

EXERCÍCIOS SIGNIFICATIVOS PARA ENSINO DE ALGORITMOS E LÓGICA DE PROGRAMAÇÃO EM CURSO DE GRADUAÇÃO

Vladimir Marques Erthal ¹

RESUMO

Contexto: O conhecimento de algoritmos e lógica de programação é a base para a programação de computadores. Por outro lado, as Metodologias Ativas (MA) têm se mostrado efetivas para facilitar a absorção do conhecimento por meio de um processo ensino-aprendizagem significativo. Porém, o ensino dos conceitos básicos de programação de computadores ao nível de graduação ainda se vale em grande parte de aulas expositivas e exercícios matemáticos, que não contribuem para o aprendizado de uma disciplina que já possui barreiras e dificuldades naturais. Objetivo: Apresentar uma proposta de aulas para uma disciplina de algoritmos e lógica de programação fundamentada em exercícios práticos baseados em problemas cotidianos e em jogos, utilizando conceitos de MA. Método: Foram desenvolvidos exercícios de acordo com cada conceito ensinado, aplicados em sala de aula para uma turma de primeiro período na graduação em Sistemas de Informação do CEFET/RJ, campus Maria da Graça. Os exercícios foram implementados utilizando Portugol Studio, um compilador para fins didáticos. Resultados: A utilização dessa abordagem de aula resultou em boa absorção do conteúdo, medido por meio de exercícios teóricos e práticos. Além disso, foi alto o índice de aprovação dos alunos, assim como o índice de satisfação dos mesmos quanto ao método de ensino e ao entendimento do conteúdo, medido qualitativamente e quantitativamente por meio de questionários enviados ao final de cada aula e ao final da disciplina. Conclusão: A utilização de exercícios práticos, baseados em problemas cotidianos e em jogos, para o ensino de algoritmos e lógica de programação facilita o entendimento do conteúdo e do pensamento estruturado. O presente trabalho pode ser utilizado como estudo de caso, oferecendo diversos exercícios como exemplos de aplicação que podem ser adaptadas para outros contextos, tendo obtido resultados satisfatórios tanto para o aprendizado quanto para a satisfação dos alunos.

Palavras-chave: Metodologias Ativas, Lógica de Programação, Algoritmos, Ensino, Aprendizado Significativo.

INTRODUÇÃO

O ensino de algoritmos e lógica de programação é uma disciplina fundamental nos cursos de Computação, pois ela é a base para as demais disciplinas de programação, e o seu bom aprendizado pode conduzir ao desenvolvimento de produtos de software de qualidade (DOS SANTOS e COSTA, 2006). Porém, muitos alunos possuem grande dificuldade com essa disciplina, o que é refletido nos altos índices de reprovação e abandono.

¹ Mestre em Engenharia de Software, professor do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ), vladimir.erthal@cefet-rj.br.

Dentre os motivos expostos na literatura técnica para essa dificuldade, podemos listar: dificuldade de entendimento dos conceitos-chave, turmas muito grandes, baixa autoconfiança e desmotivação dos alunos, dissociação entre os exercícios propostos e o mundo real, interpretação dos problemas computacionais, e falta de tempo para compreensão dos conceitos (MORAIS et al., 2020). Desses, destaco a aplicação de exercícios com temática matemática, apenas para demonstração de conceitos, mas que não produzem resultados significativos, presente em grande parte dos livros didáticos utilizados na disciplina, o que produz dificuldade de entendimento e desmotivação.

Com o propósito de facilitar a aprendizagem, o professor deve adotar métodos de ensino mais práticos e dinâmicos, que despertem a atenção e o interesse dos alunos, com ênfase no pensamento crítico que possibilite o entendimento da teoria a partir da prática, tornando os conceitos menos abstratos. Além disso, as atividades propostas devem ser baseadas em situações do cotidiano para estimular o raciocínio lógico, com questões-desafio para desenvolver agilidade e criatividade (DOS SANTOS e COSTA, 2006).

Com base nessas dificuldades e necessidades, o uso de metodologias ativas pode ser uma boa abordagem a ser adotada na disciplina de Algoritmos e Lógica de Programação. Metodologias ativas são conceitos e orientações que buscam organizar o processo de ensino e aprendizagem através de estratégias didáticas centradas nos alunos, estimulando a sua proatividade e criatividade, possibilitando assim a construção do seu próprio conhecimento, tornando-os protagonistas desse processo por meio da reflexão sobre o conteúdo, e não pela necessidade de decorá-lo (DIESEL et al., 2017) (MORAN, 2018). Assim, o conteúdo didático deve ser apresentado de forma relevante, estimulante, curiosa e desafiante (LOVATO et al., 2018).

Este trabalho é resultado da aplicação de metodologias ativas em uma turma da disciplina de Algoritmo Estruturado do curso de bacharelado em Sistemas de Informação do CEFET/RJ, campus Maria da Graça. O objetivo da aplicação das metodologias ativas nessa turma era colocar os alunos em contato com o conteúdo programático de forma reflexiva por meio de atividades práticas e significativas, estimulando o questionamento e a colaboração.

Os resultados obtidos foram positivos, tanto quantitativamente quanto qualitativamente, e confirmam os benefícios da aplicação das metodologias ativas no contexto específico onde foram executadas.

METODOLOGIA

Este trabalho foi desenvolvido a partir da experiência da aplicação de estratégias de metodologias ativas como forma de condução das aulas da disciplina de Algoritmo Estruturado (pertencente ao primeiro período) durante um semestre no curso de bacharelado em Sistemas de Informação do CEFET/RJ, campus Maria da Graça. O período de avaliação foi o primeiro semestre de 2024, e a turma possuía 43 alunos.

Para avaliar os resultados, foram comparados o percentual de aprovação e a média geral da disciplina com o registro histórico da mesma desde a criação do curso, no segundo semestre de 2018.

REFERENCIAL TEÓRICO

O ensino de algoritmos e lógica de programação deve ser realizado privilegiando os conceitos e estruturas ao invés de detalhes de sintaxe. As aulas devem ter como objetivo desenvolver o raciocínio lógico de modo a possibilitar a criação de habilidades de entendimento e resolução de problemas por meio de soluções computacionais (DOS SANTOS e COSTA, 2006) (MAEKAWA et al., 2015).

Neste artigo, essa disciplina foi ministrada com base nas metodologias ativas. Ao contrário das metodologias tradicionais de ensino, baseadas em aula expositiva centrada no professor e a avaliação por meio de uma prova ao final de um determinado período, as metodologias ativas orientam o processo ensino-aprendizagem tornando o aluno o seu protagonista por meio de atividades reflexivas, relevantes, estimulantes e desafiantes (MORAN, 2018), ao invés de basear o estudo em memorização e gerar tensão pelo processo tradicional de avaliação (DOS SANTOS e COSTA, 2006). Com isso, ele desenvolve novas competências, como iniciativa, criticidade, criatividade, reflexão, responsabilidade, flexibilidade, ética e sensibilidade no trato com outros colegas (LOVATO et al., 2018). Assim, quando ele precisar aplicar o conhecimento fora da instituição de ensino, terá maior capacidade de analisar, abstrair, comunicar e buscar soluções factíveis em contextos imprevisíveis e incertos (DIESEL et al., 2017), necessitando apenas relembrar o conhecimento aprendido, e não reconstruí-lo.

O papel do professor nessa abordagem deixa de ser o de detentor e transmissor do conhecimento para ser o de curador dos materiais mais adequados para que os alunos estudem de forma independente, da preparação das atividades práticas que possibilitarão a construção do conhecimento, e de orientador dos alunos durante a execução dessas atividades (BARBOSA e DE MOURA, 2013). As atividades práticas devem ser

planejadas de preferência para serem realizadas em duplas ou em pequenos grupos, de modo a gerar boa convivência, reconhecimento mútuo, solução de dúvidas e capacidade de expressar opiniões (LOVATO et al., 2018) (DIESEL et al., 2017). A prática social ajuda a preparar os alunos para situações do mundo do trabalho onde eles precisarão trabalhar em conjunto, e não de maneira isolada (MORAN, 2018). A colaboração entre os alunos em disciplinas de programação foi utilizada com sucesso em outros casos relatados na literatura técnica, ressaltando a “forte interação entre os componentes do grupo e o prazer em dividir o conhecimento adquirido” (DOS SANTOS e COSTA, 2006).

As metodologias ativas podem ser aplicadas com ou sem o apoio de tecnologias digitais. Porém, o seu uso potencializa esse processo (FERRARINI et al., 2019), principalmente por permitir a utilização de vídeos, textos e ferramentas disponíveis no ambiente virtual, o que otimiza o tempo do professor, reduz tarefas repetitivas e permite que ele se concentre na elaboração de atividades criativas e estimulantes, bem como na orientação mais próxima dos alunos (MORAN, 2018).

Entretanto, metodologias ativas não são uma única forma de conduzir o processo de ensino e aprendizagem. Elas se materializam na forma de diferentes estratégias que podem ser utilizadas de acordo com a melhor adequação a cada contexto (FERRARINI et al., 2019). Neste estudo de caso, foram utilizadas três estratégias ativas: Sala de Aula Invertida, *Peer Instruction* e Atividade Supervisionada em Par. Esta última foi criada pelo autor deste artigo, com o objetivo de estruturar as atividades práticas de programação.

A Sala de Aula Invertida foi criada entre 2007 e 2008 nos EUA (FERRARINI et al., 2019), mas suas bases remontam a um artigo de 1993, onde é proposta a utilização do tempo em sala de aula para o processamento das informações, ao invés da transmissão do conhecimento (MORAN, 2018). Assim, essa estratégia consiste no estudo individual da matéria antes da aula, por meio de vídeos, textos ou atividades fornecidos pelo professor, seguido por exercícios de fixação para garantir a compreensão do conteúdo. Na sala de aula, ou em laboratório, são realizadas atividades de aplicação, recuperação do conhecimento, ampliação ou discussão do conteúdo previamente estudado, com um retorno imediato sobre o desempenho dos alunos. Além disso, tanto as atividades remotas quanto as de aula são computadas na nota final dos alunos, de modo a estimular a participação. No ensino tradicional, o aluno realiza as atividades em casa após uma aula expositiva, o que pode gerar dúvidas que não terão como serem respondidas no momento de necessidade. Assim, na Sala de Aula Invertida, o professor pode trabalhar as dúvidas e as dificuldades dos alunos por meio de atividades significativas e supervisionadas,

fornecendo o apoio necessário no momento correto (VALENTE, 2014). Já as atividades teóricas realizadas em casa possibilitam ao aluno rever o material quantas vezes ele precisar, o que não é possível na aula tradicional (FERRARINI et al., 2019).

O *Peer Instruction* foi criado nos anos 1990 nos EUA (FERRARINI et al., 2019), e tem como objetivo o envolvimento dos alunos através de discussões em grupos, onde eles se ajudam mutuamente para construir o conhecimento sobre determinado tema (LOVATO et al., 2018). Essa prática é mais efetiva para estimular a participação de todos os alunos do que a realização de perguntas durante uma aula expositiva, que geralmente envolve apenas alguns poucos alunos mais motivados (GODOI e FERREIRA, 2017). Pode ser realizada com a utilização de perguntas abertas (discursivas), porém, torna-se mais produtiva a utilização de perguntas objetivas (múltipla escolha). Assim, é mais rápida a identificação do percentual de alunos que responderam corretamente e, caso o percentual de acertos seja baixo, o professor tem a possibilidade de retomar os conceitos para melhorar o entendimento ou propor a discussão das respostas em grupos (FERRARINI et al., 2019).

RESULTADOS E DISCUSSÃO

A base das metodologias ativas aplicada na disciplina foi a Sala de Aula Invertida. Ao final de todas as aulas, era enviado para o grupo da turma, por meio da ferramenta Microsoft Teams, um questionário que deveria ser respondido individualmente, composto por um ou mais vídeos selecionados pelo professor juntamente com algumas questões de fixação. Os questionários foram criados na ferramenta Microsoft Forms e os vídeos foram selecionados do YouTube por meio de avaliação do seu conteúdo, que deveria ser de fácil compreensão e possuir o tema que seria trabalhado na aula seguinte. As questões eram sobre o conteúdo dos vídeos, sendo algumas objetivas (múltipla escolha) e uma de codificação. Esta era planejada para possibilitar apenas a aplicação rápida dos conceitos expostos nos vídeos.

A estratégia da Sala de Aula invertida prescreve que o tempo em sala seja utilizado para que a turma trabalhe de maneira prática, aplicando os conceitos previamente estudados. Assim, nesse tempo foram aplicadas atividades utilizando as estratégias de metodologias ativas *Peer Instruction* e Atividade Supervisionada em Par. Cada aula possuía quatro horas de duração e foi planejada para possuir a aplicação de uma ou mais atividades. As atividades que utilizaram *Peer Instruction*, por ser uma estratégia que

incentiva a discussão, tinham como objetivo refletir sobre conceitos e suas consequências, por meio de formulários a serem preenchido no Microsoft Forms.

As atividades que utilizaram a estratégia Atividade Supervisionada em Par eram práticas, nas quais os alunos deveriam desenhar fluxogramas ou desenvolver programas na linguagem Portugol com a sintaxe do Portugol Studio. Essa linguagem foi escolhida por sua sintaxe ser de fácil entendimento e mais parecida com a linguagem C, que é a linguagem estudada no segundo período do curso. Os alunos poderiam desenvolver os programas utilizando o compilador online² ou por meio do software Portugol Studio³ instalado nos computadores. A estratégia ativa Atividade Supervisionada em Par foi criada por este autor, e seus princípios estão descritos na Tabela 1.

Conceito	Aplicação
Trabalho em pares	Todas as atividades práticas poderiam ser realizadas por dois alunos, de modo que um auxiliasse o outro, e os dois pensassem juntos na forma de resolver o problema. A discussão sobre o conteúdo facilita o aprendizado.
Tema significativo	Sempre que possível, o tema escolhido para a atividade era baseado na vida dos alunos fora do ambiente de ensino ou no mundo do trabalho. Assim, eram evitados temas matemáticos ou que produziam apenas um resultado genérico da aplicação dos conceitos, tradicionalmente comuns no ensino de ciência da computação.
Construção do conhecimento	As atividades da aula eram planejadas de forma que o conteúdo previamente estudado de maneira remota pudesse ser lembrado e construído novamente ao desenvolver a solução do exercício. Os principais conceitos poderiam ser reapresentados de forma resumida pelo professor antes ou depois da atividade, dependendo da necessidade identificada por ele na turma.
Construção desafiante	Em algumas aulas, os temas das últimas atividades da aula eram planejados para possuírem maior grau de desafio, de forma que os alunos precisassem pensar além do conceito a ser aprendido na aula, utilizando-o como parte da solução, mas não como o objetivo dela. Um exemplo eram as atividades que solicitavam que os alunos desenvolvessem jogos, pois eles precisariam pensar em toda a mecânica do jogo, além de aplicar os conceitos estudados.
Solução de dúvidas	Por estarem realizando as atividades em sala de aula, quando surgiam dúvidas que a dupla não conseguia resolver, o professor auxiliava os alunos, lembrando algum conceito esquecido ou fornecendo possibilidades de caminhos a serem seguidos. Quando o professor identificava que diversas duplas estavam com a mesma dúvida, a atividade era congelada para fornecer uma explicação para toda a turma, que retomava a atividade após o entendimento.

Tabela 1. Conceitos e aplicações da Atividade Supervisionada em Par.

² <https://portugol.dev/>

³ O *download* do compilador pode ser feito no endereço <https://univali-lite.github.io/Portugol-Studio/>

As atividades práticas de cada aula foram planejadas para iniciarem com o desenvolvimento de programas mais simples e terminarem com programas mais desafiadores. Muitas atividades aplicadas envolviam a criação de jogos, o que estimulava os alunos a pensarem na melhor maneira de utilizar os conceitos aprendidos para implementar todas as possibilidades que o jogo deveria permitir. Além disso, as atividades de desenvolvimento não forneciam o roteiro para a criação do programa, apenas seus requisitos básicos, de modo que os alunos precisavam raciocinar e discutir mais para decidir a forma de implementação. O professor auxiliava cada dupla quando alguma não conseguia desenvolver esse raciocínio. Em algumas aulas, existiam atividades extras, que poderiam ser o desenvolvimento de outro programa ou a melhoria de um dos programas já desenvolvidos. Essas atividades não eram obrigatórias, mas os alunos que as fizessem de forma correta ganhavam pontos adicionais em sua nota. A Tabela 2 mostra todas as atividades aplicadas na disciplina.

Tema da Aula	Estratégia	Descrição da Atividade
Conceituação de lógica, programação estruturada, algoritmos e fluxogramas	PI	Discutir o conceito de lógica, escrever uma definição com a qual a dupla concorde e apresentar para a turma.
	PI	Discutir as vantagens da programação estruturada com base em questões de Verdadeiro ou Falso.
	PI	Discutir e escrever o conceito de algoritmo e sua percepção na sociedade.
	PI	Dado um fluxograma de aprovação da faculdade, discutir e responder perguntas sobre o que ele representa e sobre a simbologia utilizada.
Fluxogramas e variáveis	ASP	Desenhar um fluxograma que represente um programa que gere o resultado de uma eleição com dois candidatos, mostrando apenas a quantidade de votos.
	ASP	Continuar o fluxograma anterior, acrescentando a leitura dos nomes e votos de cada candidato.
	ASP	Continuar o fluxograma anterior, acrescentando o cálculo do percentual de votos em cada candidato.
	ASP	Continuar o fluxograma anterior, acrescentando a identificação do vencedor.
	ASP	Continuar o fluxograma anterior, mudando para três candidatos.
Fluxogramas; operadores aritméticos e relacionais; variáveis	ASP	Desenhar um fluxograma que represente um programa que permita que eleitores votem em candidatos, apenas lendo e registrando a identidade do eleitor e liberando a cabine de votação.
	ASP	Continuar o fluxograma anterior, acrescentando a espera pelo próximo eleitor ou a finalização do programa.
	ASP	Continuar o fluxograma anterior, acrescentando a apresentação dos candidatos e a leitura e registro do voto.
	ASP	Continuar o fluxograma anterior, acrescentando uma restrição para impedir que o mesmo eleitor voto novamente.
Leitura e escrita em Português; estrutura condicional “se-senão”	ASP	Criar um programa que gere o resultado de uma eleição com dois candidatos, lendo os nomes e votos de cada candidato e mostrando a mesma informação.
	ASP	Continuar o programa anterior, acrescentando o cálculo do percentual de votos em cada candidato.
	ASP	Continuar o programa anterior, acrescentando a identificação do vencedor.
	ASP	Continuar o programa anterior, mudando para três candidatos.
Tipos de dados	ASP	Criar um jogo de par ou ímpar.
	ASP	Criar um programa que receba as notas de um aluno e informe sua aprovação ou reprovação.

	ASP	Criar um programa que receba as quantidades de medalhas obtidas por três países nas Olimpíadas e mostre as suas classificações.
	ASP	Criar um programa para uma loja de departamentos que calcule o valor a ser pago de acordo com o parcelamento escolhido, com desconto ou acréscimo.
Estrutura condicional “escolha”	ASP	Criar um jogo de pedra, papel ou tesoura.
	ASP	Criar um programa de calculadora que receba dois números, uma operação e exiba o resultado.
	ASP	Criar um programa para uma lanchonete que mostre o cardápio, pergunte o sanduíche, o acompanhamento e a bebida desejados e calcule o valor total.
	ASP	Criar um programa para uma loja de aluguel de carros que calcule o valor devido de acordo com o tipo de carro, a quantidade de dias e a quilometragem.
Estrutura de repetição “para”	ASP	Criar um jogo do tipo “quiz”.
	ASP	Criar um programa que gere a tabuada de um número informado.
	ASP	Criar um programa de crítica de cinema que receba notas e tipos de dez filmes e calcule o percentual de tipos e a média geral das notas.
Estrutura de repetição “enquanto”	ASP	Criar um programa para mostrar o vencedor de eleição com dez candidatos.
	ASP	Criar um programa para uma lanchonete que mostre o cardápio e permita a escolha de uma quantidade indeterminada de itens, calculando o valor total.
	ASP	Criar um programa para processamento de uma pesquisa sobre altura e peso.
	ASP	Criar um jogo de adivinhação de números gerados aleatoriamente.
Revisão	ASP	Criar a primeira parte de um jogo de RPG no qual um cavaleiro vai ao resgate de uma princesa sequestrada por um dragão. Nessa parte, são definidos os atributos do personagem principal.
	ASP	Continuar o jogo de RPG, acrescentando uma luta inicial contra inimigos que protegiam a caverna do dragão.
	ASP	Continuar o jogo de RPG, acrescentando a luta contra o dragão.
	ASP	Atividade extra. Continuar o jogo de RPG, melhorando a mecânica de algumas partes do jogo.
Estrutura condicional “faça-enquanto”	ASP	Criar um programa para controlar uma seção eleitoral.
	ASP	Criar um programa para calcular a folha de pagamento de uma empresa.
	ASP	Criar um jogo de memorização de números.
	ASP	Atividade extra. Criar um jogo de força.
Vetores	ASP	Criar um programa para calcular o resultado de uma eleição com dez candidatos, exibindo os percentuais de votos válidos.
	ASP	Criar um jogo de zero ou um.
	ASP	Criar um programa de exibição da tabuada dos números de um a dez.
Vetores	ASP	Criar um programa que receba as quantidades de medalhas obtidas por dez países nas Olimpíadas e mostre as suas classificações.
	ASP	Criar um programa para embaralhar cartas de um baralho.
	ASP	Criar um programa para receber e analisar temperaturas em um período.
Matrizes	ASP	Criar um programa de crítica de cinema que receba notas de cinco críticos para dez filmes e calcule suas médias.
	ASP	Criar um Jogo da Velha.
	ASP	Criar um jogo da memória.
Matrizes	ASP	Criar um programa de reserva de quartos de um hotel.
	ASP	Criar um jogo de cartas do tipo Paciência.
Funções	ASP	Criar um jogo de RPG no qual um personagem precisa mover-se em um mapa e chegar até o seu destino.
	ASP	Atividade extra. Melhorar a mecânica do jogo de RPG criado.

Tabela 2. Descrição das atividades realizadas na disciplina Algoritmo Estruturado. Na coluna “Estratégia”, PI significa *Peer Instruction* e ASP significa Atividade Supervisionada em Par.

Os resultados dessa abordagem foram medidos de forma quantitativa e qualitativa. Os resultados quantitativos foram obtidos por meio de comparação entre as notas finais dos alunos e os dados históricos da disciplina desde a criação do curso, no segundo período de 2018. Esses dados podem ser vistos na Tabela 3.

Período	Média % de Aprovados	Média % de Reprovados por Nota	Média % de Reprovados por Frequência
2018.2	64%	12%	24%
2019.1	55%	6%	39%
2019.2	57%	6%	37%
2020.1	68%	4%	28%
2020.2	65%	35%	0%
2021.1	89%	11%	0%
2021.2	76%	24%	0%
2022.1	65%	3%	32%
2022.2	83%	0%	17%
2023.1	62%	14%	24%
2023.2	49%	20%	31%
2018.2 a 2023.2	67%	12%	21%
2024.1	87%	0%	13%

Tabela 3. Comparação entre as médias de aprovações com as suas médias históricas.

Na disciplina analisada não houve reprovação por nota, igual apenas ao período 2022.2. A reprovação por frequência foi maior apenas do que três períodos nos quais isso não ocorreu, e a taxa de aprovação foi a segunda maior do histórico, abaixo apenas de 2021.1. Analisando a média dos períodos anteriores, os dados atuais foram melhores nas três categorias. A avaliação qualitativa, sumarizada na Tabela 4 e na Tabela 5, ajuda a entender os motivos desses dados positivos.

	% Negativa	% Neutra	% Positiva
Avaliação da Aula	2%	12%	86%
Autoavaliação	1%	19%	80%

Tabela 4. Média das avaliações respondidas pelos alunos ao final de cada aula.

As avaliações de aula foram realizadas de forma individual pelos alunos após cada aula. Os questionários eram sempre compostos por duas partes: uma avaliação da aula e uma autoavaliação sobre o aprendizado do aluno. A avaliação da aula continha itens como a didática do professor, o apoio do professor nas atividades, a dinâmica geral da aula, a atividade remota e cada atividade realizada em sala. Os alunos deveriam classificar cada item em “achei péssimo”, “achei ruim”, “achei ok”, “gostei” e “gostei muito”. Para sumarizar os resultados, as duas primeiras opções foram reunidas em “% Negativa”, a terceira tornou-se “% Neutra” e as duas últimas “% Positiva”. Já a autoavaliação continha itens relativos ao conteúdo programático da aula, que os alunos deveriam classificar em “não entendi nada”, “ainda tenho dificuldades”, “entendi, mas preciso praticar mais” e

“entendi bem”. Na sumarização, a primeira opção tornou-se “% Negativa”, a segunda “% Neutra” e as duas últimas foram reunidas em “ Positiva”.

Item avaliado	Avaliação da Disciplina		
	% Negativa	% Neutra	% Positiva
Didática do professor	0%	4%	96%
Apoio do professor nas atividades	0%	8%	92%
Qualidade das atividades	4%	4%	92%
Atividades remotas	0%	4%	96%
Dinâmica das aulas	0%	8%	92%
Aprendizado da matéria	0%	5%	95%

Tabela 5. Média das avaliações respondidas pelos alunos ao final da disciplina.

As avaliações ao final da disciplina foram realizadas de forma individual, e continham questões sobre a avaliação das aulas de forma geral, seguindo o mesmo modelo de respostas e síntese apresentados na avaliação das aulas. Também continha questões discursivas sobre os pontos fortes e fracos das aulas. Em sua quase totalidade, as respostas foram bastante elogiosas. Nos pontos fracos, alguns alunos reclamaram do tamanho da turma, o que dificultava o pronto atendimento das dúvidas durante as atividades em sala. Dentre as dificuldades identificadas por (MORAIS et al., 2020) nesse contexto, o tamanho da turma foi a única dificuldade encontrada, as demais aparentam terem sido resolvidas pela aplicação da metodologia.

Já nos pontos positivos, muitos alunos ressaltaram a boa dinâmica das aulas, o aprendizado por meio de exercícios práticos em sala, a qualidade de absorção do conteúdo, a criatividade das atividades, o apoio dado pelo professor e a formação das notas. Destaco o seguinte comentário de um aluno, que resumiu o exato objetivo da aplicação da metodologia: *“Visto que a metodologia de aprendizado por prática foi muito efetiva para o entendimento dos conceitos básicos, será mais fácil de lembrar tais conceitos da matéria no futuro.”*

Analisando esses resultados, chega-se à conclusão que a aplicação das metodologias ativas com atividades práticas relevantes provou-se efetiva tanto no percentual de aprovação dos alunos quanto na percepção deles sobre o seu próprio aprendizado. O objetivo de possibilitar a construção do conhecimento de forma independente para cada aluno foi atingido satisfatoriamente, bem como o estímulo à atuação em parceria com seus pares de forma proativa, criativa e reflexiva.

Nos próximos períodos, pretendo aplicar novamente a metodologia, buscando a melhoria contínua do processo, das atividades e as adaptações aos diferentes públicos.

Entretanto, os resultados aqui apresentados são específicos deste estudo de caso, e não podem ser generalizados para outros contextos. Cada aplicação das metodologias ativas possui especificidades que serão refletidas nos seus próprios resultados.

CONSIDERAÇÕES FINAIS

As metodologias ativas são uma abordagem didática que tem como objetivo a melhoria do processo ensino-aprendizagem por meio do deslocamento da centralidade das aulas do professor para os alunos, tornando-os protagonistas da sua própria construção do conhecimento. Diversas estratégias que utilizam os princípios das metodologias ativas podem ser utilizadas para atingir esse propósito. Neste trabalho, foi apresentado um relato de aplicação dessa abordagem em uma turma da disciplina de Algoritmo Estruturado na graduação em Sistemas de Informação do CEFET/RJ.

As estratégias utilizadas foram a Sala de Aula Invertida, a Atividade Supervisionada em Par e o *Peer Instruction*, obtendo resultados positivos. Nas avaliações quantitativas, as médias de aprovação foram maiores do que os dados histórico desde a criação do curso, no segundo semestre de 2018. Nas qualitativas, a percepção média dos alunos sobre cada aula foi 86% positiva, e a média das autoavaliações de cada aula foi 80% positiva. Na avaliação ao final das disciplinas, 96% dos alunos consideraram positiva a didática do professor, 92% acharam positivo o apoio do professor nas atividades, 92% elogiaram a qualidade das atividades realizadas, 96% aprovaram as atividades realizadas remotamente e 92% disseram ter sido boa ou muito boa a dinâmica das aulas. Além disso, 95% dos alunos consideraram que aprenderam a matéria de forma satisfatória.

Apesar de não poder ser generalizado, esses resultados mostram que a aplicação das estratégias de metodologias ativas utilizadas funcionam, especialmente o uso de atividades práticas significativas, e podem trazer os benefícios esperados para um processo ensino-aprendizagem mais aprofundado, reflexivo e efetivo. Novas aplicações da metodologia serão realizadas nos próximos semestres, buscando melhorar e adaptar as atividades a novos contextos, e validar estes resultados obtidos.

AGRADECIMENTOS

A apresentação deste trabalho no X CONEDU foi possível graças ao apoio do CEFET/RJ, campus Maria da Grça.

REFERÊNCIAS

- BARBOSA, E. F.; DE MOURA, D. G.. Metodologias ativas de aprendizagem na educação profissional e tecnológica. **Boletim Técnico do Senac**, v. 39, n. 2, p. 48-67, 2013.
- DIESEL, A.; BALDEZ, A. L. S.; MARTINS, S. N.. Os princípios das metodologias ativas de ensino: uma abordagem teórica. **Revista Thema**, v. 14, n. 1, p. 268-288, 2017.
- DOS SANTOS, R. P.; COSTA, H. A. X.. Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos iniciantes em Computação e Informática. **INFOCOMP Journal of Computer Science**, v. 5, n. 1, p. 41-50, 2006.
- FERRARINI, R.; SAHEB, D.; TORRES, P. L.. Metodologias ativas e tecnologias digitais: aproximações e distinções. **Revista Educação em Questão**, v. 57, n. 52, 2019.
- GODOI, A. F.; FERREIRA, J. V.. Metodologia ativa de aprendizagem para o ensino em administração: relatos da experiência com a aplicação do peer instruction em uma instituição de ensino superior. **REA-Revista Eletrônica de Administração**, v. 15, n. 2, p. 337 a 352, 2017.
- LOVATO, F. L.; MICHELOTTI, A.; DA SILVA LORETO, E. L.. Metodologias ativas de aprendizagem: uma breve revisão. **Acta Scientiae**, v. 20, n. 2, 2018.
- MAEKAWA, C.; NAGAI, W.; IZEKI, C.. Relato de Gamificação da disciplina Projeto e Análise de Algoritmos do curso de Engenharia de Computação. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. 2015. p. 1425.
- MORAIS, C. G. B.; NETO, F. M. M.; OSÓRIO, A. J. M.. Dificuldades e desafios do processo de aprendizagem de algoritmos e programação no ensino superior: Uma revisão sistemática de literatura. **Research, Society and Development**, v. 9, n. 10, p. e9429109287-e9429109287, 2020.
- MORAN, J.. Metodologias ativas para uma aprendizagem mais profunda. **Metodologias ativas para uma educação inovadora: uma abordagem teórico-prática**. Porto Alegre: Penso, p. 02-25, 2018.
- VALENTE, J. A.. Blended learning e as mudanças no ensino superior: a proposta da sala de aula invertida. **Educar em revista**, n. spe 4, p. 79-97, 2014.