

# Ensino de Teoria da Complexidade

Alex Luciano Roesler Rese, Rafael de Santiago

Curso de Ciência da Computação – Universidade do Vale do Itajaí (UNIVALI)  
Caixa Postal 360 – CEP 88302-202 – Itajaí – SC – Brasil

{alex\_rese, rsantiago}@univali.br

**Abstract.** *There are several papers in the literature studying the problem of low interest student in subjects related to the Theory of Computing. These studies agree that the cause of this problem is the presence of only theoretical content. The alternative is the use of appointed teaching materials aimed at the practice of content. As sub-area of Theoretical Computer Science, is the Theory of Complexity, which aims to study the problems that makes them easy or difficult to be solved by computer. In this context, we present a survey to facilitate the teaching of Complexity Theory through practical activities and assess whether these are more efficient as the students' learning. Four materials were developed and tested, with the students through a questionnaire and accompanying performance notes. With the results, there is evidence that learning has benefited.*

**Keywords:** *Computer Education. Teaching Theory of Computation. Computational Complexity Theory.*

**Resumo.** *Existem diversos trabalhos na literatura que estudam o problema do baixo interesse dos alunos por disciplinas relacionadas à Teoria da Computação. Estes trabalhos concordam que a causa deste problema é a presença de conteúdos exclusivamente teóricos. A alternativa apontada é o uso de materiais de ensino que visam a prática dos conteúdos. Como subárea da Teoria da Computação, encontra-se a Teoria da Complexidade, que visa o estudo do que faz alguns problemas serem fáceis ou difíceis de serem resolvidos pelo computador. Neste contexto, apresenta-se uma pesquisa para viabilizar o ensino de Teoria da Complexidade por meio de atividades práticas e avaliar se estas são mais eficientes na aprendizagem dos alunos. Quatro materiais foram desenvolvidos e experimentados com os alunos participantes, por meio de questionário e acompanhamento de desempenho de notas. Com os resultados obtidos, há evidências de que a aprendizagem foi beneficiada.*

**Palavras-Chave:** *Ensino de Computação. Ensino de Teoria da Computação. Teoria da Complexidade Computacional.*

## 1. Introdução

No ensino relacionado a Ciência da Computação, existem disciplinas que possuem uma forte característica desafiadora, que exigem dedicação extra aula dos alunos. O

desinteresse por elas pode causar altos índices de reprovação e nível de conhecimento baixo dos egressos em disciplinas fundamentais.

Uma das disciplinas que sofre estes problemas é a Teoria da Computação, que é fundamental para qualquer currículo de graduação e pós-graduação na área de Ciência da Computação, MEC (2009). Existem diversos trabalhos na literatura que reportam pesquisas realizadas sobre este problema e possíveis causas. Alguns destes são destacados a seguir:

- Devedzic, Debenham e Popovic (2000) indicam que o maior problema para o ensino Teoria da Computação é a quantidade de abstrações dos conteúdos abordados;
- De acordo com Cavalcante, Finley e Rodger (2004) e Cogliati, Goosey, Grinder, Pascoe e Ross (2005), a Teoria da Computação é uma área que normalmente possui a necessidade de uso de técnicas de visualização dos conteúdos. Materiais de ensino com estas características para a disciplina são escassos. A visualização dos conteúdos é apresentada nestes trabalhos como uma forma de praticar o conteúdo e de permitir ao aluno vivenciar aspectos teóricos;
- Sipser (2007) destaca que, como em qualquer aprendizado matemático, a prática é fundamental. Por ser uma disciplina com aspectos teóricos, muitos alunos possuem dificuldades em fixar os conteúdos. Esta mesma preocupação com a prática é citada e considerada em Ramos, José Neto e Vega (2009). Chesñevar, Cobo e Yurcik (2003) identificam a necessidade da prática e acrescentam que o uso de softwares simuladores auxiliam para resolver este problema; e
- Goyal e Sachdeva (2009) concordam com a ausência da prática dos conteúdos e afirmam que a utilização de técnicas visuais, sugeridas por Cavalcante, Finley e Rodger (2004) e Cogliati, Goosey, Grinder, Pascoe e Ross (2005), são importantes, mas incompletas. Segundo eles, deve-se utilizar os conceitos da Teoria da Computação de forma interdisciplinar nos cursos de computação e engenharia.

Considerando a literatura destacada, observa-se a convergência para um único problema: a de como ensinar de forma prática os conteúdos de Teoria da Computação, pois os autores destacados concordam de que o interesse dos alunos é despertado a partir do uso de atividades práticas.

Segundo Sipser (2007), há três áreas tradicionalmente centrais à Teoria da Computação: Teoria de Autômatos, Teoria da Computabilidade e Teoria da Complexidade. A Teoria da Complexidade, como subárea, está relacionada aos estudos das causas do que faz um problema ser computacionalmente difícil de resolver ou não.

Na Teoria da Complexidade podemos destacar quatro temas fundamentais: análise da complexidade (perspectivas otimista, média e pessimista), complexidade do problema, classes de complexidade e algoritmos aproximados. Estes tópicos envolvem matemática (inclusive probabilidade) e outros conceitos, técnicas e teoremas da Teoria dos Autômatos e da Computabilidade (subáreas da Teoria da Computação). Para ilustrar o amplo conteúdo desta área, somente no tema sobre as classes de complexidade, Aaronson (2011) cita 417 classes diferentes, que caracterizam problemas de acordo com outros com complexidade semelhante.

A Teoria da Complexidade sofre os mesmos problemas citados anteriormente na aprendizagem de Teoria da Computação e é alvo do trabalho relatado por este artigo. Neste contexto, duas perguntas de pesquisa foram levantadas na proposta do trabalho:

- Como pode-se viabilizar a prática de conteúdos da Teoria da Complexidade?
- Viabilizar a prática auxiliará o aluno na aprendizagem de Teoria da Complexidade?

Este trabalho teve por objetivo realizar um estudo sobre quais temas podem ser abordados para a prática dos conteúdos e a criação de materiais instrucionais. Estes conteúdos foram desenvolvidos com a tecnologia Flex e criados sob o contexto da disciplina de Complexidade de Algoritmos do curso de Ciência da Computação da Fundação Universidade do Vale do Itajaí.

Para demonstrar os esforços de pesquisa realizados, dividiu-se este em trabalhos relacionados, ensino de complexidade na UNIVALI, materiais produzidos, resultados e conclusões.

## **2. Trabalhos Relacionados**

Muitos trabalhos relatam dificuldades no ensino de Teoria da Computação e apontam algumas formas de lidar com este problema. Parece haver nestes trabalhos uma concordância de que a solução para este problema está na direção de utilizar exercícios práticos com os alunos. A seguir, encontra-se um breve levantamento realizado neste contexto.

Devedzic, Debenham e Popovic (2000) citam que uma das maiores causas do problema indicado nesta proposta é a quantidade de abstrações que os temas envolvendo Teoria da Computação possuem. Alunos de cursos de computação possuem um perfil de aprendizagem mais voltado ao uso de materiais de ensino práticos, com temas concretos. Neste contexto, os autores desenvolveram um Sistema Tutor Inteligente, denominado FLUTE (Formal Languages Automata Environment), no domínio de Linguagens Formais e Autômatos que apoia o ensino com exercícios realizados no mesmo computador, com exemplos de autômatos representados graficamente. Neste modelo, os autores criaram uma estrutura de sequenciamento dos conteúdos.

Cavalcante, Finley e Rodger (2004) citam que materiais utilizáveis em computador com recursos visuais e interativos, utilizados na prática dos conteúdos de Teoria da Computação, podem auxiliar no processo de aprendizagem do aluno. Para isto, eles citam o software JFLAP, que permite criar autômatos finitos, autômatos de pilha, gramáticas, expressões regulares, máquinas de Turing, entre outros.

O uso do JFLAP também foi alvo dos estudos de Chesñavar, Cobo e Yurcik (2003), onde este é comparado com outras ferramentas (Deus ex Machina e Java Computability Toolkit). Uma das principais contribuições deste trabalho é a percepção dos autores que materiais que permitem a simulação e experimentação das teorias, propiciam um maior envolvimento dos alunos.

Cogliati, Goosey, Grinder, Pascoe e Ross (2005) concordam com o trabalho de Cavalcante, Finley e Rodger (2004), em que a relação materiais visuais e ensino de Teoria da Computação possam melhorar o desempenho dos alunos. Neste trabalho

também é realizado um estudo, com resultados positivos quanto ao uso de tecnologias Web, associando hipertextos, vídeos, figuras e sons.

Goyal e Sachdeva (2009) concordam com a necessidade prática dos conteúdos, mas afirmam que materiais neste sentido são mais efetivos quando utilizam conteúdos interdisciplinares.

### **3. Ensino de Complexidade na UNIVALI**

O curso de Ciência da Computação da UNIVALI é de dedicação noturna, com duração de cinco anos. Em sua matriz curricular há algumas disciplinas ligadas diretamente ao que se convencionou como Teoria da Computação: Linguagens Formais e Autômatos (sexto período), Compiladores (sétimo período) e Complexidade de Algoritmos (sétimo período).

A disciplina mais relacionada a Teoria da Complexidade é a de "Complexidade de Algoritmos", que possui carga-horária de 36 horas-aula. Como ementa e respectivos conteúdos têm-se:

- Análise de Complexidade: apresentação e introdução a complexidade, complexidade pessimista e complexidade média;
- Noções de Problemas NP-Completo e Computabilidade: classes P, NP, NP-completo e complexidade de espaço;
- Métodos de Projeto de Algoritmos: casos práticos sobre projetos de algoritmos; e
- Máquina de Turing e de Post: conceitos sobre máquinas de Turing e Post, máquinas de Turing e decidibilidade.

As aulas são basicamente expositivas e dialogadas, com alguns exemplos práticos principalmente na unidade de "Análise de Complexidade". A disciplina foi utilizada no projeto para realização de testes com os materiais produzidos como produto da pesquisa relatada neste artigo.

### **4. Materiais Produzidos**

Analisando os temas envolvidos nas disciplinas de Teoria da Computação na UNIVALI, foram desenvolvidos materiais que tem como principal características possibilitar a interatividade dos alunos, visualizando uma demonstração passo a passo.

Na maioria dos materiais produzidos utilizou-se problemas de grafos, pois são fáceis de serem representados visualmente e de realizar analogias com objetos do cotidiano das pessoas. São também utilizados em referências de ensino na área de Teoria da Complexidade como: Toscani e Veloso (2005), Sipser (2007), Ziviani (2007), Arora e Barak (2009), Ramos, José Neto e Vega (2009).

Além disso, grafos são problemas muito presentes na literatura da Teoria da Complexidade. Para fortalecer a crença da dimensão que pode ser dada ao se trabalhar com problemas de grafos em conteúdos de complexidade, cita-se a obra de Garey e Johnson (1979), que na época reuniram 65 destes problemas, apresentando-os como pertencentes a classe NP-Completo que é tema fundamental para o estudo de Teoria da Complexidade.

Quanto aos temas de complexidade, foram abordados pelos materiais desenvolvidos: análise de algoritmos, classes de complexidade, tentativa e erro (com poda na árvore de *backtracking*), provando NP-completo com o objetivo de reforçar o conteúdo teórico ensinado em sala de aula pelo professor, portanto oferecendo uma alternativa no estudo dos temas.

#### 4.1. Análise da Complexidade

O material de ensino Análise de Complexidade tem características de colocar em prática a análise pessimista usando como base os algoritmos de ordenação. Os alunos informam o tamanho da entrada (campo "Tamanho da entrada"), selecionando as operações fundamentais (como demonstrado na Figura 1, atribuindo-se um "x" na coluna M, à esquerda da linha de código) podendo então realizar os teste, que tem como resposta a quantidade de vezes em que operações fundamentais foram executadas (campo "Total") e a resposta correta (campo "Resposta Correta").

The screenshot shows a window titled "Análise de Complexidade". At the top, there is an "Abrir" button. Below it, a text box contains the instruction: "O algoritmo abaixo é o Bubblesort. O Bubblesort é utilizado para ordenação de vetores. Marque as operações fundamentais deste algoritmo." Below this, there is a "Tamanho da entrada:" field with the value "100" and a "Teste" button. The main area is a table with columns "M", "L", and "Código". The code is a Bubble Sort implementation. The table has 10 rows, with "x" in the "M" column for rows 2 and 5. At the bottom, there are two fields: "Total" with the value "2373" and "Resposta Correta" with the value "4950".

M	L	Código
	1	for(var i=0; i<N-1; i++){
x	2	for (var j=0; j<N-1-i; j++){
	3	if (ENTRADA[j] > ENTRADA[j+1]){
	4	var aux = ENTRADA[j];
x	5	ENTRADA[j]= ENTRADA[j+1];
	6	ENTRADA[j+1] = aux;
	7	}
	8	}
	9	}
	10	

Total 2373      Resposta Correta 4950

Figura 1. Análise de Complexidade

#### 4.2. Classes de Complexidade

Um material foi desenvolvido para auxiliar o estudo de classes de complexidade de problemas (Figura 2). É importante ressaltar que o material foi desenvolvido supondo que as classes de complexidade NP, PSPACE, NP-Difícil e P não são equivalentes entre si.

Um diagrama de conjuntos (Diagrama de Venn) representando as classes de complexidade de problemas é demonstrado ao aluno. Alguns problemas muito comuns na literatura são dispostos dentro das classes, onde pode-se clicar tanto nos problemas quanto nas classes. Por meio desta ação, um quadro é exibido com explicações que podem auxiliar o estudo do aluno e a compreensão do tema.

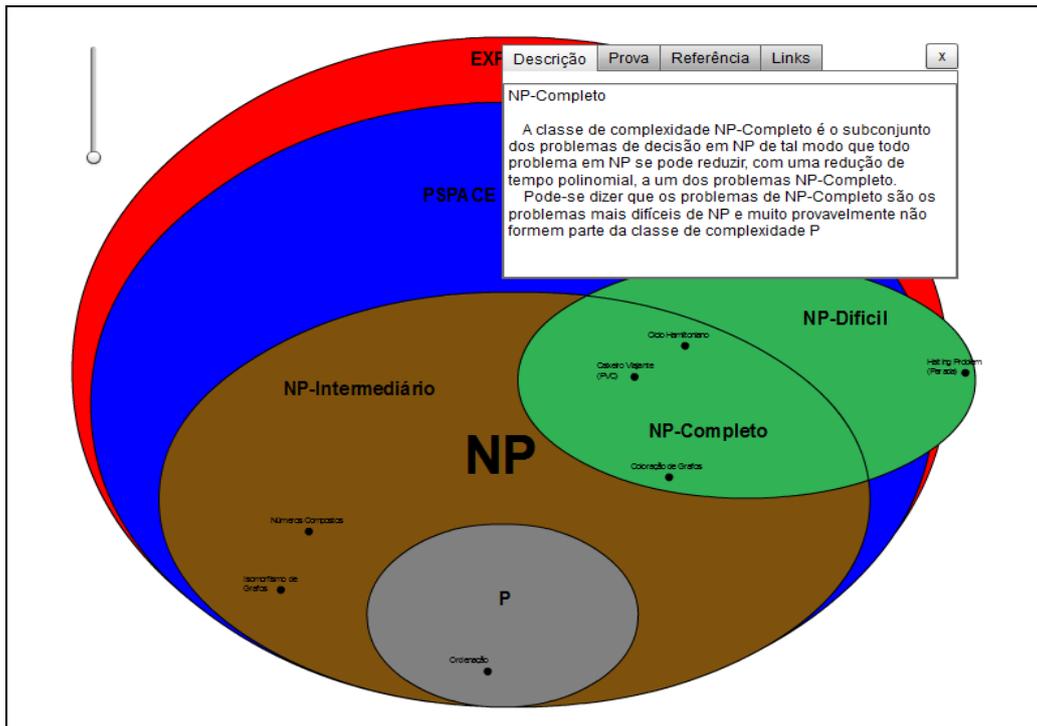


Figura 2. Classes de Complexidade

### 4.3. Algoritmos de Tentativa e Erro

O programa Tentativa e Erro com Poda (Figura 3) demonstra o conceito de poda em Árvore de Backtracking (por meio de explicação passo-a-passo), como alternativa de reduzir os tempos de execução de problemas NP-Completo.

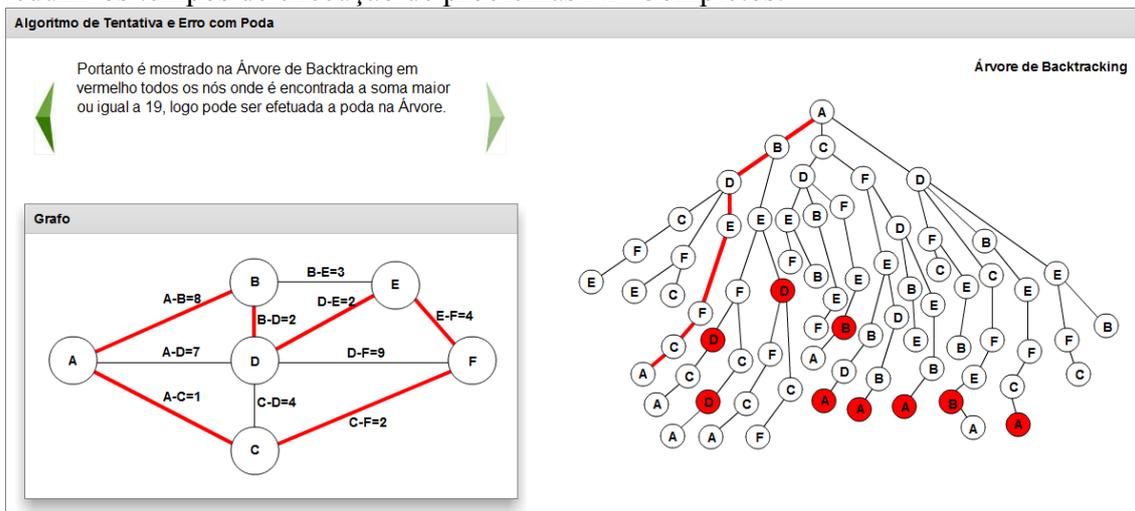


Figura 3. Tentativa e Erro

### 4.4. Provando NP-Completo

O material “Provando NP-Completo” (Figura 4) sugere a prática ao aluno para realização da prova de que os problemas Conjunto de Vértices, Clique e Conjunto

Independente são NP-Completo, por meio do conceito da transformação polinomial. Para compreender a prova, cada um dos problemas é apresentado interativamente.

Figura 4. Provando NP-Completo

## 5. Resultados

A pesquisa teve como alvo os alunos que cursaram a disciplina Complexidade de Algoritmos das turmas do primeiro e segundo semestres de 2011, e somente foram avaliados os alunos que concordaram com o Termo de Consentimento Livre e Esclarecido (TCLE). Dois tipos de avaliação foram executados na pesquisa: um questionário e análise do desempenho dos participantes.

O questionário foi desenvolvido com a ferramenta Google Docs (Figura 5) e é composto por perguntas objetivas. O mesmo foi submetido por e-mail aos alunos participantes que avaliaram os objetos de aprendizagem, quanto à clareza da linguagem usada e se estes materiais ajudariam na compreensão dos conteúdos vistos em sala de aula, por último os participantes poderiam fazer sugestões e críticas referente a estes trabalhos. Ao finalizar a aplicação do questionário, os dados foram compilados e, como resultado, todos os participantes avaliaram os materiais como de linguagem compreensível, e que estes ajudariam na aprendizagem de Complexidade de Algoritmos.

<p><b>Análise de Complexidade</b></p> <p>O quanto este programa ajudaria na compreensão dos conteúdos de Análise de Complexidade?</p> <p><input type="radio"/> Não Ajudaria  <input type="radio"/> Ajudaria  <input type="radio"/> Ajudaria Muito</p> <p>A linguagem utilizada no programa para explicação dos conteúdos é:</p> <p><input type="radio"/> Não Compreensível  <input type="radio"/> Compreensível</p> <p><b>Definições de Classes de Complexidade</b></p> <p>O quanto este programa ajudaria na compreensão dos conteúdos de Classes de Complexidade?</p> <p><input type="radio"/> Não Ajudaria  <input type="radio"/> Ajudaria  <input type="radio"/> Ajudaria Muito</p> <p>A linguagem utilizada no programa para explicação dos conteúdos é:</p> <p><input type="radio"/> Não Compreensível  <input type="radio"/> Compreensível</p>	<p><b>Provando NP-Completo</b></p> <p>O quanto este programa ajudaria na compreensão dos conteúdos de Provando NP-Completo?</p> <p><input type="radio"/> Não Ajudaria  <input type="radio"/> Ajudaria  <input type="radio"/> Ajudaria Muito</p> <p>A linguagem utilizada no programa para explicação dos conteúdos é:</p> <p><input type="radio"/> Não Compreensível  <input type="radio"/> Compreensível</p> <p><b>Tentativa e Erro com Poda</b></p> <p>O quanto este programa ajudaria na compreensão dos conteúdos de Tentativa e Erro com Poda?</p> <p><input type="radio"/> Não Ajudaria  <input type="radio"/> Ajudaria  <input type="radio"/> Ajudaria Muito</p> <p>A linguagem utilizada no programa para explicação dos conteúdos é:</p> <p><input type="radio"/> Não Compreensível  <input type="radio"/> Compreensível</p>
<p><b>Críticas e Sugestões</b></p> <div style="border: 1px solid black; height: 40px; width: 100%;"></div>	

**Figura 5. Formulário**

Uma avaliação sobre o desempenho dos alunos foi realizada, com o intuito de identificar fragilidades ou potencialidades no uso do material desenvolvido. Este trabalho de pesquisa foi realizado junto ao professor de Complexidade de Algoritmos (co-autor deste artigo) que avaliou o desempenho das duas turmas: a do primeiro semestre de 2011 (2011-1) que teve participação de 56% e a turma do segundo semestre (2011-2) com 69% de participação.

Esta análise foi realizada sobre as médias dos alunos participantes. Em cada semestre os alunos contam com três médias, chamadas de M1, M2 e M3, cujo a média final é obtida por meio da média aritmética das mesmas. Obtendo nota seis ou superior, os alunos são aprovados, senão reprovados. No levantamento do desempenho a Tabela 1 foi gerada.

**Tabela 1. Resultados**

Semestre	M1	M2	M3	M.Final	Nº Aprovados	Nº Reprovados	% participação
2011-1	5,85	7,90	6,64	6,80	88,89%	11,11%	56,25%
2011-2	7,35	7,73	6,22	7,10	100,00%	0,00%	69,23%

A partir da análise da Tabela 1, foram extraídas as seguintes constatações:

- na Média M1, nota-se que a turma 2011-2 que recebeu os materiais desenvolvidos obteve maior média. Nesta média, concentravam os conteúdos nos quais os alunos apresentavam maior dificuldade de

aprendizagem (Análise do desempenho de algoritmos (pessimista), e introdução a complexidade);

- para os alunos da turma 2011-2 em relação a 2011-1, a taxa de aprovação na disciplina Complexidade de Algoritmos cresceu 12,5% e não houve reprovações; e
- segundo o professor da disciplina, os resultados inferiores das médias M2 e M3 da turma 2011-2 em relação a 2011-1, podem estar relacionados a bons resultados obtidos na primeira média portanto, diminuindo o comprometimento da turma com as médias seguintes.

Com base nas seguintes constatações acima, percebe-se que é possível obter uma melhora dos alunos quanto ao desempenho na disciplina com a utilização de materiais práticos durante o processo de aprendizagem.

## 6. Conclusão

Foram produzidos quatro materiais abordando temas foco da disciplina de Complexidade de Algoritmos. O objetivo em curto prazo foi auxiliar os alunos da disciplina no curso de Ciência da Computação na UNIVALI. Pretende-se agora, realizar uma divulgação dos materiais em eventos e em listas de discussões sobre ensino de computação.

Como resultado já obtido, pode-se destacar uma das observações do professor da disciplina na UNIVALI: desde que a leciona, esta foi a primeira vez em que os índices de reprovação chegaram a zero.

Ao final, as duas perguntas de pesquisa que surgiram na proposta deste projeto foram respondidas parcialmente, pois os resultados obtidos na pesquisa permitiram apenas coletar evidências de suas respostas:

- **Pergunta:** *Como pode-se viabilizar a prática de conteúdos da Teoria da Complexidade?*

**Resposta:** a viabilização da prática dos conteúdos de Teoria da Complexidade pode ser realizada por meio do uso de enunciados com apelo visual e a proposta de exercícios, os quais se poderia verificar as respostas corretas.

- **Pergunta:** *Viabilizar a prática auxiliará o aluno na aprendizagem de Teoria da Complexidade?*

Resposta: de acordo com a análise de desempenho realizada entre os participantes da pesquisa e o questionário enviado aos mesmos, há evidências que demonstram que o uso da “prática” com os alunos, pode favorecer a aprendizagem.

Com os resultados obtidos na pesquisa, acredita-se que o uso de materiais práticos podem auxiliar o ensino de Teoria da Complexidade, bem como destacaram os seguintes trabalhos, na área de Teoria da Computação: Devedzic, Debenham e Popovic (2000), Chesñevar, Cobo e Yurcik (2003), Cavalcante, Finley e Rodger (2004), Cogliati, Goosey, Grinder, Pascoe e Ross (2005), Goyal e Sachdeva (2009).

Como trabalho futuro, pretende-se ampliar a pesquisa, produzindo mais materiais de ensino, aplicando outros experimentos com alunos e egressos. Os materiais foram disponibilizados na Internet e podem ser acessados livremente em Rese e Santiago (2011). Este projeto foi fomentado pelo programa do Artigo 170 do governo do estado de Santa Catarina e pela Fundação Universidade do Vale do Itajaí.

### Referências Bibliográficas

- Aaronson, S. (2011) "The Complexity Zoo", [www.cse.unl.edu/~cbourke/latex/ComplexityZoo.pdf](http://www.cse.unl.edu/~cbourke/latex/ComplexityZoo.pdf), Janeiro.
- Arora, S. e Barak, B. (2009), Computational Complexity: a modern approach. 1ª edição, Cambridge University Press, ISBN 9780521424264
- Cavalcante, R., Finley, T. e Rodger, S. H. (2004). A Visual and Interactive Automata Theory Course with JFLAP 4.0. In *Proceedings of 35th SIGCSE Technical Symposium on Computer Science Education*, New York. ACM Press.
- Chesñevar, C., Cobo, M. L. e Yurcik, W. (2003). Using Theoretical Computer Simulators for Formal Languages and Automata Theory. In *35th ACM SIGCSE Bulletin*, New York. ACM. doi>10.1145/782941.782975.
- Cogliati, J. J., Goosey, F. W., Grinder, M. T., Pascoe, B. A. e Ross, R. J. (2005). Realizing the Promise of Visualization in the Theory of Computing. In *Journal on Educational Resources in Computing (JERIC)*, v.5. New York. ACM. ISSN 1531-4278.
- Devedzic, V., Debenham, J. e Popovic, D. (2000). Teaching Formal Languages by an Intelligent Tutoring System. In *Educational Technology Society*, v.3. ISSN 1436-4522.
- Garey, M. R. e Johnson, D. S. (1979), Computers and Intractability: a guide to the theory of NP-Completeness, Bell Telephone Laboratories Inc, ISBN 0-7167-1044-7.
- Goyal, M.; Sachdeva, S. (2009). Enhancing Theory of Computation Teaching Through Integration with other Courses. In *International Journal of Recent Trends in Engineering*, v.1. Academy Publisher.
- MEC (2009). "Referenciais Nacionais de Cursos de Graduação (LICENCIATURA E BACHARELADO): consulta pública", [http://portal.mec.gov.br/index.php?option=com\\_content&view=article&id=13812&Itemid=995](http://portal.mec.gov.br/index.php?option=com_content&view=article&id=13812&Itemid=995).
- Ramos, M. V. M., José Neto, J. e Vega, I. S. (2009). Linguagens Formais: teoria, modelagem e implementação, Bookman, ISBN 978-85-7780-453-5.
- Rese, A. L. R. e Santiago, R. (2011), "Materiais para o Ensino de Complexidade". <http://lop2p.org/comp/>, Dezembro.
- Sipser, M. (2007). Introdução à Teoria da Computação, 2ª edição, Cengage Learning, ISBN: 978-85-221-0499-4.
- Toscani, L. V. e Veloso, P. A. S. (2005). Complexidade de algoritmos, 2ª edição, Bookman.
- Ziviani, N. (2007). Projeto de algoritmos, Thomson, 2007.