dois números no formato “<número\_binário1#número\_binário2”, sendo o símbolo “<” o marcador de início da fita e o símbolo “#” para delimitar o fim do primeiro número e início do segundo. A visualização gráfica da Máquina de Turing tem um alto grau de complexidade, uma vez que a máquina resultante para essa operação é formada por mais de oitenta estados e mais de duzentas transições.

O funcionamento da MT pode ser visualizado na Figura 5, sendo que primeiramente é inserido o símbolo “#” após o segundo número, que delimitará o início do resultado. Após isso é verificada a quantidade de bits de cada número e marcado essa mesma quantidade de bits somada com um, para o número resultante. Finalizada a etapa de marcar a quantidade de bits de cada número e do número resultante, os números são somados, começando pelo bit mais a direita do primeiro número e com o bit mais a direita do segundo

número, sendo o resultado colocado no bit mais a direita do resultado.

Para verificar a quantidade de bits do número resultante, são analisados os bits dos dois números um a um, sendo um ciclo de buscar um bit no primeiro número e depois outro no segundo. Quando em alguma dessa busca, um bit não for encontrado indica que o outro número tem uma quantidade maior de bits. Isso deve ser feito, pois uma adição de dois números é iniciada pelo bit mais à direita, porém na MT devemos saber quantos bits têm o número resultante e começar a soma também pelo bit mais à direita, sendo essa a etapa de marcar bits do número resultante, e por fim somar os dois números, e colocar o resultando. Como pode ser visualizada na Figura 5, a análise dos bits de um número altera o próprio número para letras (maiúsculas e minúsculas), com o objetivo de fazer a contagem dos bits. Assim o algoritmo de soma por MT sabe diferenciar as letras, considerando que as letras “a” e “x” representam o número zero, enquanto as letras “b” e “y” representam o número um tanto em maiúsculas quanto minúsculas.



Figura 3. Funcionamento da MT que soma dois números binários

O aprendizado do funcionamento da MT pode ser facilitado no JFLAP utilizando-se da visualização passo a passo, na qual é possível ver o processamento de uma palavra e qual é o estado ativo no momento do processamento.

Na Figura 5 é representado o processamento da entrada “<100#111”, sendo que os números foram transformados em letras segundo o algoritmo de processamento da máquina, de modo que o número zero pode ser representado por “a” ou “x”, e o número um pode ser representado por “b” ou “y”, ambos em maiúsculo ou minúsculo, o caractere “Z” representa o resultado da soma dos dois números e números zero ou um representam o resultado da soma até o momento.

Ainda na Figura 5, em (a) podemos visualizar o estado atual da MT, representando que a partir do estado “q27” temos uma transição para o estado “q23” que altera o símbolo “Z”



para “1”, e move o cabeçote da máquina para a esquerda, como pode ser visualizado em (b). Essa operação indica que a máquina somou o símbolo “a” do primeiro número com “y” do segundo número, ou no caso 0 somado com 1 e o resultado foi escrito no lugar do símbolo “Z” mais à direita, sendo o símbolo 1 representado na Figura 5 (b).

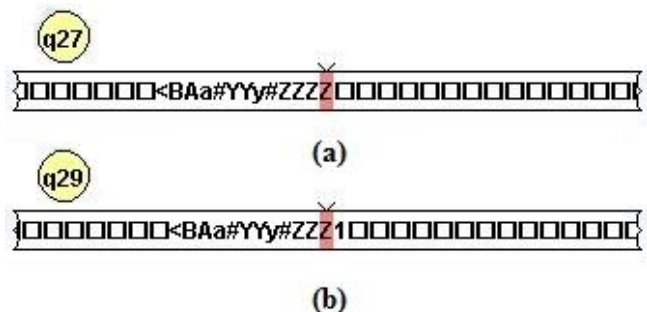


Figura 4. Processamento da entrada “<100#111” na MT, sendo (a) o estado anterior a soma de dois bits, e (b) o estado posterior à soma

A descrição formal da MT de somar números binários, como dita anteriormente, tem uma complexidade muito alta devido à quantidade de estados (mais de 80) e quantidade de transições (mais de 200).

Nessa MT temos como alfabeto auxiliar os símbolos  $\Gamma = \{<, \square, \#, X, Y, A, B, x, y, a, b, Z\}$ , sendo os símbolos X, Y, A, B, x, y, a, b, Z utilizados nas etapas intermediárias da soma dos números.

Os testes realizados para esta MT podem ser visualizados na Figura 5, sendo que a primeira linha exemplifica a soma de dois mais dois, mostrando o resultado correto e os demais exemplos mostram somas de números apresentados na coluna “Input” e o resultado sendo apresentado na coluna “Output”.

Input	Output	Result
<10#10	10#10#100	Accept
<111#111	111#111#1110	Accept
<100000#01010	100000#01010#01001010	Accept
<011#0001	011#0001#00100	Accept
<1#111	1#111#1000	Accept

Figura 5. Resultados da MT de soma de números binários apresentados no JFLAP

O uso do simulador JFLAP e a MT que realiza a soma de números binários será utilizada como auxílio à aprendizagem do funcionamento da Máquina de Turing, demonstrando como operações complexas podem ser resolvidas utilizando a MT e visualizadas por meio de um simulador para melhorar o desempenho dos estudantes na matéria de Linguagens Formais e Autômatos.

Na Figura 6 encontram-se as avaliações do software para operações de números binários utilizando uma Máquina de Turing de 29 alunos na disciplina de Linguagens Formais,

Autômatos e Computabilidade para o segundo ano do curso de Ciência da Computação da Universidade, enquanto que operações com números binários foram utilizadas na matéria de Arquitetura e Organização de Computadores para a mesma turma e curso.

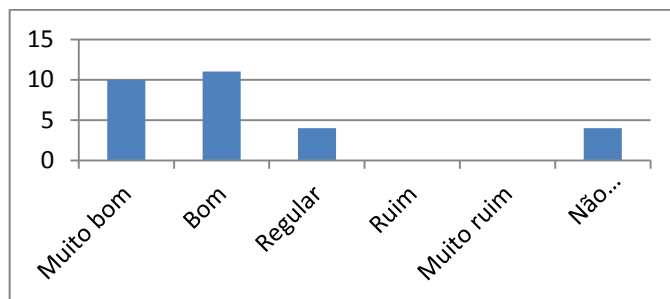


Figura 6. Avaliações dos Alunos sobre o Software para Operações de Números Binários utilizando Máquina de Turing

### CONCLUSÕES

A área da Computação é uma área que exige um alto nível de abstração dos alunos na compreensão dos conteúdos didáticos, em especial a área da Teoria da Computação. Assim o ensino desses conteúdos deve utilizar de metodologias e práticas em salas de aula e laboratórios para aproximar cada vez mais o aluno desse conteúdo teórico, que muitas vezes são bem complexos a ele. Outros conteúdos podem ser mais fáceis de serem aprendidos pela semelhança com conteúdos mais básicos, como a matemática, a aritmética computacional que envolve as mesmas operações matemáticas simples, porém com números binários.

Assim, este trabalho visa a multidisciplinariedade no ensino da Computação, aliando conceitos da aritmética computacional e de Máquina de Turing, criando-se assim uma Máquina de Turing que realiza a soma de dois números binários quaisquer. O exemplo deste trabalho pode ser utilizado tanto na matéria de aritmética computacional quanto no ensino de Máquina de Turing, aliando cada objetivo de aprendizado com os conceitos abordados na máquina em si, utilizando de recursos didáticos auxiliares, como a simulação e a visualização gráfica da máquina e da soma dos números.

Como trabalhos futuros pretende-se também implementar outras Máquinas de Turing para operações binárias mais complexas e agrupando essas em uma única máquina que seria capaz de calcular qualquer uma das quatro operações básicas da matemática, resultando em uma calculadora que se utiliza a Máquina de Turing como ferramenta.

### REFERÊNCIAS

- Alan Turing. 1950. *Computing Machinery and Intelligence*. Mind, LIX, 236: 433-460,
- Alan Turing. 1937. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, Ser. 2, 42.

3. Alan Turing. 1937. Computability and  $\lambda$ -definability. *Journal of Symbolic Logic*, 2, 4: 153–163.
4. Paulo B. Menezes e Tiaraju. A. Diverio. 2011. *Teoria da Computação: Máquinas Universais e Computabilidade* (3ª ed.). Porto Alegre: Bookman.
5. Newton. J. Vieira. 2006. *Introdução aos Fundamentos da Computação*. Pioneira Thomson Learning.
6. Michal Sipser. 2007. *Introdução à Teoria da Computação*. (2ª ed.). Cengage Learning.
7. John. E. Hopcroft, Jeffrey. D. Ullman, Rajeev e Motwani. 2002. *Introdução à Teoria de Autômatos, Linguagens e Computação*. Rio de Janeiro: Campus.
8. Ronald. J. Tocci, Neal. S. Widmer e Gregory L. Moss. 2011. *Sistemas Digitais: Princípios e Aplicações*. (11ª ed.). Pearson.
9. Paulo B Menezes. 2011. *Linguagens Formais e Autômatos* (6ª ed.). Porto Alegre: Artmed.
10. Juventino. F. L. Neto e Ricardo Terra. 2015. *LFApp: um Aplicativo Móvel para o Ensino de Linguagens Formais e Autômatos*. Conclusion Paper, Universidade Federal de Lavras.
11. Juventino F. L. Neto e Ricardo Terra. 2016. LFApp: Um Aplicativo Móvel para o Ensino de Linguagens Formais e Autômatos. In *Anais do 24º Workshop sobre Educação em Computação* (WEI'16), Porto Alegre: SBC, 2196-2205.
12. JFLAP Home Page. Recuperado Janeiro 15, 2019 de <http://www.jflap.org/>
13. Jody Paul. 2015. Using JFLAP to engage students and improve learning of computer science theory: tutorial presentation, *Journal of Computing Sciences in Colleges*, 31(2), 145-148.
14. Carlos. H. Pereira and Ricardo Terra. 2018. A mobile app for teaching formal languages and automata. *Computer Applications in Engineering Education*. 26, 1742–1752.
15. Rajeev Alur, Loris D'Antoni, Sumit Gulwani, Dileep Kini and Mahesh Viswanathan. 2013. Automated grading of DFA constructions. In *Proceedings of International Joint Conference on Artificial Intelligence* (IJCAI'13), 1976-1982.
16. Loris D'Antoni, Weavery, M., Weinert, A. and Alur, R. 2015. Automata Tutor and what we learned from building an online teaching tool. *Bulletin of the EATCS*, 117. doi: 10.1145/2723163
17. Loris D'Antoni Dileep Kini., Rajeev Alur, Sumit Gulwani, Mahesh Viswanathan and Björn Hartmann. 2015. How Can Automatic Feedback Help Students Construct Automata? *ACM Transactions on Computer-Human Interaction*. 2, 2. doi: 10.1145/2723163
18. Isabel Cafezeiro e Leonardo Costa. 2016. Modos contemporâneos de aprendizado e construção do conhecimento: reflexões sobre o ensino de Teoria da Computação para Sistemas de Informação. In *Anais do Workshop em Educação em Informática* (WEI'16), Porto Alegre: SBC, 2245-2254. doi: 10.22533/at.ed.46919160111
19. Luiz O. R. Gavaza, Lais. N. Salvador e, David M. B. Santos. 2017. Uma experiência de aplicação de uma abordagem baseada em problemas no ensino de Teoria da Computação em sala de aula tradicional. In *Anais do 25º Workshop sobre Educação em Computação* (WEI'17), São Paulo: SBC, 2257-2266.
20. Luiz O. R. Gavaza, Lais. N. Salvador e David M. B. Santos. 2018. Percepção de estudantes sobre motivação e aprendizagem em Teoria da Computação com PBL. In *Anais do 26º Workshop sobre Educação em Computação* (WEI'18), Natal: SBC, 125-134.
21. Juliana. P. C. Pirovani e Guilherme. V. Mataveli. 2013. Estudo e adaptação de software para o ensino de Linguagens Formais e Autômatos. *Revista Brasileira de Informática na Educação*. 21,3: 53-68. doi: 10.5753/RBIE.2013.21.03.53
22. Icaro Souza, Ecivaldo Matos, Débora Abdalla e Ranansamir da Silva. 2016. Recursos Computacionais Para Suporte Ao Ensino De Teoria Da Computação, Linguagens Formais E Autômatos. In *Anais do 24º Workshop sobre Educação em Computação* (WEI'16), Porto Alegre: SBC, 2373-2382.
23. Icaro Souza, Ecivaldo Matos e Débora Abdalla. 2016. Abordagens Metodológicas para Ensino de Teoria da Computação, Linguagens Formais e Autômatos. In *Anais da 1ª Escola de Informática Teórica e Métodos Formais* (ETMF'16), Natal. v.1, 25-134.