# PENSAMENTO COMPUTACIONAL APLICADO AO ENSINO DE PROGRAMAÇÃO NO ENSINO SUPERIOR

**Rebeca Einhardt Fiss** [1]

**Rebeca Fiss** [2]

**Tulon Pereira da Fontoura** [3]

**Dhyonatan Santos de Freitas** [4]

**Ana Paula Ludtke Ferreira** [5]

**Resumo:**

O ensino de programação de computadores está presente nos primeiros semestres dos cursos da área de Computação. A falta de hábito em abstrair procedimentos genéricos para solução de problemas é uma das maiores dificuldades que o aluno enfrenta logo no início da graduação, provocando altos percentuais de reprovações e consequente aumento da evasão. A mecanização dos processos na aprendizagem apresenta a forma, mas não os motivos da utilização de certos conceitos de programação, dificultando o entendimento por parte dos alunos. O desenvolvimento do raciocínio computacional, necessário aos alunos em disciplinas de programação, exige a adoção de um método de ensino no qual seja possível apresentar as noções de algoritmo mas, sobremaneira, trabalhar a noção de algoritmo como tradução de uma processo de solução de um problema. O termo pensamento computacional (PC) foi introduzido por Jeannette Wing, com o objetivo de englobar desde a estruturação do raciocínio lógico, até o comportamento humano para a ação de resolução de problemas. A adoção do PC pode orientar de forma ampla a atividade mental de abstrair problemas e formular soluções descritas em algoritmos. A partir dos conceitos e do problema apresentado, este trabalho tem os seguintes objetivos: i) apresentar uma proposta de ensino, aos alunos do curso de Engenharia de Computação da Universidade Federal do Pampa, utilizando o PC como forma de ensino aprendizagem de dois tópicos da disciplina de programação; e ii) realizar uma avaliação qualitativa do desempenho dos alunos nas tarefas propostas, comparando os resultados atingidos (utilizando o método de ensino proposto) em relação aos conhecimentos obtidos através de métodos tradicionais de ensino. A estratégia escolhida para o desenvolvimento do trabalho foi a execução de uma oficina com carga horária total de 12 horas, sendo 4 horas de atividades presenciais e 8 horas de atividades à distância, ofertadas por meio do ambiente virtual de aprendizagem Moodle. O público alvo foi composto por discentes do curso de Engenharia de Computação, da Universidade Federal do Pampa, Campus Bagé. Ao final da oficina pode se perceber uma melhora na capacidade dos estudantes de interpretar e abstrair os problemas após a utilização da

metodologia de ensino aprendizagem proposta, apresentando soluções satisfatórias para os problemas tratados. Na sua autoavaliação, realizada presencialmente, os estudantes afirmaram ter um melhor entendimento sobre os conteúdos abordados do que tinham antes da participação na oficina. Como trabalhos futuros, deseja-se replicar este experimento em uma quantidade maior de discentes.

**Palavras-chave:** Pensamento Computacional, Recursão, Programação, Algoritmos

**Modalidade de Participação:** Pós-Graduação

## PENSAMENTO COMPUTACIONAL APLICADO AO ENSINO DE PROGRAMAÇÃO NO ENSINO SUPERIOR

[1] Aluno de pós-graduação. rebecafiss@ifsul.edu.br. Autor principal

[2] Aluno de pós-graduação. rebecafiss@ifsul.edu.br. Apresentador

[3] Aluno de pós-graduação. tulonfontoura@unipampa.edu.br. Co-autor

[4] Aluno de pós-graduação. dhyonatan.freitas@gmail.com. Co-autor

[5] Docente. anaferreira@unipampa.edu.br. Orientador

# PENSAMENTO COMPUTACIONAL (PC) APLICADO AO ENSINO DE PROGRAMAÇÃO NO ENSINO SUPERIOR

## 1 INTRODUÇÃO

O ensino de programação de computadores está presente nos primeiros semestres dos cursos da área de Computação, por se tratar de competência necessária aos egressos desses cursos. A tarefa dos docentes dessa disciplina, contudo, não é trivial. A programação exige capacidade de abstração para construir a solução de um problema em passos de uma tarefa e, posteriormente, traduzi-los em códigos de linguagens que possam ser interpretados pelo computador. A falta de hábito em abstrair procedimentos genéricos para solução de problemas é uma das maiores dificuldades que o aluno enfrenta logo no início da graduação, provocando altos percentuais de reprovações e consequente aumento da evasão [FURTADO, 2012].

A evasão é um problema que aflige as instituições de ensino públicas e privadas. As perdas de estudantes que iniciam e não terminam seus cursos provocam desperdícios sociais, acadêmicos e econômicos [SALVA FILHO, 2007]. No setor público, são recursos públicos investidos sem o devido retorno à sociedade; no setor privado, é uma importante perda de receitas. Em ambos os casos, a evasão pode ser tomada como fonte de ociosidade de professores, funcionários, equipamentos e espaço físico. O levantamento realizado pela Pró-Reitoria de Graduação da Universidade Federal do Pampa constatou que, nos últimos cinco anos, os cursos de Ciência da Computação e Engenharia de Computação apresentaram uma taxa de aproximadamente 66% e 63%, respectivamente, de desistências por parte dos alunos [PROGRAD, 2018]. Uma das razões para essa acentuada desistência é a dificuldade encontrada pelos alunos com os conteúdos da disciplina de Algoritmos e Programação, que requerem o desenvolvimento do raciocínio lógico e abstrato, voltado para a resolução de problemas, com o auxílio da computação.

As metodologias de ensino-aprendizagem desenvolvidas em sala de aula têm um impacto significativo nos resultados apresentados pelos alunos. Na maioria dos casos, o ensino de programação utiliza fluxogramas e pseudocódigos para demonstração de conceitos abstratos, com foco na linguagem de representação e não no processo de construção da solução em si. O ambiente único de aprendizagem também é um obstáculo, no sentido de os métodos de estudo estarem baseados em leituras sucessivas, memorização e mecanização de procedimentos. A mecanização dos processos na aprendizagem apresenta a forma, mas não os motivos da utilização de certos conceitos de programação, dificultando o entendimento por parte dos alunos [GOMES, 2008]. O desenvolvimento do raciocínio computacional, necessário aos alunos em disciplinas de programação, exige a adoção de um método de ensino no qual seja possível apresentar as noções de algoritmo mas, sobremaneira, trabalhar a noção de algoritmo como tradução de uma processo de solução de um problema. A construção da capacidade de abstração é fundamental para o sucesso na aprendizagem de programação, principalmente para compreender problemas e propor soluções para eles.

O termo *pensamento computacional* foi introduzido por Jeannette Wing [WING, 2006], com o objetivo de englobar desde a estruturação do raciocínio lógico, até o comportamento humano para a ação de resolução de problemas. A adoção do PC pode orientar de forma ampla a atividade mental de abstrair problemas e formular soluções descritas em algoritmos, em qualquer área do conhecimento; aplica-se, de forma mais natural e direta, ao ensino de programação. Para se chegar à solução de um problema utilizando o PC, é sugerido a utilização de alguns passos, no primeiro momento é necessário quebrar o problema em partes menores que possam ser solucionadas mais facilmente, logo depois identificar padrões, processo fundamental para a recursão, para então se construir uma solução em alto nível de abstração. De posse desta solução, torna-se fácil chegar à sua codificação.

A partir dos conceitos e do problema apresentado, este trabalho tem os seguintes objetivos: i) apresentar uma proposta de ensino, aos alunos do curso de Engenharia de Computação da Universidade Federal do Pampa, utilizando o PC como forma de ensino-aprendizagem de dois tópicos da disciplina de programação; e ii) realizar uma avaliação qualitativa do desempenho dos alunos nas tarefas propostas, comparando os resultados atingidos (utilizando o método de ensino proposto) em relação aos conhecimentos obtidos através de métodos tradicionais de ensino.

## 2 METODOLOGIA

Nesta seção será apresentada a metodologia utilizada para aplicar o PC como técnica de ensino-aprendizagem em dois tópicos abordados na disciplina de programação: recursão e pilha de ativação. A metodologia escolhida tem por objetivo proporcionar uma forma de ensino sobre esses dois assuntos de uma forma distinta da habitual, apresentada como exemplos de código e considerada pelos discentes de difícil entendimento e operacionalização. Para esse fim, foram selecionados conceitos do PC juntamente com os conceitos da chamada *computação desplugada*, que visa ensinar computação sem utilização de dispositivos programáveis.

A estratégia escolhida para o desenvolvimento do trabalho foi a execução de uma oficina com carga horária total de 12 horas, sendo 4 horas de atividades presenciais e 8 horas de atividades à distância, ofertadas por meio do ambiente virtual de aprendizagem Moodle. O público alvo foi composto por discentes do curso de Engenharia de Computação, da Universidade Federal do Pampa, Campus Bagé. Como o objetivo geral da oficina foi proporcionar uma nova visão sobre o processo de recursividade e pilha de ativação, a seleção dos discentes foi feita por convite enviado por e-mail a todos os alunos, sem exigências de créditos ou conteúdos cursados; os participantes da oficina poderiam ter níveis diferentes de conhecimento sobre os assuntos abordados. As subseções 2.1 e 2.2 mostram a metodologia aplicada em cada etapa da oficina, a etapa presencial e a etapa à distância.

### 2.1 Atividades Presenciais

As atividades presenciais foram criadas a partir de exemplos clássicos de algoritmos que pudessem explorar o PC por meio da computação desplugada. Os discentes utilizaram itens do cotidiano para ilustrar exemplos clássicos do emprego da técnica de recursão, sem a necessidade da utilização do computador. A primeira atividade desenvolvida foi o problema da *indexação de fila*; a segunda atividade realizada foi *a soma dos primeiros N números inteiros positivos*; a última atividade realizada foi o funcionamento do algoritmo de ordenação de vetores *bubble sort*.

**Problema da indexação de fila -** indexar uma fila consiste em atribuir um índice a cada elemento da fila, de acordo com sua posição em relação ao primeiro elemento da fila. Os discentes foram dispostos em fila, com os olhos vendados e com a mão no ombro da pessoa da frente (dessa forma, o primeiro da fila era o único que sabia que não havia ninguém à sua frente) e precisavam descobrir qual a sua posição na fila apenas questionando a pessoa posicionada a sua frente e somente respondendo para a pessoa posicionada atrás de si. A tarefa consistia na escolha de uma única pergunta que deveria ser respondida pela pessoa imediatamente à sua frente na fila; uma vez escolhida, deveriam realizar o procedimento da pergunta, decidindo como responder para o aluno posicionado atrás de si a partir da resposta do aluno posicionado à sua frente. A recursão foi, dessa forma, exemplificada na escolha da pergunta única a fazer e do cálculo a realizar, com base na posição do aluno anterior, para descobrir a sua posição. Logo após a realização do experimento, os alunos foram divididos em dois grupos, e precisaram construir um pseudocódigo que representasse essa solução.

**Soma dos primeiros *N* números inteiros positivos –** a soma dos N primeiros números inteiros positivos representa o resultado da soma de uma progressão aritmética de razão 1. Ou seja, é o resultado de 1 + 2 + 3 + ... + (N-1) + N ou ainda o processo de adição de todos os números naturais menores ou iguais a N. A execução do exercício foi operacionalizada com a disposição, em uma mesa, de três sequências de cartas: uma com 3 elementos, outra com 4 elementos e a última com 5 elementos. Com base na sequência de cartas, os alunos precisaram recorrer os passos do PC para a construção da solução, que envolvia dizer qual era a soma de todas as sequências com base na soma das sequências anteriores e posterior construção do pseudocódigo correspondente. De posse do pseudocódigo, foi exemplificada como ocorreriam os processos de empilhar e desempilhar as chamadas da função *Soma(N)* na pilha de ativação utilizando caixas de papelão, onde cada caixa adicionada ou retirada da pilha fazia referência chamada ou ao retorno da função.

***Bubble sort* –** o bubble sort é um algoritmo de ordenação de sequências de elementos que possuam alguma relação de ordem definida entre eles, analisando repetidamente os elementos par a par e trocando-os entre si, caso estejam fora da ordem definida. O procedimento é terminado quando não houver nenhuma troca de elementos a ser feita. Para essa atividade, um grupo de cartas foi disposto sobre uma mesa, com os valores desordenados. A partir disso, foi feito o processo de ordenação que o algoritmo realiza, mostrando passo a passo da execução. Observando a execução, os discentes se juntaram em grupos e precisaram permear os passos do PC até construir uma solução abstrata que pudesse ser usada para qualquer sequência. Com a solução definida, os alunos deveriam apresentar um pseudocódigo da solução.

### 2.2 Atividades a Distância

Para a realização das atividades a distância foram propostos quatro problemas: i) cálculo do valor do fatorial de um número; ii) apresentação da sequência de Fibonacci; iii) soma dos elementos de um vetor; e iv) cálculo do máximo divisor comum (MDC) entre 2 números. Para cada um dos problemas, os alunos deveriam empregar os passos do PC para solucionar o problema, até a construção do algoritmo, utilizando o princípio da recursividade. Para o primeiro problema, foi pedido ainda que os alunos construíssem a pilha de ativação para o problema do fatorial do número 4 (*Fatorial(4)*). A última atividade proposta foi identificar um método que utilize recursão para resolver o problema do máximo divisor comum. Para este problema, foi pedido que os alunos descrevessem todo o raciocínio empregado para a solução do problema, utilizando os conceitos do PC.

## 3 RESULTADOS e DISCUSSÃO

A oficina de atividades presenciais foi realizada durante o turno da manhã nas dependências do Campus Bagé da UNIPAMPA. Compareceram quatro discentes, sendo um do terceiro semestre, dois do quinto semestre, e um do sétimo semestre. Acredita-se que o pequeno número de discentes presentes deve-se à forma como o convite foi realizado: o contato pessoal com eles, explicando a atividade, poderia ter motivado um número maior de participantes. O pequeno número de participantes teve impacto no tempo necessário para desenvolver cada atividade, além de prejudicar um pouco a discussão em grupo, visto terem sido divididos em duplas.

Durante as atividades presenciais percebemos que dois discentes do 3º e 5º semestre (grupo A) precisavam de mais esclarecimentos sobre o problema a ser tratado, demonstrando certa dificuldade na abstração do problema. O grupo B (composto por discentes do 5º e 7º semestre) tinham mais facilidade neste processo. A facilidade dos discentes mais adiantados no curso é explicada pelo fato de haverem cursado mais disciplinas do curso de Engenharia de Computação, conhecendo as soluções dos problemas propostos. Os conteúdos de algoritmos são aprofundados ao longo do curso e formam a base da construção do conhecimento em

outros componentes curriculares; dessa forma, a maturidade de um discente no curso se reflete na sua proficiência no desenvolvimento de soluções algorítmicas para problema de forma geral.

A primeira atividade realizada foi a da indexação da fila e os discentes, no início, mostraram-se um pouco receosos com as atividades. Após solucionarem o problema, demonstraram maior receptividade ao testar a solução encontrada, e para as demais atividades propostas posteriormente. No momento da construção do algoritmo, que foi o último passo da atividade, ambos os grupos obtiveram sucesso. Já a segunda atividade, soma dos números de N até 1, começaram aparecer as dificuldades de compreensão por parte dos discentes envolvidos. Durante o desenvolvimento da atividade compreenderam o funcionamento; porém no momento de construção da solução abstrata e do pseudocódigo apresentaram um pouco mais de dificuldade em acompanhar o raciocínio. Após uma breve discussão em grupo, as soluções apresentadas estavam corretas.

A terceira atividade, referente ao algoritmo de ordenação *bubble sort*, era a atividade que exigia um maior nível de abstração para sua compreensão. Após a demonstração do modo de funcionamento do algoritmo, os discentes foram novamente divididos em duplas e instados a construir a solução do problema com base nos passos do PC. Como eles já conheciam alguns algoritmos de ordenação, acabaram utilizando os conhecimentos prévios para a resolução e, dessa forma, um dos grupos acabou retornando o algoritmo de ordenação *insertion sort*. Assim, foi possível notar que o conhecimento prévio acabou interferindo no desenvolvimento da atividade, onde o grupo negligenciou os princípios de funcionamento inicialmente apresentados; a solução, ainda assim, atendeu ao objetivo de ordenar um sequência de elementos com base nos conceitos do PC.

Por conta do pequeno número de alunos e do conhecimento prévio apresentado pelos discentes, o tempo previsto foi maior do que o necessário para a execução das tarefas; por este motivo, a primeira atividade prevista para ser resolvida a distância foi trabalhada presencialmente.

Todo o processo do PC para solucionar o problema do fatorial de um número é muito semelhante à solução do problema da soma que já havia sido resolvido anteriormente, de forma que os dois grupos tiveram êxito na atividade, sem dificuldades. As demais atividades foram desenvolvidas a distância, conforme o planejado. Estas atividades foram avaliadas com base nas entregas realizadas pelos discentes através do Moodle. Para a atividade da soma de vetores, apenas um dos grupos implementou o algoritmo de forma correta, o outro conseguiu gerar um algoritmo muito próximo à solução ideal, porém a condição de parada estava incorreta.

Os envios referentes à sequência de Fibonacci, do ponto de vista da abstração estavam corretos, porém ambos os algoritmos apresentaram erros de implementação: um deles com a condição de parada incorreta e o outro, na implementação da função principal do programa, misturou os conceitos de implementação por laço e por recursão. O último problema, do máximo divisor comum, precisava apresentar toda a descrição do método utilizado, desde o entendimento do problema, até a solução utilizando um algoritmo. Tanto as descrições, como as implementações obtiveram bons resultados, e solucionaram o problema, utilizando um método recursivo.

## 4 CONSIDERAÇÕES FINAIS

O ensino de programação exige certo nível de abstração dos alunos para obter a solução de um problema específico por meio da construção de uma sequência de passos que possam ser usados para a construção da solução de qualquer instância desse mesmo problema e sua posterior tradução em códigos capazes de ser interpretados pelo computador. Isso, sem dúvida, é uma das maiores dificuldades que os discentes encontram, no início da graduação,

nas disciplinas que exigem essa habilidade, visto ser essa uma competência raramente trabalhada no ensino médio ou fundamental. A utilização de métodos alternativos e potencialmente mais eficientes de ensino é uma forma de atenuar esses problemas enfrentados pelos alunos e de obter maior êxito na aprendizagem, com correspondente diminuição da evasão por essa razão específica.

Este trabalho teve como objetivo realizar uma oficina trazendo uma proposta de ensino de dois tópicos da disciplina de programação: a recursão e a pilha de ativação, aos alunos do curso de Engenharia de Computação da Universidade Federal do Pampa, utilizando o PC como forma de ensino-aprendizagem. Após a aplicação da metodologia de ensino proposta, observou-se que os assuntos ``recursividade" e ``pilha de ativação" demandam bastante concentração do estudante, grande capacidade de abstração e principalmente bom conhecimento do problema que está sendo proposto para ser resolvido. Observou-se também que, nas atividades a distância, os alunos enviaram soluções para os desafios propostos com muitos problemas de implementação, o que demonstra que os algoritmos desenvolvidos não foram testados previamente. Com isso, pode-se concluir que um acompanhamento continuado dos alunos teria melhor efetividade, principalmente, se fosse realizado em paralelo à disciplina que trata deste tópico. Também poderá propiciar uma melhor avaliação do quanto o método é realmente efetivo no ensino de recursividade em relação ao método tradicional de ensino.

Por fim, percebe-se que houve melhora na capacidade dos estudantes de interpretar e abstrair os problemas após a utilização da metodologia de ensino-aprendizagem proposta. Na sua autoavaliação, realizada presencialmente, os estudantes afirmaram ter um melhor entendimento sobre os conteúdos abordados do que tinham antes da participação na oficina.

Como trabalhos futuros, deseja-se replicar este experimento em uma quantidade maior de discentes, se possível durante a disciplina que aborda os conceitos de recursividade e pilha de ativação. Com isso, será possível estimar a efetividade do PC quanto metodologia de ensino-aprendizagem de programação.

## REFERÊNCIAS


FURTADO, Vanessa Viégas Alves; ALVES, Tiago Wickstrom. Fatores determinantes da evasão universitária: uma análise com alunos da Unisinos. **Contextus**, v. 10, n. 2, 2012.

GOMES, Anabela; HENRIQUES, Joana; MENDES, António. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. **Educação, Formação & Tecnologias-ISSN 1646-933X**, v. 1, n. 1, p. [93-103], 2008.

PROGRAD. Pró-Reitoria de Graduação: graduação em números, 2018. Disponível em: <http://porteiras.r.unipampa.edu.br/portais/prograd/graduacao-em-numeros/>. Acesso em: 28 Ago. 2018.

SILVA FILHO, Roberto Leal Lobo et al. A evasão no ensino superior brasileiro. **Cadernos de pesquisa**, v. 37, n. 132, p. 641-659, 2007.

WING, Jeannette M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33-35, 2006.