

Explorando uma metodologia híbrida: Relato de experiência em programação introdutória para cursos de ciência da computação e sistemas de informação

Marluce Rodrigues Pereira¹, Paula C. F. Cardoso²

Juliana Galvani Greghi¹, Joaquim Quinteiro Uchoa¹, Renato Ramos da Silva¹

¹Departamento de Computação Aplicada/Instituto de Ciências Exatas e Tecnológicas
Universidade Federal de Lavras, Lavras, Brasil

²Faculdade de Computação/Instituto de Ciências Exatas e Naturais
Universidade Federal do Pará (UFPA) - Belém, PA - Brasil

{marluce,juliana,joukim,renato.ramos}@ufla.br pcardoso@ufpa.br

Resumo. O processo de aprendizagem de algoritmos e programação costuma ser um desafio para muitos alunos dos cursos de graduação em Ciência da Computação e Sistemas de Informação. Com o objetivo de aumentar os índices de aprendizagem, o ensino tradicional está dando lugar a novas metodologias, conhecidas como metodologias ativas, como aprendizagem baseada em problemas, sala de aula invertida e gamificação. Este artigo apresenta uma experiência com uma metodologia que integra todas essas metodologias ativas para o ensino de programação introdutória para alunos calouros de Ciência da Computação e Sistemas de Informação. Os resultados indicam uma melhora no aprendizado, conforme evidenciado por um número maior de alunos aprovados em comparação com aqueles ensinados com métodos tradicionais.

1. Introdução

No contexto e na estrutura de um currículo de computação, especialmente em cursos STEM (Ciência, Tecnologia, Engenharia e Matemática), é necessário incluir um curso introdutório de programação. Esse curso, adaptado para alunos iniciantes, abrange habilidades de resolução de problemas, conceitos fundamentais de programação, a sintaxe e a semântica de uma linguagem de programação e sua aplicação prática no desenvolvimento de soluções. Apesar do progresso nos métodos de ensino e nas ferramentas educacionais para a programação introdutória, as taxas de desistência e reprovação continuam a afetar esses cursos. Ainda há uma falta de consenso em relação aos principais desafios, bem como uma categorização clara e exaustiva desses obstáculos (Medeiros, Ramalho, & Uchoa, 2018)

De acordo com Barcelos, Tarouco e Bercht (2009), os alunos muitas vezes têm dificuldade para organizar seu raciocínio, desenvolver estratégias de resolução de problemas, manter a atenção, concentrar-se e estimular os processos de cálculo mental. Consequentemente, as habilidades associadas ao raciocínio lógico - como experimentar, observar, levantar hipóteses e deduzir - desempenham um papel crucial no aprendizado nessa área de conhecimento e podem afetar significativamente a eficácia do processo de aprendizado.

Morais, Neto e Góes (2020) realizaram uma revisão sistemática da literatura e apresentaram as dificuldades apontadas por trabalhos de diversos países. Os autores apontaram que há uma deficiência na competência dos alunos, principalmente na resolução de problemas, no raciocínio lógico e na matemática básica. Os alunos geralmente entendem as soluções apresentadas

pelo professor, mas têm dificuldade para reproduzi-los de forma independente. É necessário um tempo de estudo flexível para resolver exercícios, consolidar o conhecimento e manter a motivação para buscar soluções. Além disso, turmas grandes impedem um monitoramento mais individualizado e a implementação de tarefas mais interessantes. Também foi observado que as possíveis causas das dificuldades dos alunos podem surgir da natureza da programação, da formação e do histórico dos alunos, das atitudes em relação ao estudo e das estratégias pedagógicas comumente usadas nas disciplinas de algoritmos e programação. Essas dificuldades podem causar reprovações nas disciplinas de algoritmos e programação até a desistência do curso.

Santos, Gorgônio, Lucena e Gorgônio (2015) discutem que a falta de interesse dos alunos pelo conteúdo pode ser mais acentuada quando eles estão repetindo um curso e revisando conceitos abordados em um semestre anterior. Para resolver esse problema, a inclusão de atividades práticas complementares no final de algumas aulas - como o uso de recursos visuais e de programação em plataforma, como o Arduino - pode ser fundamental para aprimorar o processo de ensino e aprendizagem, aumentando a motivação dos alunos.

Nesse contexto, a abordagem tradicional do ensino de algoritmos contrasta com os métodos de aprendizagem ativa. As abordagens tradicionais de ensino de algoritmos geralmente se baseiam em aulas, leituras de livros didáticos e avaliação por meio de provas escritas, o que pode levar a uma aprendizagem passiva, em que os alunos recebem informações sem se envolver ativamente com elas. A Aprendizagem Ativa (AL), por sua vez, é uma abordagem educacional na qual os alunos estão envolvidos em um ambiente controlado, com preparação prévia, podendo pensar e refletir ativamente durante todo o processo. Três abordagens comuns de AL são a aprendizagem baseada em problemas (PBL), a sala de aula invertida (FC) e a gamificação (GAM) (Pirker, Riffnaller-Schiefer, & Gütl, 2014). Essas abordagens ajudam a desenvolver o pensamento crítico e as habilidades de resolução de problemas, tornando a experiência de aprendizagem mais dinâmica e eficaz. Ao envolver ativamente os alunos em sua educação, esses métodos visam aumentar a motivação, melhorar a compreensão e reduzir as taxas de evasão, abordando muitos dos desafios identificados no ensino tradicional de algoritmos (Grotta & Prado, 2018; de Moraes, da Costa, & Scholz, 2022; de Oliveira, da Silva, & Rodrigues, 2022).

Assim, o objetivo deste artigo é descrever um relato de experiência de uma metodologia de ensino híbrida que combina várias estratégias de aprendizagem ativa no curso introdutório de algoritmos para cursos de graduação em Ciência da Computação e Sistemas de Informação. Nossa abordagem combina PBL, FC e gamificação para o curso introdutório de programação usando a linguagem C++. Este artigo está organizado da seguinte forma: A Seção 2 apresenta uma base teórica da metodologia de ensino. A Seção 3 explica a metodologia proposta. A Seção 4 apresenta os resultados iniciais. Por fim, a Seção 5 apresenta as observações finais e estabelece trabalhos futuros.

2. Fundamentos teóricos

2.1. Ensino tradicional e aprendizagem combinada

Altos índices de evasão, reprovação e desistência têm afetado o ensino de programação de computadores nos cursos de graduação em Ciência da Computação e Sistemas de Informação, tanto no Brasil quanto no exterior (Zanetti & Oliveira, 2015; de Moraes et al., 2022). Há vários fatores que tornam o aprendizado de programação de computadores um desafio, e as abordagens tradicionais de ensino não têm sido capazes de lidar com isso de forma eficaz. O ensino tradicional de programação de computadores geralmente segue um formato estruturado e linear, que inclui as seguintes características principais: aulas predominantemente baseadas em palestras conduzidas pelo instrutor;

sessões de laboratório para oferecer aos alunos oportunidades de prática de programação; uso de um livro-texto padrão que os alunos devem seguir; e avaliação por meio de exames escritos (Grotta & Prado, 2018).

No entanto, especialmente durante o isolamento social devido à pandemia da COVID-19, as instituições de ensino tiveram que explorar outras abordagens e recursos de ensino, e todas as atividades tiveram que ser realizadas remotamente, conforme observado nos trabalhos (Silveira, Bertolini, Parreira, da Cunha, & Bigolin, 2021; Dias, 2022). Os autores apontaram que o cenário trouxe muitos desafios práticos e técnicos para ser implementado de forma inclusiva. Do ponto de vista educacional, havia a necessidade de produção rápida de materiais didáticos e ferramentas de monitoramento das atividades educacionais, incorporando a Tecnologia da Informação e Comunicação (TIC). A aprendizagem remota e combinada tornou-se mais comum, com maior uso de ferramentas on-line, plataformas de aprendizagem e recursos digitais. Além disso, houve uma maior ênfase na flexibilidade e acessibilidade do ensino, bem como na adaptação de metodologias para melhor atender às necessidades dos alunos em um ambiente virtual.

Com base nas lições aprendidas durante a pandemia, a aprendizagem remota e combinada se tornou mais comum, com maior uso de ferramentas on-line, plataformas de aprendizagem e recursos digitais (de Oliveira et al., 2022). Além disso, há uma maior ênfase flexibilidade e acessibilidade do ensino, bem como na adaptação de metodologias para melhor atender às necessidades dos alunos em um ambiente virtual. Preuss e de Lima (2023) descrevem uma lista de ferramentas que podem ajudar a superar as adversidades enfrentadas em cursos introdutórios de programação no contexto do ensino remoto. Isso aumenta o debate no campo da educação sobre as oportunidades e os desafios da AL em cursos introdutórios de programação. Diferentemente abordagem tradicional, a EaD visa à participação efetiva dos alunos e seu envolvimento por meio de várias metodologias, estratégias, abordagens e técnicas pedagógicas centradas no aluno.

2.2. Aprendizagem baseada em problemas

A aprendizagem baseada em problemas (PBL) é uma metodologia de instrução centrada nos alunos, em que eles colaboram em pequenos grupos para resolver problemas da vida real, introduzindo assim novos materiais de aprendizagem. O processo de aprendizagem é predominantemente autodirigido: por meio da solução de problemas, da resposta a perguntas, da discussão e da análise, os alunos desvendam a teoria por meio de leitura e estudo independentes. A avaliação inclui autoavaliação, avaliação dos colegas e do professor, promovendo a responsabilidade do aluno (Caceffo, Gama, & Azevedo, 2018; Goletti, Mens, & Hermans, 2021).

Yew e Goh (2016) fornecem evidências de que o desempenho superior dos alunos que estão aprendendo em condições de PBL, em oposição às condições de aulas expositivas, tem sido amplamente adotado em diversos campos e contextos educacionais para promover o pensamento crítico e a resolução de problemas em situações autênticas de aprendizagem.

2.3. Sala de aula invertida

No modelo de sala de aula invertida, o professor auxilia os alunos em vez de apenas ministrar a formação, enquanto os alunos assumem a responsabilidade por seu próprio processo de aprendizagem e gerenciam seu próprio ritmo de estudo. Uma parte significativa do processo de aprendizado do aluno é aprimorada pelas tecnologias, que são realizadas antes do horário da aula. Consequentemente, o professor agora pode se envolver com os alunos por meio de várias atividades de aprendizagem, como discussões, resolução de problemas propostos pelos alunos, atividades práticas e orientação (Ayır, 2021).

& Ağayır, 2018; Onyema et al., 2021). Ağayır e Ağayır (2018) também afirmaram que, em geral, o modelo invertido na educação produz resultados acadêmicos positivos. A maioria dos estudos revisados relatou que o modelo invertido promove melhorias no desempenho da aprendizagem dos alunos. Entretanto, não há evidências suficientes para garantir a generalização.

Para cursos de tecnologia, o uso do modelo de sala de aula invertida demonstrou aumentar o desempenho, a autoeficácia e a interação dos alunos. Ele promove a participação ativa na troca de ideias e informações, o que pode potencialmente aumentar seu interesse e promover o pensamento crítico e uma compreensão mais profunda dos materiais de aprendizagem. (Al-Samarraie, Shamsuddin, & Alzahrani, 2020).

2.4. Gamificação

A literatura resume "Gamificação" como "o uso de elementos de design característicos de jogos em contextos que não são de jogos". Dicheva, Dichev, Agre e Angelova (2015), por sua vez, afirmam que a penetração da tendência de gamificação em ambientes educacionais parece ainda estar subindo ao topo. Mas a maioria dos trabalhos atuais descreve apenas alguns mecanismos e dinâmicas de jogos e reitera seu possível uso no contexto educacional, enquanto a verdadeira pesquisa empírica sobre a eficácia da incorporação de elementos de jogos em ambientes de aprendizagem ainda é escassa.

3. Nossa abordagem

A ideia principal por trás de nossa abordagem foi combinar todos os aspectos positivos da AL em uma metodologia híbrida para o curso de Introdução a Algoritmos em Ciência da Computação e Sistemas de Informação, implementado nos dois semestres de 2023 na Universidade Federal de Lavras. Entretanto, antes de nos aprofundarmos na metodologia e nos materiais desenvolvidos pelos professores, são necessárias algumas informações importantes sobre esse curso. Na instituição, o curso compreende duas horas de instrução remota e quatro horas de sessões práticas em sala de aula, divididas em duas sessões de igual duração. Todas as sessões práticas em sala de aula ocorreram às terças e quintas-feiras. Esse arranjo, acordado com todos os coordenadores de curso, onde esse curso é obrigatório, manteve todas as aulas no mesmo horário e evitou interrupções devido a feriados. O semestre tem duração de dezessete semanas.

Ao final do curso, espera-se que o aluno tenha adquirido as seguintes habilidades representar vários problemas cotidianos por meio da lógica de programação; dominar os conceitos fundamentais da lógica de programação e das linguagens de programação para a construção de algoritmos simples; reconhecer e dominar os elementos básicos para o desenvolvimento de algoritmos (operações lógico-matemáticas e atribuição; estruturas sequenciais, condicionais e repetitivas; modularização e recursão; matrizes, strings, matrizes e registros); e lidar adequadamente com arquivos de texto e binários, ponteiros e alocação dinâmica de memória. A linguagem de programação usada no curso é C++.

Até o segundo semestre de 2022, esse curso utilizava uma metodologia ensino tradicional com duração de 17 semanas, com 4 horas semanais de aulas teóricas em sala de aula e 2 horas de aulas em laboratório. Durante a pandemia da COVID-19, as aulas foram ministradas on-line com materiais disponíveis em um Learning Management System (LMS) e exercícios em uma Online Judge Tool (OJT). Como informação adicional, nossa Instituição tem três tipos de classificação de um aluno de graduação que é reprovado: i) quando o aluno faz uma simulação e não consegue expressá-la nos exames; ii) por não assistir a uma aula mínima e iii) por desistir da aula no respectivo semestre ou curso.

3.1. Materiais

Os materiais do curso são produzidos de forma colaborativa pelos professores envolvidos. São usados dois sistemas: um LMS e um OJT desenvolvido pela instituição onde o curso é ministrado. O LMS é uma versão personalizada do Moodle (Modular Object Oriented Dynamic Learning Environment). Os professores usam o LMS para fornecer vídeos, slides derivados da bibliografia básica e complementar do curso, lições com perguntas avaliativas em tentativa semanal, fóruns para perguntas e anúncios, além de materiais e ferramentas adicionais. A Figura 1 mostra a página principal do LMS onde os alunos podem interagir¹.

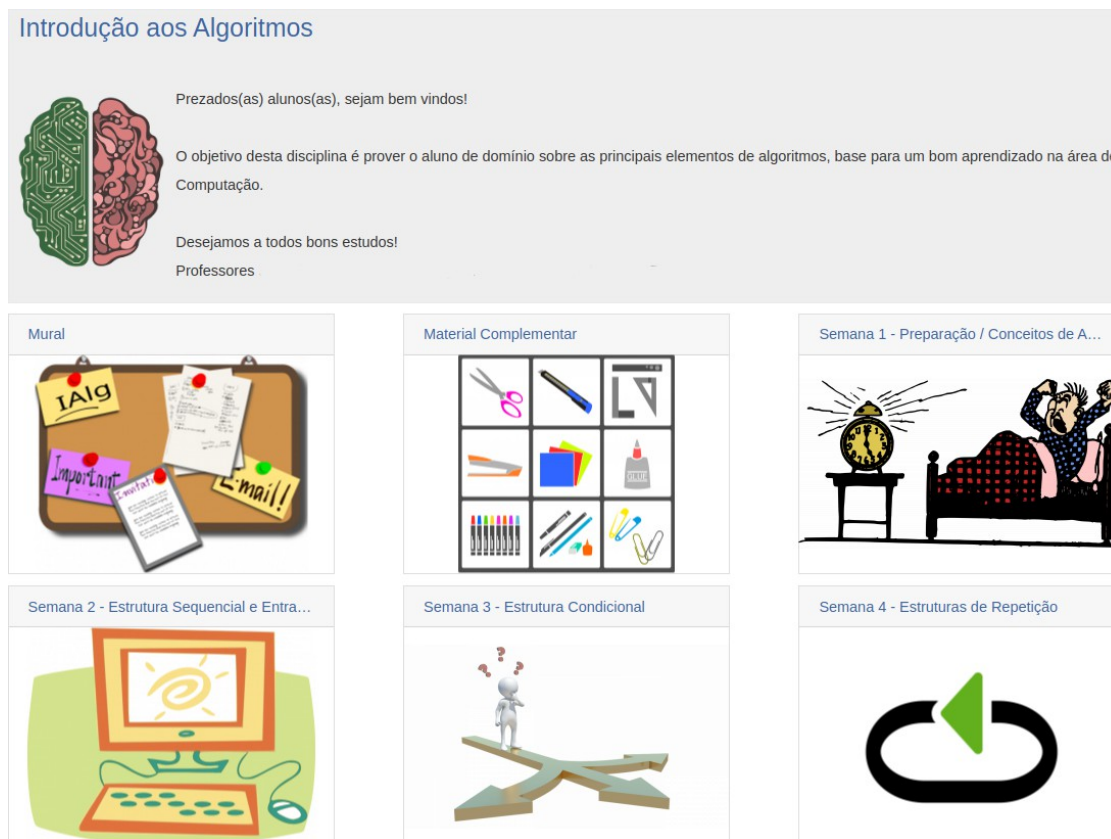


Figura 1. A sala de aula virtual no LMS

O curso tem todas as aulas gravadas em vídeo e disponibilizadas no YouTube². Há vídeos em que o professor explica os conceitos e outros em que são feitos exercícios para dar uma ideia aos alunos de como resolver problemas semelhantes. Os slides são simples, mas algumas alterações em relação ao anterior aumentaram a legibilidade. Eles também são usados nos vídeos gravados.

As lições consistem em um conjunto de páginas, cada uma terminando com perguntas que podem ser de múltipla escolha, verdadeiro ou falso, de correspondência, dissertativas, numéricas ou de resposta curta, para avaliar o conteúdo estudado. A Figura 2 mostra um exemplo de uma pergunta para a primeira semana.

A seção Materiais complementares e ferramentas contém informações sobre IDEs, links para o OJT e o canal do YouTube, instruções sobre como compilar usando o IDE

¹ Os números estão em português porque esse é o idioma usado no curso.

² Disponível em https://youtu.be/xbZL11yc8r8?si=Gt1kmGV-E3g_9Cmw

O que é um sistema computacional?

- ☐ É um conjunto de algoritmos que trabalham de forma coesa e ordenada
- ☐ É um conjunto de programas que trabalham de forma coesa e ordenada
- ☐ É um conjunto de dispositivos elétricos/eletrônicos que englobam CPU, memória e dispositivos de entrada e saída.
- ☐ É um conjunto de comandos enviados metodologicamente para um servidor

Enviar

Figura 2. Exemplo de lição

e linha de comando, entre outros recursos.

O OJT³ é um sistema on-line de apoio ao processo de ensino e aprendizagem de programação, desenvolvido no Departamento de Computação Aplicada da UFLA. Ele oferece suporte à avaliação automática de programas escritos em C, C++ e Python. O sistema emprega vários tipos de avaliadores de programas para avaliar as soluções enviadas pelos alunos, atribuindo uma pontuação de 0 a 100 e indicando o quanto o programa enviado está correto.

A Figura 3 apresenta um exemplo de problema para o aluno resolver no OJT. A estrutura é semelhante à dos problemas registrados no sistema. Ela inclui uma declaração explicando o problema, exemplos de entrada e saída e uma seção na parte inferior em que o aluno pode enviar seu arquivo de origem. No curso em questão, a avaliação principal é baseada em casos de teste, que indicam a semelhança entre os resultados gerados pelo programa enviado por um aluno e os resultados esperados, com base em um conjunto de casos de teste previamente registrados pelo instrutor (Figura 4). Para esse problema específico, é necessário registrar pelo menos 12 casos de teste no sistema, cada um correspondendo a um mês diferente do ano.

Na prática, o OJT é importante no processo de ensino e aprendizagem, pois permite aos alunos um certo grau de autonomia. Por meio desse sistema, os alunos recebem uma avaliação quantitativa inicial da correção de seu programa, o que lhes permite fazer correções, ajustar suas soluções e experimentar novas avaliações. Por exemplo, se o aluno não inserir todos os nomes de meses possíveis a serem exibidos na saída padrão, o resultado será incorreto e o OJT exibirá a mensagem explicando que "a quantidade de dados gravados pelo programa é diferente da quantidade de dados esperada" porque o mês correto está faltando.

3.2. Metodologia

Considerando o baixo desempenho dos alunos no curso, a experiência adquirida com aulas on-line e vários relatos na literatura, foi desenvolvida uma nova metodologia. Essa metodologia inclui 4 horas de aulas presenciais e 2 horas de aulas não presenciais (on-line) por semana. As horas on-line são obrigatórias para todos os alunos, enquanto as sessões práticas são divididas em grupos com base no curso, com um máximo de 25 alunos por grupo, sendo que a maioria deles contém 20 alunos.

A parte não presencial (NFF) do curso ocorre na sala de aula virtual do LMS. Espera-se que os alunos desenvolvam a capacidade de se organizar e estudar de forma autônoma, trazendo suas dúvidas para as aulas práticas. Nessa abordagem, os alunos os protagonistas de seu aprendizado, enquanto o professor atuará como mediador.

Os slides e as videoaulas foram produzidos para cobrir o conteúdo de uma semana.

³) <https://dredd.dac.ufla.br>

Questão 1: Estrutura Condicional - Exibindo Datas

As informações de data e hora podem ser exibidas para o usuário de diversas formas: podemos ter formato abreviado com números e barras ou ter a data escrita por extenso, por exemplo. Faça então um programa que recebe três números inteiros representando, respectivamente, um dia, um mês e um ano e exiba a data por extenso como no exemplo: 9 de maio de 2016.

Entradas:

- Três números inteiros representando um dia, um mês e um ano, respectivamente.

Saídas:

- Um texto com a data por extenso (ex: 20 de outubro de 2016).

Exemplo de entrada:

```
25
12
2016
```

Exemplo de saída:

```
25 de dezembro de 2016
```

Peso: 1

Nova Resposta: _____

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Nenhum arquivo selecionado.

Figura 3. Exemplo de pergunta no sistema Online Judge Tool

Yu e Gao (2022) mostram que os vídeos com duração inferior a 10 minutos são assistidos com mais frequência em um ambiente de sala de aula invertida. Portanto, as videoaulas produzidas são curtas, não ultrapassando esse limite para cobrir parte do conteúdo. Pode haver mais de uma videoaula para cada semana. As lições no LMS consistem em um conjunto de páginas para avaliar o conteúdo estudado. É possível configurar a navegação entre as páginas, dependendo da resposta escolhida pelo aluno, permitindo que ele avance para a próxima página, permaneça na mesma página ou retorne à página anterior.

A parte presencial do curso é realizada no laboratório de informática, com um aluno por máquina. O objetivo durante esse período é que os alunos solidifiquem seus conhecimentos e tirem suas dúvidas. Às terças-feiras, são realizadas atividades gamificadas com o uso de ferramentas como Kahoot (como também usado em de Sousa, Maranhão, Borges e Neto (2023)), Coding Dojo (como usado por Marinho, Moreira, Coutinho, Paillard e de Lima (2016); Scherer e Mór (2020); de Sousa et al. (2023), entre outros), revisão por pares (Brown, Narasareddygar, Singh, & Walia, 2019) e exercícios de desenvolvimento individual para reforçar o conteúdo da semana de forma prática. Os alunos recebem problemas para resolver e podem consultar o professor, os materiais do curso e seus colegas. Cada problema vem com uma descrição detalhada, exemplos de entrada e exemplos de saída. Cada exercício enfatiza o conteúdo estudado naquela semana, mas também pode incorporar conteúdo de semanas anteriores. Por exemplo, para resolver um problema que envolve estruturas de repetição, a solução pode exigir a leitura de um arquivo de entrada e o uso de

ID	Módulo	Parâmetro1	Parâmetro2
1389	basicoESA.pl	stdin 22 1 2015	stdout 22 de janeiro de 2015
1390	basicoESA.pl	stdin 12 4 2014	stdout 12 de abril de 2014
		stdin	

Figura 4. Exemplos de casos de teste para a pergunta da Figura 3

estruturas condicionais.

Nas aulas de quinta-feira, o aluno terá de desenvolver individualmente a solução para um problema e enviá-la para o OJT que realiza a correção de acordo com os casos de teste registrados no sistema. Além disso, os professores revisam manualmente o código enviado pelos alunos e fornecem feedback para ajudá-los a entender seus erros.

A avaliação do curso inclui atividades (aulas) da NFF, participação nas atividades de terça-feira, conclusão das atividades de quinta-feira, dois testes práticos e trabalho prático em equipe com base em um tópico escolhido pela equipe. Esse trabalho em equipe deve atender aos requisitos especificados para cobrir todo o conteúdo do curso.

Para avaliar os alunos usando a metodologia proposta, foi adotada uma nova distribuição de pontos, em relação aos semestres anteriores, para avaliar a participação em grupo (projeto prático), coletiva (Coding Dojo, Kahoot, Peer Review) e individual (teste individual), conforme apresentado na Tabela 1. A tabela mostra dois exemplos de semestres: o primeiro usando a metodologia tradicional e o último empregando a nova abordagem. Nessa tabela, se o semestre tiver mais de uma atividade com o mesmo nome, como "Exame", um número é adicionado ao final para melhor compreensão (por exemplo, Exame 1 e Exame 2).

Semestre	2022/1 (metodologia tradicional)	2023/2 (metodologia híbrida)
Critérios de avaliação	<p>Lista de atividades 1- 15%</p> <p>Exame 1 - 25%</p> <p>Lista de atividades 2 - 15%</p> <p>Exame 2 - 25% Projeto prático - 20%</p>	<p>Atividades às quintas-feiras (semanas 3 a 7) - 11%</p> <p>Exame 1 - 20%</p> <p>Atividades do NFF (semanas 1 a 7) - 8%</p> <p>Atividades às quintas-feiras (semanas 10 a 13) - 9%</p> <p>Atividades às terças-feiras - 5%</p> <p>Exame 2 - 20%</p> <p>Atividades da NFF (semanas 8 a 13) - 7%</p> <p>Projeto prático - 20%</p>

Tabela 1. Critérios de avaliação na metodologia tradicional e ativa

É importante observar que houve ajustes em 2022/2 e que 2023/2 apresenta pequenas melhorias e diferenças em relação a 2023/1. Esses ajustes se devem principalmente à ocorrência de feriados, reduzindo, por exemplo, o valor das atividades de quinta-feira. Entretanto, a mudança na filosofia de avaliação é perceptível quando se comparam esses dois semestres. O número de itens avaliativos aumentou e novos elementos foram acrescentados.

Os alunos começaram a ser avaliados. Em vez de pontuar as listas de atividades tradicionais que estavam disponíveis para uma semana, por exemplo, agora algumas atividades avaliadas destacam a sala de aula invertida e as atividades das terças-feiras são realizadas em sala de aula. Com as pontuações distribuídas de forma mais eficaz, agora é possível avaliar diferentes aspectos da aprendizagem de forma mais abrangente, permitindo um melhor envolvimento com os alunos.

4. Resultados

Considerando que vários estudos demonstraram a eficácia de diferentes metodologias de aprendizagem em cursos introdutórios de programação, o objetivo inicial era propor uma metodologia híbrida que integrasse aulas remotas e presenciais, juntamente com atividades gamificadas. Consequentemente, os resultados se concentram na comparação do número de aprovações de alunos na metodologia híbrida atual com os da abordagem de ensino tradicional anterior.

Um aspecto promissor foi que as diversas formas de avaliação incentivaram os alunos a acessar os materiais no LMS com mais frequência. Além disso, os alunos postaram perguntas nos fóruns de discussão, responderam a perguntas sobre o assunto e chegaram às aulas práticas com uma melhor compreensão do material e das perguntas específicas. Entretanto, alguns alunos ainda não desenvolveram esse hábito, muitas vezes esperando para revisar ou acessar o conteúdo somente durante as aulas práticas. Isso indica que há vários aspectos em que a estratégia poderia ser aprimorada.

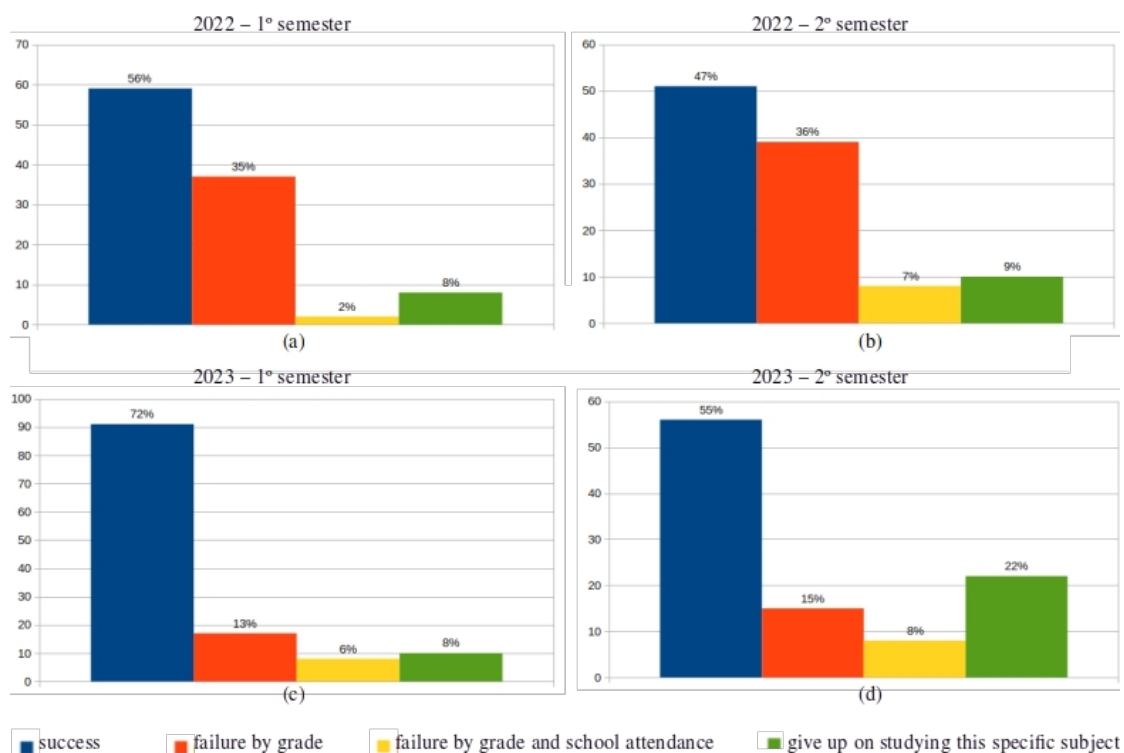


Figura 5. Resultados da metodologia tradicional e da metodologia híbrida

A Figura 5 apresenta os resultados obtidos em 2022 com a metodologia de ensino tradicional e em 2023 com a metodologia híbrida. Na Figura 5a, havia 106 alunos, de ambos os cursos de graduação, matriculados no curso no primeiro semestre

de 2022. Quase metade dos alunos foi reprovada no curso. Na Figura 5b, havia 108 alunos, de ambos os cursos de graduação, matriculados no curso no segundo semestre de 2022. Mais da metade dos alunos foi reprovada, por qualquer motivo. Na Figura 5c, mostra-se que 126 alunos de ambos os cursos de graduação estavam matriculados no curso no primeiro semestre de 2023. Essa foi a primeira vez que se usou a metodologia proposta, e o número de alunos reprovados foi significativamente menor em comparação com os semestres anteriores.

Esses resultados nos incentivam a manter a metodologia proposta, com pequenos ajustes. Os resultados no segundo semestre de 2023, apresentados na Figura 5d, estão relacionados a 101 alunos, de ambos os cursos de graduação, matriculados no curso. Houve uma diminuição no número de alunos bem-sucedidos em comparação com o primeiro semestre do mesmo ano. Entretanto, os resultados não foram piores do que os obtidos em 2022.

Embora o número de reprovações tenha aumentado no segundo semestre de 2023, acreditamos que a metodologia híbrida trouxe benefícios ao considerar o número total de alunos aprovados. Há vários motivos possíveis para esse comportamento, sobre os quais a seção seguinte discute em profundidade.

5. Discussão

Pelos resultados, fica claro que o número de alunos aprovados aumenta com nossa metodologia. No entanto, isso deve ser visto com cautela, pois esse estudo precisa de uma análise mais profunda. Um aspecto importante a ser observado é que esses alunos são novatos, estão no primeiro semestre dos cursos de graduação. Alguns deles não estão seguros para levar o curso até o fim. Outro aspecto importante é saber se o formato de avaliação do aluno funciona como um viés a favor da nova metodologia. Considerando que este é o primeiro estudo, o foco é apresentar nossas ideias e compartilhar nossa boa perspectiva.

Apesar de estar descrito no programa do curso, notamos que, às terças-feiras, os alunos ficavam particularmente entusiasmados com as atividades gamificadas. A atividade gamificada, especificamente usando o software Kahoot, despertou o entusiasmo dos alunos e proporcionou benefícios tangíveis ao processo de aprendizagem. A capacidade de fornecer feedback imediato e identificar as dificuldades da turma durante o jogo é uma vantagem significativa, permitindo intervenções oportunas para esclarecer conceitos antes de avançar. Isso não apenas aumenta o envolvimento dos alunos, mas também promove uma compreensão mais profunda do conteúdo.

Na maioria das aulas do Coding Dojo, usamos o formato kake, em que os alunos trabalham em duplas auto-organizadas (um piloto e um copiloto), usando apenas um computador e alternando-se periodicamente (a cada 5 minutos). Observamos que, durante as sessões, o copiloto precisava prestar atenção à codificação do parceiro para não se perder quando chegasse a sua vez. Os alunos que dominavam melhor os conceitos da disciplina explicavam a construção da solução para o parceiro. Podemos dizer que o Coding Dojo facilitou o aprendizado e a colaboração entre os alunos.

A principal atividade por trás da sala de aula invertida foram as atividades das quintas-feiras. Nesses dias, uma lista de exercícios era disponibilizada e cada aluno era selecionado aleatoriamente com uma ou duas atividades, a maioria delas incluindo a implementação no OJT. Um resultado negativo percebido pelos professores é que a maioria dos alunos só resolvia as atividades que eram indicadas, mesmo com tempo suficiente para tentar outros problemas. Isso nos leva à reflexão de que a maioria dos alunos não conseguia assumir a responsabilidade por seu próprio aprendizado, uma premissa importante no AL. No entanto, essas listas estavam disponíveis para serem praticadas antes do dia da atividade. Infelizmente,

grande parte dos alunos não usou esses recursos para aumentar seu aprendizado.

Esses resultados mostraram que a metodologia é bastante promissora, mas há vários aspectos a serem aprimorados e várias questões de aprendizagem a serem investigadas. Em um contexto em que as tecnologias de informação e comunicação têm um impacto significativo na educação, os professores precisam aperfeiçoar continuamente suas estratégias para orientar melhor os alunos em seu aprendizado.

Observações finais

O objetivo deste documento foi relatar uma metodologia de ensino híbrida combinada com metodologias ativas no ensino de introdução à programação de computadores. Essa metodologia foi aplicada em dois períodos e, embora tenham sido observados alguns resultados positivos iniciais, reconhece-se que é necessário investigar mais. Enfatiza-se que essa não é uma metodologia fechada, portanto, ajustes podem ser feitos conforme necessário, como a inclusão de outras atividades gamificadas ou a alteração da forma como os alunos são recompensados.

Como trabalho futuro, planejamos coletar dados sobre as opiniões dos alunos em relação à metodologia de ensino, reunir estatísticas sobre a visualização do material, identificar perfis de aprendizagem e mapear períodos de maior e menor envolvimento dos alunos durante todo o processo de aprendizagem. Como desafio metodológico, o grupo de professores está interessado em implementar o aprendizado em ritmo próprio, avaliando como incorporar essa abordagem em uma estrutura curricular relativamente amarrada.

Referências

- Açayır, G., & Açayır, M. (2018). A sala de aula invertida: Uma análise de suas vantagens e desafios. *Computers & Education*, 126, 334-345. doi: <https://doi.org/10.1016/j.compedu.2018.07.021>
- Al-Samarraie, H., Shamsuddin, A., & Alzahrani, A. I. (2020, 6). A flipped classroom model in higher education: a review of the evidence across disciplines (Um modelo de sala de aula invertida no ensino superior: uma revisão das evidências entre as disciplinas). *Educational Technology Research and Development*, 68, 1017-1051. doi: 10.1007/s11423-019-09718-8
- Barcelos, R., Tarouco, L., & Bercht, M. (2009). O uso de mobile learning no ensino de algoritmos. *Revista Novas Tecnologias na Educação*, 7(3), 327-337. doi: <https://doi.org/10.22456/1679-1916.13573>
- Brown, T., Narasareddygar, M. R., Singh, M., & Walia, G. (2019). Usando a revisão de código por pares para apoiar a pedagogia em um curso introdutório de programação de computadores. Em *2019 IEEE Frontiers in Education Conference (fie)* (pp. 1-7). doi: 10.1109/FIE43999.2019.9028509
- Caceffo, R., Gama, G., & Azevedo, R. (2018). Explorando abordagens de aprendizagem ativa para aulas de ciência da computação. Em *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 922-927). Nova York, NY, EUA: Association for Computing Machinery. doi: 10.1145/3159450.3159585
- de Moraes, R. P., da Costa, V. F., & Scholz, R. E. (2022). Mapeamento didático do ensino híbrido de programação nos ensinos técnico e superior no Brasil. *Revista Brasileira de Informática na Educação*, 30, 628-647. doi: 10.5753/rbie.2022.2611
- de Oliveira, M. G., da Silva, M. F., & Rodrigues, C. B. (2022). Curso híbrido baseado em moocs de loveace e oficinas presenciais para aprendizagem ativa e nobre de pensa-

- mento computacional e programação. In *Anais do XXVIII Workshop de Informática na Escola* (pp. 179-188). doi: <https://doi.org/10.5753/wie.2022.225669>
- de Sousa, J. S., Maranhao, D., Borges, P. V., & Neto, C. d. S. S. (2023). Uma análise da utilização da metodologia ativa coding dojo a uma plataforma de ensino-aprendizagem de algoritmos. In *Anais do XXXIV Simpósio Brasileiro de Informática na Educação* (pp. 127-138). doi: <https://doi.org/10.5753/sbie.2023.234703>
- Dias, A. I. d. A. S. (2022). Pensamento computacional para gerar redes inclusivas no contexto brasileiro pós-crise da pandemia de covid-19. In *Anais do I Workshop de Pensamento Computacional e Inteligência* (pp. 63-72). doi: <https://doi.org/10.5753/wpci.2022.226879>
- Dicheva, D., Dichev, C., Agre, G., & Angelova, G. (2015). Gamificação na educação: Um estudo de mapeamento sistemático. *Journal of educational technology & society*, 18(3), 75-88.
- Goletti, O., Mens, K., & Hermans, F. (2021). Experiências dos tutores no uso de estratégias explícitas em um curso de programação introdutória de aprendizagem baseada em problemas. Em *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education v. 1* (pp. 157-163).
- Grotta, A., & Prado, E. P. (2018). Um ensaio sobre a experiência educacional na programação de computadores: a abordagem tradicional versus a aprendizagem baseada em projetos. In *Anais do XXVI Workshop sobre Informática em Computação*.
- Marinho, C., Moreira, L., Coutinho, E., Paillard, G., & de Lima, E. T. (2016). Experiências no uso da metodologia coding dojo nas disciplinas básicas de programação de computadores em um curso interdisciplinar do ensino superior. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* (Vol. 5, p. 1097).
- Medeiros, R. P., Ramalho, G. L., & Neto, T. P. (2018). Uma revisão sistemática da literatura sobre ensino e aprendizagem de programação introdutória no ensino superior. *IEEE Transactions on Education*, 62(2), 77-90. doi: [10.1109/TE.2018.2864133](https://doi.org/10.1109/TE.2018.2864133)
- Morais, C., Neto, F. M., & Góes, A. (2020). Dificuldades e desafios no processo de aprendizagem de algoritmos e programação no ensino superior: uma revisão sistemática da literatura. *Pesquisa, Sociedade e Desenvolvimento*, 9(10). doi: <https://doi.org/10.33448/rsd-v9i10.9287>
- Onyema, E. M., Choudhury, T., Sharma, A., Atonye, F. G., Phylistony, O. C., & Edeh, E. C. (2021). Effect of flipped classroom approach on academic achievement of students in computer science (Efeito da abordagem de sala de aula invertida no desempenho acadêmico dos alunos em ciência da computação). Em *Data driven approach towards disruptive technologies: Proceedings of midas 2020* (pp. 521-533). Cingapura. doi: https://doi.org/10.1007/978-981-15-9873-9_41
- Pirker, J., Riffnaller-Schiefer, M., & Güntel, C. (2014). Motivational active learning: engaging university students in computer science education (Aprendizagem ativa motivacional: engajando estudantes universitários no ensino de ciência da computação). Em *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education* (pp. 297-302).
- Preuss, J. O., & de Lima, C. C. (2023). Ferramentas online na aprendizagem de programação de computadores no contexto do ensino remoto. *Revista Brasileira de Informática na Educação*, 31(1), 790-813. doi: <https://doi.org/10.5753/rbie.2023>

.2867

- Santos, A., Gorgônio, A., Lucena, A., & Gorgônio, F. (2015). A importância do fator motivacional no processo ensino-aprendizagem de algoritmos e lógica de programação para alunos repetentes. In *Anais do XXIII Workshop sobre Educação em Computação* (pp. 168-177).
- Scherer, A. P. Z., & Moura, F. N. (2020). Uso da técnica coding dojo em aulas de programação de computadores. In *Anais do XXVIII Workshop sobre Educação em Computação* (pp. 6-10).
- Silveira, S. R., Bertolini, C., Parreira, F. J., da Cunha, G. B., & Bigolin, N. M. (2021, maio). Impactos do ensino remoto na disciplina de paradigmas de programação durante o isolamento social devido à pandemia de covid-19. *Revista Ciência e Desenvolvimento*, 18(2), 200-213. doi: 10.25112/rgd.v18i2.2455
- Yew, E. H., & Goh, K. (2016). Aprendizagem baseada em problemas: Uma visão geral de seu processo e impacto na aprendizagem. *Health professions education*, 2(2), 75-79. doi: <https://doi.org/10.1016/j.hpe.2016.01.004>
- Yu, Z., & Gao, M. (2022). Effects of video length on a flipped english classroom (Efeitos da duração do vídeo em uma sala de aula de inglês invertida). *Sage Open*, 12(1), 21582440211068474. doi: 10.1177/21582440211068474
- Zanetti, H., & Oliveira, C. (2015). Técnicas de ensino de programação de computadores com lógica e pensamento computacional. In *Anais dos workshops do Congresso Brasileiro de Informática na Educação* (Vol. 4, p. 1236). doi: <https://doi.org/10.5753/cbie.webie.2015.1236>