

# Utilização da Plataforma Beecrowd de Maratona de Programação como Estratégia para o Ensino de Algoritmos

Allan Kássio Beckman Soares da Cruz<sup>1</sup>, Carlos de Salles Soares Neto<sup>1</sup>,  
Pamela Torres Maia Beckman da Cruz<sup>2</sup> Mário Antonio Meireles Teixeira<sup>1</sup>,

<sup>1</sup>DCCMAPI - Programa de Pós-Graduação Doutorado em Ciência da Computação  
Associação UFMA-UFPI – Universidade Federal do Maranhão (UFMA)  
São Luís – Maranhão – Brasil

<sup>2</sup>Doutoramento em Ciência da Informação - Faculdade de Letras  
Universidade de Coimbra  
Coimbra – Portugal

allankassio@gmail.com, {carlos.salles,mario.meireles}@ufma.br,  
pamela.cruz@student.fl.uc.pt

**Abstract.** *This paper presents a model of a teaching plan based on the platform for programming marathons Beecrowd and active methodologies. The use of game based learn, during the course, consists of the application of the flipped classroom, supported by the basic content of the Algorithms discipline. This strategy aims to cover all the content and allow an efficient evolution of students with a smaller learning curve. The proposed teaching plan aims to help the teacher in the application of active methodologies. It is expected that with this model plan the teachers of Algorithms courses can adopt strategies for the use of active methodologies that create possibilities of interaction between students and learning.*

**Keywords:** *teaching algorithms, game-based learn, programming marathon.*

**Resumo.** *Este trabalho apresenta um modelo de plano de ensino pautado na plataforma para maratonas de programação Beecrowd e em metodologias ativas. O uso de maneira gamificada, no decorrer da disciplina, consiste na aplicação da sala de aula invertida, apoiada no conteúdo básico da disciplina de Algoritmos. Essa estratégia busca abranger todo o conteúdo e possibilitar uma evolução eficiente dos alunos com uma curva de aprendizagem menor. O plano de ensino proposto visa auxiliar o professor na aplicação das metodologias ativas. É esperado que com esse modelo de plano os professores das disciplinas de Algoritmos possam adotar estratégias de utilização de metodologias ativas que criem possibilidades de interação entre os alunos e o aprendizado.*

**Palavras-chave:** *ensino de algoritmos, gamificação, maratona de programação.*

## 1. Introdução

O desenvolvimento de planos de ensino para disciplinas de algoritmos em cursos superiores ainda hoje são realizados com bases em práticas realizadas desde a concepção dos cursos de tecnologia da informação. Em geral esses planos apóiam a prática docente tradicional e não estão alinhados com as necessidades dos alunos, cada vez mais conectados e com muita informação rápida ao alcance.

As disciplinas de algoritmos e programação são fundamentais para a sequência em qualquer curso superior na área de informática. Elas também consistem em um grande obstáculo, sendo causa de altos índices de reprovação e evasão [Lindemann et al. 2008]. Essas disciplinas exigem dos alunos habilidades em lógica, resolução de problemas e capacidade de abstração. [Raabe and Silva 2005] citam a abstração como um dos principais fatores da desmotivação e do elevado grau de desistência dos alunos.

Sendo assim, melhorar o processo de capacitação para a abstração, para a modelagem e resolução de problemas, pode ser um caminho para que além do alcance da motivação desses alunos, eles se tornem melhores profissionais em programação e possam ter mais facilidade ao longo do curso, reduzindo a evasão. Para isso são necessárias metodologias que contribuam de forma efetiva para a melhoria do processo de ensino-aprendizagem destes alunos. Elas devem estar inseridas desde o planejamento da disciplina, para que o professor possa oferecer aos alunos opções de aprendizagem.

Tendo como base planos de ensino-aprendizagem de algoritmos e as dificuldades de aprendizado dos alunos, além da utilização de metodologias ativas que buscam potencializar a aprendizagem ativa dos alunos, este trabalho propõe um modelo de plano de ensino que visa auxiliar o professor na utilização dessas metodologias ativas.

## **2. Metodologias Ativas**

Em sua pesquisa, [Calderon Ribeiro and Passos 2020] questionam acerca da educação formal como podemos crescer para alcançar relevância e ao mesmo tempo possibilitar que todos consigam aprender e moldar seus projetos de vida. De acordo com [Romero-García et al. 2020] os processos de organizar o currículo, as metodologias, os tempos e os espaços precisam ser constantemente revistos.

Para a criação de propostas pedagógicas inovadoras, [Roque Herrera et al. 2020] sugerem que os cursos de graduação têm sido incentivados, durante suas reestruturações, a incluírem metodologias de ensino que alcancem os novos perfis de alunos e professores. [Marin et al. 2010] afirmam que para os estudantes, pelo fato de partirem de situações reais ou de fazerem paralelos com a realidade, esses métodos possibilitam que o aluno mantenha uma continuidade no estudo, uma maior autonomia e uma responsabilidade com as atividades propostas.

É comum que em grande parte do tempo o ensino seja desenvolvido utilizando materiais e comunicações escritas, orais e audiovisuais, previamente selecionados ou elaborados [Hosseini-Mohand et al. 2021]. No entanto a mescla dessa forma de ensinar com atividades, desafios e informações contextualizadas permite aos alunos uma melhor aprendizagem. Como exemplo [Moran 2015] afirma que para aprender a dirigir um carro, não basta ler muito sobre esse tema; tem que experimentar, rodar com ele em diversas situações com supervisão, para depois poder assumir o comando do veículo sem riscos. Ele afirma ainda que as metodologias precisam acompanhar os objetivos pretendidos, pois se existe a necessidade de que os alunos sejam proativos, ou seja, construtores do próprio conhecimento, é necessária a adoção de metodologias nas quais os alunos se envolvam em atividades cada vez mais complexas, que permitam que eles tomem decisões e avaliem seus próprios resultados.

Entre as várias metodologias disponíveis na literatura que buscam alcançar essa independência do aluno de forma ativa, pode-se destacar a utilização da gamificação e da

sala de aula invertida. Ambas as metodologias consistem em possibilitar ao aluno uma maneira diferente da tradicional de construir a aprendizagem.

## **2.1. Gamificação**

A gamificação é a aplicação de elementos popularmente presentes em jogos, como uma forma eficiente para motivar a iniciativa, resolução de problemas e o aumento da aprendizagem dos alunos em várias áreas do conhecimento [Fardo 2013]. A gamificação é uma ferramenta de motivação para os estudantes resolverem problemas com soluções mais eficientes [Hossein-Mohand et al. 2021].

A aplicação da gamificação pode ocorrer tanto dentro de um ambiente virtual como em um ambiente real. É importante entender que não se trata de um jogo desenvolvido especificamente para o ambiente educacional, e sim, de uma abstração de elementos relevantes que venham a aumentar a motivação e a participação dos alunos dentro da sala de aula. Um dos vários objetivos da gamificação dentro da educação é reinventar a maneira de como os estímulos são lançados ao aluno para aumentar seu engajamento. No entanto esta combinação deve ser feita de forma a obter a maior efetividade possível.

## **2.2. Sala de aula invertida**

Para [Sailer and Sailer 2021] a sala de aula invertida surge com a proposta de dinamizar a transmissão de informações dentro de sala de aula. Este conceito pode ser aplicado através da utilização de ferramentas online [Luo et al. 2019], pois o objetivo da sala de aula invertida é promover a iniciativa dos alunos em estudar um pouco de determinado conteúdo antes do professor ministrar a aula.

Segundo o relatório elaborado por [Maya et al. 2013] devem existir algumas regras para que se possa inverter uma sala de aula. As regras consistem em uma quantidade significativa de questionamento, resolução de problemas e de outras atividades de aprendizagem ativa; feedback imediato após a realização das atividades presenciais; incentivo a participação nas atividades; e uso de materiais altamente estruturados e bem planejados.

## **3. Ensino de Algoritmos**

Para facilitar a utilização dessas metodologias ativas no ensino de ciências exatas e, mais especificamente, no ensino de algoritmos, se faz importante a construção adequada de planos de ensino que prevejam a correta utilização dessas metodologias, bem como as ferramentas adequadas para isso.

A disciplina de Algoritmos é, para [Rapkiewicz 2006], a base para o ensino de programação na maioria dos cursos na área de Informática. Abordando princípios de lógica de programação, buscando desenvolver a capacidade de análise e resolução de problemas na forma de algoritmos. Segundo [Doukakis et al. 2021], essa é uma disciplina com alto índice de evasão e reprovação, sendo um dos gargalos existentes nos cursos de graduação, dificultando e, as vezes, impedindo a continuidade dos alunos no curso.

Segundo a estrutura curricular constante nas Diretrizes Curriculares do MEC [Azerredo 2000], a matéria de Programação faz parte da área de formação básica em Ciência da Computação, juntamente com as matérias ‘Computação e Algoritmos’ e ‘Arquitetura de Computadores’. Seu conteúdo abrange, além do ensino de linguagens de

programação propriamente ditas, os conceitos, os princípios e os modelos de programação e o estudo de estruturas de dados e de métodos de classificação e pesquisa de dados. Conforme consta na caracterização da matéria de programação, entendida como programação de computadores, ela é uma atividade voltada à solução de problemas. Nesse sentido, ela está relacionada com uma variada gama de outras atividades como especificação, projeto, validação, modelagem e estruturação de programas e dados, utilizando as linguagens de programação como ferramentas.

Com o intuito de adequar da melhor maneira a disciplina, [Setubal 2000] apresenta algumas recomendações genéricas a serem aplicadas no ensino de disciplinas relacionadas a algoritmos: coerência com os objetivos fundamentais, ênfase no pensamento crítico e a teoria na prática.

### 3.1. Maratona de Programação como elemento didático

A Maratona de Programação é um evento da Sociedade Brasileira de Computação que existe desde o ano de 1996. A Maratona nasceu das competições regionais classificatórias para as finais mundiais do concurso de programação, o International Collegiate Programming Contest, e é parte da regional sulamericana do concurso [IME-USP and SBC 2019].

Os participantes de cada um dos times devem avaliar problemas, projetar testes, e implementar soluções que sejam aprovadas pelos juízes da competição. Alguns desses problemas requerem apenas interpretação e conhecimento básico em programação, para outros são necessários conhecimentos de técnicas de programação mais avançadas e até mesmo de conhecimento matemático aprofundado. Uma questão da maratona consiste em um nome, um enunciado, uma explicação da entrada de dados, uma explicação da saída de dados esperada, exemplos da entrada de dados e respectivos exemplos da saída de dados. Ou seja, é um problema passível de solução computacional, que pode ser resolvido através de um algoritmo.

Ao se utilizar questões da maratona inseridas no contexto de sala de aula permite-se aos alunos uma busca essencial pela resposta do problema. Ao passo que eles realizam essa busca, eles podem também adquirir conhecimento sobre assuntos ainda não trabalhados em sala de aula e ao mesmo tempo reforçar o que já foi trabalhando anteriormente. As questões da maratona podem ser aplicadas de forma avulsa ou em forma de competição, simulando uma maratona real. Neste trabalho propõe-se três formas diferentes de utilizar a maratona em uma disciplina de Algoritmos:

**a) Questão Avulsa:** uma questão avulsa pode ser elencada para a turma. Antes ou após determinado assunto. Se a questão elencada for de baixa dificuldade, ela pode ser usada como exercício durante a aula. Caso tenha uma complexidade maior, pode ser utilizada como atividade para a sala de aula invertida.

**b) Competição curta:** nessa modalidade, divide-se a turma em equipes de três alunos (tal como acontece na maratona). São apresentadas três questões de baixa a média complexidade (no máximo uma de média complexidade). Os alunos têm o período da aula (em média 100 minutos) para resolver as três questões. Vence a equipe que responder mais questões em menos tempo.

**c) Competição longa:** nessa modalidade, a turma também é separada em equipes de três alunos (como na maratona). São apresentadas dez questões de baixa, média e alta

complexidade (no máximo uma de alta complexidade e pelo menos quatro de média complexidade). Os alunos têm o período de uma semana, antes ou depois da aula (aplicando-se a sala de aula invertida), para resolver as dez questões. Vence a equipe que responder mais questões em menos tempo.

Na primeira situação, será normal os alunos apresentarem dificuldades no início. Pode ser útil usar essa forma para ambientar os alunos aos conceitos de resolução de questões de maratona. Nas demais formas, e principalmente na última, os alunos serão levados a realizar pesquisas adicionais às aulas ministradas pelo professor. Somente as aulas não devem ser suficientes para que os alunos resolvam as questões de média e alta complexidade, o que os levará a um processo de aprendizagem ativa, pois para atingir o objetivo da resposta eles deverão pesquisar, testar, validar e executar.

Para as atividades de maratona é importante a utilização de uma ferramenta, de preferência online, para a realizar a competição e avaliação automática das questões submetidas pelos alunos. Uma das principais plataformas de treinamento é a Beecrowd.

### **3.2. Plataforma Beecrowd**

Segundo [Tonin and Bez 2012], o principal problema nos algoritmos e no ensino de programação para novos estudantes é atender com eficiência a grande diversidade de alunos e seus diferentes modos e ritmos de aprendizado. Sendo assim, a possibilidade de utilização de ferramentas online com grande disponibilidade para aprendizagem ativa pode ser uma forma de possibilitar que cada aluno aprenda em seu próprio ritmo e velocidade. No método tradicional de ensino de algoritmos, o professor deve analisar lógica, sinteticamente e lexicalmente uma solução desenvolvida por um estudante de graduação. Este é um processo demorado e pode ter falhas, pois alguns erros ou detalhes podem passar despercebidos [Tonin and Bez 2012].

Além do Beecrowd outras ferramentas possibilitam resolução de problemas e competição entre os alunos, ao mesmo tempo que auxiliam o professor com a correção automatizada e em tempo real. Duas dessas ferramentas são: O "BOCA" que é um juiz online para competições. Ele é utilizado principalmente na competição brasileira da Maratona de Programação. Outro juiz on-line é o UVa Online Judge da Universidade de Valladolid, na Espanha, que contém uma enorme variedade de problemas em todos os assuntos de algoritmos.

No entanto, o BOCA por ser uma plataforma que precisa de constante manutenção do professor na atualização de questões, competições e correções, dificulta o uso na proposta de metodologias ativas. O mesmo acontece com o UVa, mas, principalmente, pela ausência de uma versão em português da plataforma e da dificuldade da criação de novas questões pelo professor. Sendo assim, visando atender a diversidade de alunos e suas diferentes formas e ritmo de aprendizagem, foi escolhido o Beecrowd, buscando principalmente a interatividade, flexibilidade, novas fontes de informação e usabilidade.

## **4. Classificação de Atividades do Beecrowd**

Na plataforma Beecrowd os problemas são separados em categorias e em dificuldades. As categorias são divididas em: Iniciante, AD-HOC, Strings, Estruturas e Bibliotecas, Matemática, Paradigmas, Grafos, Geometria Computacional e SQL. As dificuldades são identificadas pelos níveis de 1 a 10, sendo 1 as questões mais fáceis e 10 as mais difíceis.

Além disso, é possível identificar em cada questão a quantidade de vezes que ela foi resolvida.

A pontuação de cada problema é definida, de acordo com [Tonin and Bez 2012], pelo nível de dificuldade. Esse por sua vez, é estimado com base em quantos usuários resolveram o problema para determinar sua dificuldade. Nesse sistema, problemas que são resolvidos por muitos usuários com poucas tentativas recebem uma dificuldade menor, enquanto problemas que foram mais tentados e pouco resolvidos são classificados em um nível de dificuldade maior. A dificuldade dos problemas é reajustada semanalmente. Isto significa que problemas podem mudar de nível e ter sua pontuação alterada. Desta forma a sua pontuação total poderá apresentar uma variação.

Tendo por base o entendimento de como os níveis das questões variam e suas categorias e entendendo que a disciplina de Algoritmos somente é ofertada a alunos que estão aprendendo a programar e muita das vezes tendo seu primeiro contato com a programação, foi escolhida apenas a categoria “Iniciantes” para ser utilizada nesta proposta.

No modelo proposto a categoria Iniciantes passa por uma subdivisão, buscando se adequar aos conteúdos da disciplina. Para isso, à cada assunto foram selecionadas dez questões da categoria iniciantes. Os níveis foram reorganizados em Baixo, Médio e Alto da seguinte maneira: questões de nível Baixo correspondem as questões de nível 1 até 3 no Beecrowd; questões de nível Médio correspondem a questões de nível 4 a 5 no Beecrowd; e questões de nível Alto correspondem a questões de nível 6 a 10 no Beecrowd.

A distribuição das dificuldades em cada assunto ficou na proporção de 10% para questões de nível Alto, 40% para questões de nível Médio e 50% para questões de nível Fácil. Essa distribuição foi pensada para atender a maior demanda possível, que nesse caso seria a Competição Longa, que utiliza até dez questões. Então para cada assunto disponível no plano de ensino, existirão dez questões disponíveis para utilizar as metodologias ativas.

Como base para a distribuição das questões nos conteúdos da disciplina, a mesma foi organizada em quatro módulos. O primeiro consiste em um módulo introdutório, onde são explanados conceitos de fluxogramas; pseudolinguagem; linguagem de programação; declaração e inicialização de variáveis; entrada e saída; operadores aritméticos e de atribuição; operadores relacionais e lógicos. O segundo módulo são expressões e estruturas de controle condicionais (comando se e comando escolha); o terceiro módulo traz estruturas de controle de laços (comando enquanto e comando para); já o quarto módulo apresenta estruturas de dados básicas (vetores e matrizes) e funções.

É importante que o professor que aplique essa metodologia tenha em mente que este trabalho propõe um modelo, que pode ser seguido ou adaptado a partir dessas regras de distribuição. As questões apresentadas a seguir são meramente sugestões de aplicação. Para o módulo de Introdução as questões sugeridas estão na Tabela 1, do lado esquerdo, são questões que podem ser feitas utilizando apenas os conceitos básicos de aritmética, leitura e escrita. Para o módulo de Condicionais as questões sugeridas estão também na Tabela 1, porém do lado direito, estas questões podem ser resolvidas utilizando apenas condicionais e as operações básicas que constam no módulo de introdução.

Para o módulo de Laços/Repetições as questões sugeridas estão na Tabela 2, do

**Table 1. Problemas selecionados para os Módulos 1 e 2**

Módulos do 1º Trimestre			
1 - Introdução		2 - Condicionais	
Problema	Título	Problema	Título
1001	Extremamente Básico	1043	Triângulo
1005	Média 1	1046	Tempo de Jogo
1004	Produto Simples	1050	DDD
1008	Salário	1061	Tempo de um Evento
1010	Cálculo Simples	1041	Coordenadas de um Ponto
1012	Área	1049	Animal
1013	O Maior	1078	Tabuada
1038	Lanche	1114	Senha Fixa
1021	Notas e Moedas	2057	Fuso Horário
1037	Intervalo	2059	Ímpar, Par ou Roubo

lado esquerdo, todas as questões devem ser feitas utilizando algum tipo de estrutura de controle de laço ou outra forma de repetição. Os conteúdos apresentados nos módulos 1 e 2 são fundamentais para o desenvolvimento desse módulo. Para o módulo de Vetores, Matrizes e Funções as questões sugeridas estão na Tabela 2, do lado direito, e são diretamente relacionadas aos conceitos de vetores, matrizes e utilização de funções.

**Table 2. Problemas selecionados para os Módulos 3 e 4**

Módulos do 2º Trimestre			
3 - Laços/Repetições		4 - Vetores, Matrizes e Funções	
Problema	Título	Problema	Título
1059	Números Pares	1435	Matriz Quadrada I
1071	Soma de Impares Consecutivos I	1541	Contruindo Casas
1078	Tabuada	1759	Ho Ho Ho
1080	Maior e Posição	1858	A Resposta de Theon
1094	Experiências	1957	Converter para Hexadecimal
1098	Sequencia IJ 4	1963	O Filme
1117	Validação de Nota	2003	Domingo de Manhã
1144	Sequência Lógica	2163	O Despertar da Força
1151	Fibonacci Fácil	2167	Falha do Motor
1153	Fatorial Simples	2221	Batalha de Pomekons

## 5. Resultados e Discussão

Com a organização das questões por conteúdo, o plano de ensino passa a viabilizar a metodologia ativa da maratona de programação diretamente nos conteúdos. Esse modelo de plano de ensino busca ser encaixado em qualquer carga horária divisível por quatro. Ou seja, independente da carga horária ser de 40, 60 ou 80 é possível realizar a aplicação do plano.

Com o plano concebido, foram utilizados seis semestres da disciplina de Algoritmos para avaliar a proposta, onde três semestres utilizaram as metodologias ativas e três semestres não utilizaram, tudo de forma intercalada.

### 5.1. Proposta de Plano de Ensino

A Tabela 3 mostra a distribuição do conteúdo durante o período letivo e quando a metodologia ativa deve ser utilizada, além da forma que a mesma pode ser aplicada. Seguindo o modelo, o professor será capaz de utilizar as metodologias ativas no decorrer da disciplina. Isso possibilitará ao aluno estudar de forma antecipada os conteúdos que serão apresentados na aula seguinte, ao mesmo tempo que revisa os conteúdos da aula anterior. Isso tudo sendo realizado pelo próprio aluno, de forma ativa e lúdica, na figura de um desafio.

**Table 3. Conteúdos e aplicação da metodologia ativa**

Carga Horária (100%)	CONTEÚDO	METODOLOGIA ATIVA
<b>MÓDULO I</b>		
5%	Fluxogramas e Pseudolinguagem;	Sala de Aula Invertida: Questão Avulsa
5%	Linguagem de Programação;	Sala de Aula Invertida: Questão Avulsa
5%	Declaração e Inicialização de Variáveis;	Sala de Aula Invertida: Questão Avulsa
5%	Operadores Aritméticos e de Atribuição;	Gameificação: Competição Curta
5%	Operadores Relacionais e Lógicos;	Gameificação: Competição Curta
<b>MÓDULO II</b>		
5%	O Comando SE;	Sala de Aula Invertida: Questão Avulsa
5%		Gameificação: Competição Curta
5%		Sala de Aula Invertida: Questão Avulsa
5%	O Comando ESCOLHA;	Gameificação: Competição Longa
5%		
<b>MÓDULO III</b>		
5%	O Comando PARA;	Sala de Aula Invertida: Questão Avulsa
5%		Gameificação: Competição Curta
5%		Gameificação: Competição Questão Avulsa
5%	O Comando ENQUANTO;	Gameificação: Competição Longa
5%		
<b>MÓDULO IV</b>		
5%	Vetores;	Sala de Aula Invertida: Questão Avulsa
5%		Gameificação: Competição Curta
5%	Matrizes;	Sala de Aula Invertida: Questão Avulsa
5%		Gameificação: Competição Longa
5%	Funções;	

## 5.2. Aplicação prática na disciplina

Durante seis períodos intercalados (de 2018-2 até 2021-1) essa metodologia foi aplicada em uma disciplina de Algoritmos, para alunos que em sua maioria não possuíam conhecimento de programação.

Para efeitos comparativos, iniciou-se avaliando um período sem a aplicação da metodologia, período identificado como 2018-2. No período seguinte, identificado como 2019-1, foi implantada a metodologia. Em sequência com a devida frequência, foram avaliados os períodos seguintes: 2019-2 e 2020-2 sem metodologias ativas e 2020-1 e 2021-1 com metodologias ativas. Adicionalmente, devido a pandemia de COVID-19 os períodos de 2020-1 até 2021-1 foram ministrados de maneira remota.

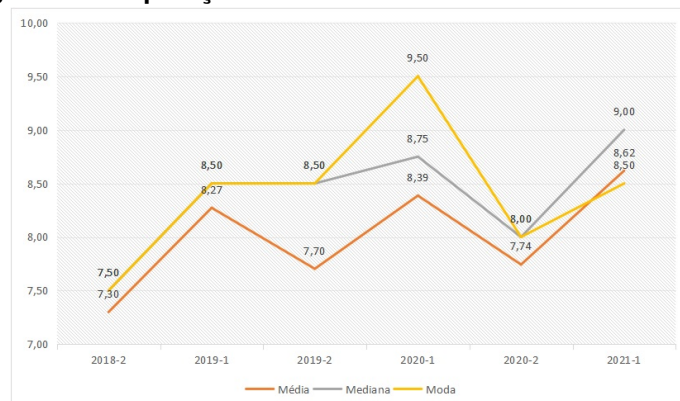
A Figura 1 apresenta o gráfico resultante dos dados da média, moda e mediana das notas dos alunos durante os seis semestres. Para efeito de análise objetiva e de correção dos dados, foram desconsiderados dos cálculos os alunos reprovados por falta. Esses alunos não podem ser incluídos, pois os mesmos não vivenciaram as disciplinas em todo o contexto e isso criaria um resultado enviesado de acordo com a quantidade de alunos reprovados dessa maneira em cada semestre.

Podemos avaliar com esses dados que ao aplicar as metodologias de gamificação e sala de aula invertida, o aproveitamento da turma tende a obter crescimentos. Inclusive fazendo que as notas mais altas se repitam com mais constância. Nos períodos sem as metodologias o que ocorre é uma minoração das notas, ou pode-se dizer que é a ocorrência normal das notas.

A partir do período 2020-1 com a pandemia de COVID-19 tivemos uma migração para o sistema remoto de ensino. Ainda assim, de acordo com os dados, o uso das metodologias ativas proporcionou um ganho substancial à não utilização. Ao mesmo tempo podemos observar que o período pandêmico somado ao uso dessas metodologias potencializou os objetivos da gamificação e da sala de aula invertida. Uma vez que ambos os processos se dão prioritariamente em ambiente virtual proporcionando um maior



**Figure 1. Comparação entre notas dos alunos nos semestres**



engajamento dos alunos.

Esse engajamento não ocorreu com a mesma intensidade no período 2020-2, no qual não foram aplicadas as metodologias ativas. Nesse período as aulas foram ministradas de maneira remota, mas de forma tradicional, expositiva e dialogada. Foi possível perceber uma dificuldade maior dos alunos em relação aos outros períodos.

## 6. Conclusão

A gamificação como metodologia ativa é viável para a implantação em disciplinas de algoritmos. A utilização delas através do conceito da maratona de programação permite ao professor criar uma imersão dos alunos em um contexto totalmente ambientado em programação de computadores.

O plano de ensino ao abordar quando e como utilizar as metodologias ativas fornece ao professor apoio para desenvolver a disciplina. O que implica na melhoria do processo de ensino-aprendizagem e consequentemente poderá impactar em uma redução da evasão dos alunos nos cursos superiores em informática.

Perspectivas futuras na aplicação do plano de ensino em outras instituições são importantes para verificação da eficácia global. Além disso, podem existir diferenças de perfis entre alunos, turmas e instituições de ensino superior, o que pode resultar em adaptações das propostas metodológicas de acordo com as características desejadas. Além disso, será interessante avaliar, no período de colação de grau dessas turmas, o índice de evasão.

### 6.1. Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001

## References

Azerredo, P. A. (2000). Uma proposta de plano pedagógico para a matéria de programação. In *Anais do II Curso: Qualidade de Cursos de Graduação da Área de Computação e Informática (WEI 2000)*. Editora Universitária Champagnat, Curitiba.

- Calderon Ribeiro, M. I. and Passos, O. M. (2020). A study on the active methodologies applied to teaching and learning process in the computing area. *IEEE Access*, 8:219083–219097.
- Doukakis, S., Psaltidou, A., Stavradi, A., Adamopoulos, N., Tsiotakis, P., and Stergou, S. (2021). Measuring the technological pedagogical content knowledge (TPACK) of in-service teachers of computer science who teach algorithms and programming in upper secondary education. *CoRR*, abs/2105.09252.
- Fardo, M. L. (2013). *A gamificação aplicada em ambientes de aprendizagem*. RS: Universidade de Caxias do Sul.
- Hossein-Mohand, H., Trujillo-Torres, J.-M., Gómez-García, M., Hossein-Mohand, H., and Campos-Soto, A. (2021). Analysis of the use and integration of the flipped learning model, project-based learning, and gamification methodologies by secondary school mathematics teachers. *Sustainability*, 13(5).
- IME-USP and SBC (2019). Maratona de programação - o que é? *IME-USP*.
- Lindemann, V., Tarouco, L., and Bercht, M. (2008). Estilos de aprendizagem: um estudo de casos em turmas de algoritmos e programação. *SBIE. Fortaleza*.
- Luo, H., Yang, T., Xue, J., and Zuo, M. (2019). Impact of student agency on learning performance and learning experience in a flipped classroom. *British Journal of Educational Technology*, 50(2):819–831.
- Marin, M. J. S., Lima, E. F. G., Paviotti, A. B., Matsuyama, D. T., Silva, L. K. D., Gonzalez, C., Druzian, S., and Ilias, M. (2010). Aspectos das fortalezas e fragilidades no uso das metodologias ativas de aprendizagem. *Revista Brasileira de Educação Médica*, v. 34:13–20.
- Maya, A., Boneh, D., Fisher, D., and Klemer, S. (2013). Portal flipped classroom field guide.
- Moran, J. (2015). Mudando a educação com metodologias ativas. In *Coleção Mídias Contemporâneas. Convergências Midiáticas, Educação e Cidadania: aproximações jovens. v. II*. Foca Foto-PROEX/UEPG, PG.
- Raabe, A. L. A. and Silva, J. M. C. (2005). Um ambiente para atendimento às dificuldades de aprendizagem de algoritmos. In *XIII Workshop sobre educação em Computação, 2005*. São Leopoldo, RS.
- Rapkiewicz, E. C. (2006). Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. In *CINTED-UFRGS: Novas Tecnologias na Educação*, volume v. 4, n. 2, page 02–11. Porto Alegre.
- Romero-García, C., Buzón-García, O., and de Paz-Lugo, P. (2020). Improving future teachers' digital competence using active methodologies. *Sustainability*, 12(18).
- Roque Herrera, Y., Zalagaz Sánchez, M. L., Valdivia-Moral, P., Marín-Marín, J.-A., and Alonso García, S. (2020). Active methodologies in the training of future health professionals: Academic goals and autonomous learning strategies. *Sustainability*, 12(4).
- Sailer, M. and Sailer, M. (2021). Gamification of in-class activities in flipped classroom lectures. *British Journal of Educational Technology*, 52(1):75–90.

Setubal, J. C. (2000). Uma proposta de plano pedagógico para a matéria de computação e algoritmos. In *Anais do II Curso: Qualidade de Cursos de Graduação da Área de Computação e Informática (WEI 2000)*. Editora Universitária Champagnat. Curitiba.

Tonin, N. and Bez, J. L. (2012). Online judge: a new classroom tool for interactive learning. *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS)*, (ens):1–5.