

## **Aulas Interativas como Catalizador no Processo de Aprendizagem em Programação**

**Daniela C. B. da Silva<sup>1</sup>, Herleson P. Pontes<sup>1,2</sup>**

<sup>1</sup>Centro de Ciências Tecnológicas – Universidade de Fortaleza (UNIFOR)  
Fortaleza – CE – Brasil

<sup>2</sup>Faculdades Nordeste (DeVry Brasil)  
Fortaleza – CE – Brasil

danisilva.bc@gmail.com, herleson@unifor.com

**Abstract.** *This paper describes the implementation and evaluation of interactive lessons focused on the maximization of the teaching-learning process in the disciplines of computer programming, aiming to outline a specific methodology for teaching students that plea of Computing and Engineering. More specifically, students participated in two different classes, and then evaluated the concepts presented in these lessons. The results show that these classes increase student interest and participation in the formation of their programming skills, and narrowing the teacher-student relationship and enhance the fixation of the content presented.*

**Resumo.** *Esse trabalho descreve a aplicação e avaliação de aulas interativas voltadas para a maximização do processo de ensino-aprendizado em disciplinas de programação de computadores, com o objetivo de delinear uma metodologia específica para o ensino desse fundamento aos alunos de Computação e Engenharias. Mais especificamente, os alunos participaram de dois diferentes tipos de aula, e avaliados em seguida sobre os conceitos apresentados nessas aulas. Os resultados mostram que essas aulas aumentam o interesse e a participação do aluno na formação das suas habilidades em programação, além de estreitar a relação aluno-professor e potencializar a fixação do conteúdo apresentado.*

### **1. Introdução**

O currículo de programação constitui um importante alicerce na formação de profissionais na área da Computação, pois é responsável pela lapidação do raciocínio lógico do aluno para torna-lo um solucionador de problemas [Pontes 2013]. Apesar da sua importância, o ensino de programação constitui-se um desafio para professores que lecionam as disciplinas que envolvem esse fundamento. Isso porque a compreensão dos tópicos apresentados envolve não somente o trabalho mecânico de praticar e entender o assunto, mas também o uso o raciocínio lógico para desenvolver as habilidades necessárias para solucionar problemas. Ademais, o processo de aprendizado torna-se mais desafiador quando o professor não possui as habilidades necessárias para o ensino dessa disciplina. Conseqüentemente, muitos alunos acabam não incorporam os vários conceitos ministrados pelo professor.

Deste modo, faz-se necessário, durante o ensino de disciplinas que envolvam programação, o empenho dos professores para o desenvolvimento de atividades que

atraiam a atenção e interesse do aluno pelo assunto, tornando produtivo o processo de aprendizado. A realização de atividades diferenciadas em sala, como aulas dinâmicas e exercícios práticos, são atividades que contribuem para o aprendizado do aluno, especialmente quando os tópicos apresentados envolvem conceitos abstratos e fundamentos matemáticos. Essas atividades proporcionam ao aluno a repetição do que foi lecionado, mas sob diferentes perspectivas, auxiliando-o na fixação dos tópicos da área de programação [Soares e Pinto 2001].

À luz desse cenário, realizamos a aplicação e avaliação de aulas envolvendo diferentes dinâmicas, com o objetivo de tornar eficiente e menos traumático o processo de ensino-aprendizado em disciplinas de programação. Acreditamos que seja possível delinear uma série de atividades, como as abordadas pelo nosso estudo, que permitam ao professor uma melhor maneira de lecionar disciplina na área de programação.

## 2. Trabalhos Relacionados

Pesquisas relativas à criação de metodologias de ensino que otimizam o aprendizado do aluno formam uma das linhas de pesquisa mais abordadas pela comunidade acadêmica. Segundo [Ynoguti 2005], um método eficiente para a compreensão dos tópicos apresentados aos alunos é a resolução de problemas complexos através da divisão em pequenos problemas, de menor complexidade. De acordo com o autor, os alunos apresentam menor dificuldade na resolução de questões quando esses problemas são divididos em problemas menores, devido a facilidade na compreensão e implementação das soluções. Entretanto, esse estudo também aborda a aversão ao método dos alunos que já estão familiarizados com programação.

De acordo com [Soares e Pinto 2001], a resolução de exercícios em sala de aula merece atenção especial, pois é a partir dessa resolução que aumenta o envolvimento do aluno com a disciplina, além de possibilitar a esse aluno visualizar suas aplicações práticas. Para as autoras, a resolução de exercícios estimula o aluno a desenvolver um raciocínio lógico mais apurado, preparando-o para enfrentar novas situações.

O estudo apresentado por [Falckembach e Araujo 2006] ressalta outro importante requisito no ensino de programação: a importância da construção de uma base sólida acerca do assunto abordado. Os autores destacam que todas as atividades inerentes ao processo de aprendizado devem ser realizadas em partes, como a apresentação de um novo conceito e a resolução de um algoritmo.

Na mesma linha do estudo anterior, [Jesus e Brito 2010] discute as dificuldades no ensino da programação e no desenvolvimento do raciocínio lógico pelo aluno. Esse estudo recomenda ao professor o completo domínio do assunto e a facilidade em entender a lógica de seus alunos. Além disso, os autores apresentam um processo de resolução de exercícios em dois estágios. No primeiro, o professor avalia técnicas e discute com os alunos a respeito do problema, facilitando a compreensão da questão proposta pelo aluno. Em seguida, o professor trabalha a implementação em código do algoritmo requisitado.

Diversos estudos propõem processos dinâmicos e interativos para o aprendizado. Em [Silva, Bernardi e Müller 2011] e [Prikladnicki *et al* 2009], técnicas mais dinâmicas para o ensino de disciplinas são abordadas. O primeiro estudo avalia o impacto no aprendizado a partir da realização de um *Serious Game* [Pontes 2013], enquanto o segundo trabalhou com dinâmicas de grupos, atividades lúdicas, jogos e práticas em sala. Ainda na mesma linha de pesquisa, [Ternório, Silva e Rocha 2013] concluíram que é

benéfico aos estudantes a aplicação dos conhecimentos práticos adquiridos em sala através de simulações envolvendo cenários do dia-a-dia do profissional. Com atividades práticas, dinâmicas e coletivas, os autores simularam uma experiência de desenvolvimento de um software, onde os alunos aplicaram todo o conhecimento fixado.

Outro problema frequente presente no ensino de programação reside na determinação da corretude de um código. [Santos e Costa 2006] mostram algumas ferramentas que auxiliam o professor no processo de ensino, como o *Astral* [Garcia, Rezende e Calheiros 1997]. De maneira semelhante, os programas *TBC-AED* e *TCB-AED/WEB* [Santos e Costa 2005] apresentam uma simulação gráfica do código desenvolvido pelo aluno. Através dessa simulação, é possível simplificar a aprendizagem do aluno.

Com todas as contribuições vindas dos autores citados, para este artigo, usamos esses dados coletados para assim poder obter resultados específicos para o ensino de disciplina de programação.

### **3. Metodologia de Ensino**

Com o objetivo de reduzir os problemas relativos ao ensino de programação, desenvolvemos três técnicas para o ensino e avaliação dos alunos em disciplinas de programação. Cada um desses métodos visa aumentar o envolvimento do aluno no processo de aprendizagem e, conseqüentemente, aumentar a sua motivação em sala de aula.

#### **3.1. Questão Assistida**

Essa técnica consiste, inicialmente, na divisão de um exercício de programação complexo em pequenas partes. Em seguida, os alunos são solicitados a implementarem cada um desses trechos. Para cada um desses trechos, foi dado um tempo para que os alunos pudessem pensar e implementar suas soluções, estimulando-os a desenvolver habilidades como criatividade e raciocínio lógico.

Ao término do tempo, o professor então apresenta a solução detalhada do trecho abordado, com o objetivo de incentivar os alunos a assimilarem a lógica envolvida e possibilitar a solução de problemas similares no futuro. Em seguida, inicia-se a contagem de tempo para os alunos desenvolverem outro trecho do programa. Todo esse processo é repetido até que, finalmente, todos os trechos estejam implementados e o programa totalmente funcional.

A *questão assistida* tenta sanar o problema dos alunos que possuem alguma dificuldade nos diversos trechos do exercício e que, devido a dependência desses trechos na montagem da solução final, sentem dificuldade em construir todo o programa. Essa técnica ajuda o aluno a fixar e aplicar a ideia de divisão e conquista, a qual divide um algoritmo em pequenos problemas, de compreensão e resolução simplificada. Portanto, essa técnica tenta envolver aquele aluno que não compreende o início de um exercício possibilitando implementação o resto da questão.

#### **3.2. Aula Dinâmica**

Nessa técnica, a qual trabalha com o processo de ensino de determinado assunto em sala de aula, consiste no uso de diferentes dinâmicas em sala para lecionar importantes tópicos

na área de programação. Nessas aulas, duas características são essenciais para o sucesso das atividades:

- O maior número possível de alunos deve se envolver na realização das atividades;
- A aula deve ser dividida em duas partes: um momento de *apresentação*, onde o professor apresenta uma visão geral do tópico a ser abordado pela dinâmica; e o momento da *interação*, onde os alunos cooperam entre si afim de resolver o problema apresentado, com o auxílio do professor.

Ao final da atividade, os alunos são desafiados pelo professor, que apresenta um problema final. Caso as atividades tenham sido realizadas com sucesso, cada aluno terá a capacidade de solucionar, por si só, o desafio.

A *aula dinâmica* pode ser aplicada de diversas maneiras, sempre de acordo com a criatividade do professor. Essa técnica possibilita uma maior participação dos alunos na aula e, conseqüentemente, uma maior eficiência no processo de aprendizado.

#### 4. Estudo de Caso

Como estudo de caso, aplicamos as técnicas desenvolvidas em uma turma de *Estrutura de Dados* com alunos dos cursos de *Ciência da Computação* e *Engenharia de Computação*. Os tópicos abordados nas aulas que utilizaram essas técnicas foram *Estruturas Lineares*, *Estruturas Encadeadas* e *Árvores Binárias*.

Foram realizadas duas aulas aplicando a técnica de *questões assistidas*, ambas realizadas em um laboratório de Informática. Em cada aula, os alunos resolveram dois problemas relativos ao tópico de estruturas lineares, com o tempo limite de 10 minutos para cada trecho do código, seguido de mais 10 minutos de apresentação da solução desse trecho.

Dois *aulas dinâmicas* foram realizadas com os alunos em sala de aula, cada uma com uma atividade interativa específica. Na primeira aula, na qual o tópico abordado foi o funcionamento das estruturas de dados encadeadas, os alunos foram posicionados no centro da sala, e cada um segurou uma pequena lousa. Cada lousa representava as informações de cada nó que formava a estrutura encadeada, com seu respectivo *dado*, *endereço de memória* e uma referência para o *próximo elemento* da lista. Em seguida, foi pedido que eles informassem seus nomes ao atributo *dado*. Em seguida, utilizamos os alunos como nós de uma lista e simulamos os principais algoritmos pertencentes a essa estrutura de dados. Ao fim, os alunos foram desafiados a realizarem os métodos apresentados anteriormente, sem a interferência do professor.

A segunda aula dinâmica foi uma competição de programação entre grupos de alunos, chamada de *Code-Off*. Nessa disputa, as equipes foram desafiadas a implementar uma árvore binária somente com os fundamentos teóricos apresentados durante a aula. O principal diferencial dessa competição está no fato que cada equipe tinha um [único computador disponível], e cada integrante tinha um tempo para codificar parte do programa. A cada três minutos, o integrante à frente do computador era trocado, de tal forma que todos os alunos participaram da montagem do código final.

Para verificar se o uso das técnicas apresentadas possibilitou uma melhoria na construção do conhecimento na disciplina de *Estrutura de Dados*, uma análise foi realizada a partir dos resultados obtidos a partir de um questionário, aplicado ao término das aulas envolvendo *questões assistidas* e das *aulas dinâmicas*, no qual foram feitas oito

perguntas sobre o perfil do aluno e sua percepção das atividades realizadas no seu aprendizado em programação. A Figura 1 consolida os resultados dessa avaliação.

## 5. Resultados e Discussão

No total, 33 estudantes com média de idade de 20 anos, participaram dessa avaliação. Desses, 30 são do sexo *masculino* e 3 são do sexo *feminino*. A maior parte desse grupo, 29 alunos, utilizam o computador diariamente por períodos de, no mínimo, 4 horas. Em relação a atuação profissional, 21 participantes do experimento só estudam, 8 estudam e trabalham, e 4 já trabalharam, mas atualmente só estudam.

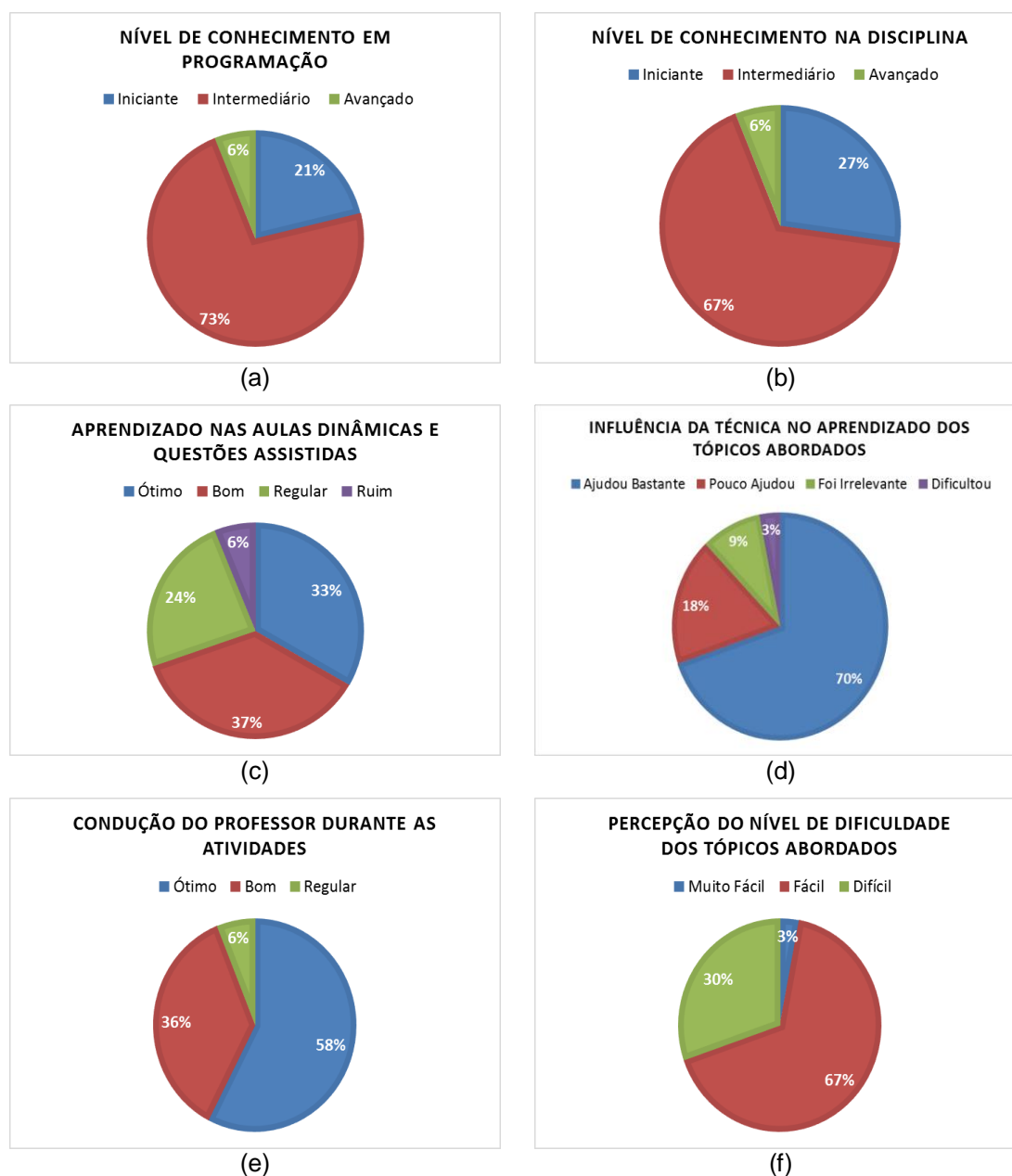


Figura 1. Resultados obtidos a partir do questionário

Para classificar os grupos de alunos de acordo com o nível de maturidade em programação, os alunos classificaram seu nível de conhecimento na área. De acordo com a Figura 1a, 73% do total de participantes classificou seu nível como *intermediário*. Isso indica que mais de 3/4 da turma considera possuir o pré-requisito necessário para ingressar na disciplina de estruturas de dados.

Em seguida, os alunos classificaram seu nível de conhecimento na disciplina. No total, 73% dos alunos informam possuir maturidade para resolver problemas envolvendo estruturas de dados, sugerindo que o aprendizado é considerado satisfatório.

Iniciando a avaliação das técnicas de *questões assistidas* e *aulas dinâmicas*, perguntou-se ao participante a sua percepção em relação ao seu aprendizado. Como observa-se na Figura 1c, 33% dos alunos classificaram o aprendizado como *ótimo* e 37% avaliaram sua experiência como *boa*. Portanto, mais da metade da turma avaliou como positivo o impacto dessas técnicas em seu aprendizado.

Ainda avaliando o impacto das técnicas, a Figura 1d ilustra que 70% do total acreditam que as questões assistidas e as aulas dinâmicas contribuíram de forma significativa e positiva no aprendizado dos fundamentos de programação apresentados nessas aulas.

Outro ponto importante foi a conduta do professor durante a aplicação das técnicas. Apenas 6% dos alunos não aprovaram o comportamento e a interação do professor durante a realização das atividades. Isso constata a forte contribuição do docente para o sucesso das técnicas aplicadas, como mostra a Figura 1e.

Por fim, os participantes classificaram o nível de dificuldade dos assuntos abordados durante as atividades. 70% dos alunos consideraram os tópicos trabalhados como *fácil* ou *muito fácil*. Esse resultado é um dos mais significativos para a pesquisa, pois os temas abordados são geralmente considerados difíceis pelos alunos que cursam a disciplina de estrutura de dados. Portanto, os resultados sugerem que as questões assistidas e as aulas dinâmicas podem reduzir a complexidade dos temas inerentes as disciplinas de programação.

## 6. Conclusão

As técnicas desenvolvidas para auxiliar o processo de aprendizado em disciplinas de programação possibilitam um envolvimento maior do aluno com o seu aprendizado, e estimula a descoberta de novos conhecimentos através do hábito de estudo. Os resultados obtidos indicam que as *aulas dinâmicas* despertam maior interesse do aluno, devido a sua participação direta durante a aula, e que os *exercícios assistidos* auxiliaram boa parte dos alunos que se encontravam em dificuldades.

Como trabalhos futuros, recursos adicionais, como redes sociais, serão utilizados e avaliados como ferramenta no processo de ensino. Também pretende-se delinear regras específicas para cada uma das técnicas apresentadas nesse estudo, e validá-las em outras disciplinas de programação. Adicionalmente, pretende-se ampliar a pesquisa para um número maior de turmas, afim de validar os resultados obtidos em cenários mais abrangentes e possibilitar uma comparação com turmas nas quais essas metodologias não sejam aplicadas.

## Referências

- Falckembach, G. A. M., Araujo, F. V. (2006) “Aprendizagem de Algoritmos: Dificuldades na Resolução de Problemas”, Anais SULCOMP, Vol. 2, No. 2.
- Garcia, I. C., Rezende, P. J. D., Calheiros, F. C. (1997) “Astral: Um Ambiente para Ensino de Estruturas de Dados através de Animações de Algoritmos”, Revista Brasileira de Informática na Educação, No. 1.
- Jesus, A., Brito, G. S. (2010) “Concepção de Ensino-Aprendizagem de Algoritmos e Programação de Computadores: A Prática Docente”. Revista Varia Scientia. Vol. 09, No. 16, p. 149-158.
- Pontes, H. P. (2013) "Desenvolvimento de Jogos no Processo de Aprendizado em Algoritmos e Programação de Computadores." In: Proceedings of the XII Simpósio Brasileiro de Games e Entretenimento Digital (SBGames). São Paulo.
- Prikladnicki, R., Albuquerque, A. B., von Wangenheim, C. G., Cabral, R. (2009) “Ensino de Engenharia de Software: Desafios, Estratégias de Ensino e Lições Aprendidas”. In: XII Simpósio Brasileiro de Engenharia de Software, Fortaleza.
- Santos, R. P. D., Costa, H. A. X. (2005) “TBC-AED e TCB-AED/WEB: Um Desafio no Ensino de Algoritmos, Estruturas de Dados e Programação”. In: IV Workshop de Educação em Computação e Informática do Estado de Minas Gerais (WEIMIG’ 2005), Varginha.
- Santos, R. P. D., Costa, H. A. X. (2006) “Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos Iniciantes em Computação e Informática”. In: INFOCOMP, Vol. 5, No. 1.
- Silva, T. G. D., Bernardi, G., Müller, F. M. (2011) “Abordagem de Apoio ao Ensino e Aprendizagem de Teste de Software Baseada em Jogos Sérios e Mundos Virtuais”. Anais do XXII SBIE - XVII WIE, Aracaju.
- Soares, M. T. C., Pinto, N. B. (2001) “Metodologia da Resolução de Problemas”. In: 24<sup>a</sup> Reunião ANPEd, Caxambu.
- Ternório, J. V. S., Silva, P. M. D., Rocha, T. A. B. (2013) “Projetos de Desenvolvimento como Ferramenta no Ensino de Engenharia de Software”. In: XIII Jornada de Ensino, Pesquisa e Extensão – JEPEX 2013 – UFRPE: Recife.
- Ynoguti, C. A. (2005) “Uma Metodologia para o Ensino de Algoritmos”. Global Congress on Engineering and Technology Education. São Paulo, p. 684-687.