

Metodologias Ativas no ensino EaD de Algoritmos e Programação para alunos do Ensino Médio

Pâmela Michele Cândida Cortez¹, Douglas Oliveira de Jesus¹,
Roberto Almeida Bittencourt¹,
Ana Lúcia Lima Marreiros¹

¹Departamento de Ciências Exatas – Universidade Estadual de Feira de Santana (UEFS)
Feira de Santana – BA – Brasil

pamela@uefs.br, dojesus@ecompu.uefs.br, roberto@uefs.br, allmmaia@uefs.br

Abstract. *This article presents the experience of teaching Algorithms and Programming with Python in a distance learning (DL) format as part of an extension project aimed at public high school students. We adopted flipped classroom and Problem-Based Learning (PBL) approaches, with the goal of not only teaching the programming language but also fostering the development of logical reasoning and computational thinking. In addition to the video lectures recorded by instructors, synchronous virtual meetings focused on practical activities are facilitated by students from the COmputer Engineering program at State UNiversity of Feira de Santana (Bahia). Data collected through a survey suggest a potential influence of the course in motivating participants to pursue higher education in technology fields, highlighting the importance of educational initiatives like this in attracting new talent.*

Resumo. *Este artigo apresenta uma experiência de educação a distância (EaD) em Algoritmos e Programação em Python como parte de um projeto de extensão voltado a estudantes do ensino médio de escolas públicas. Adotamos metodologias de classe invertida e Aprendizagem Baseada em Problemas (PBL), com o objetivo não apenas de ensinar a linguagem de programação, mas também de promover o desenvolvimento do raciocínio lógico e do pensamento computacional. Além das videoaulas gravadas por professores, os encontros virtuais síncronos, focados em atividades práticas, são mediados por estudantes do curso de Engenharia de Computação da Universidade Estadual de Feira de Santana (Bahia). Os dados coletados por meio de questionários indicam uma possível influência do curso na motivação dos participantes para ingressar em cursos superiores na área de tecnologia, destacando a relevância de iniciativas educacionais como esta para a atração de novos talentos.*

1. Introdução

A introdução da computação na educação básica é motivada por diferentes perspectivas: mercado de trabalho, domínio do pensamento computacional, letramento computacional e equidade de participação [Blikstein and Moghadam 2019]. Neste trabalho, o foco está principalmente nas perspectivas do mercado de trabalho e do domínio do pensamento computacional. Computação é relevante para o mercado de trabalho não apenas para profissionais de computação, mas também para uma variedade de profissões do Século XXI.

Por outro lado, pensamento computacional é uma habilidade que pode ser transferida para outros domínios e problemas corriqueiros.

O interesse dos jovens brasileiros na área de TI não tem acompanhado o crescimento do número de vagas de emprego. De 2011 a 2021, o número de ingressantes em cursos de TI cresceu apenas 0,6% [Freire et al. 2023]. Enquanto isso, o relatório “Panorama de talentos em tecnologia”, realizado pelo *Google for Startups*, aponta um déficit de 530 mil profissionais de TI até 2025 [Barrence 2023]. A falta de mão de obra qualificada tem impactado a capacidade de inovação do país, uma vez que as *startups* sofrem para atrair e reter talentos, o que impede o crescimento de seus negócios.

Cientes da necessidade de capacitar os jovens para as novas realidades do mercado, através da Base Nacional Comum Curricular (BNCC) [Brasil 2018], visando promover habilidades digitais e tecnológicas entre os estudantes da educação básica, o Conselho Nacional de Educação (CNE) formalizou a inclusão do ensino de computação neste nível educacional, que passou a ser obrigatório desde 2023 [MEC/CNE/CEB 2022]. Segundo o próprio CNE, Pensamento Computacional pode ser definido como o “conjunto de habilidades necessárias para compreender, analisar, definir, modelar, resolver, comparar e automatizar problemas e soluções de forma metódica e sistemática através do desenvolvimento da capacidade de criar e adaptar algoritmos” [Brasil 2022]. No entanto, a implementação prática da inclusão do Pensamento Computacional ainda enfrenta obstáculos na rede pública de ensino. Conforme apontado pela TIC Educação [NIC.BR 2020], mais de 60% dos professores relatam dificuldades no uso de tecnologias digitais em suas atividades pedagógicas, revelando uma disparidade significativa entre as exigências do currículo e a formação técnica dos educadores.

Essa realidade, aliada à necessidade de atrair mais jovens para a área de TI, tem inspirado diversas iniciativas extensionistas de ensino de programação para estudantes do ensino médio [Menezes et al. 2023, Pimentel et al. 2024], mas muitos dos cursos oferecidos são de curta duração. Com o ensino continuado de programação através de metodologias ativas de ensino, buscamos contribuir para a formação de jovens sobretudo em relação à capacidade de criar algoritmos para a resolução de problemas, mas também desejamos derrubar mitos relacionados a TI ser apenas para *nerds e geeks*. Nosso projeto de extensão visa não apenas ensinar programação em Python em maior profundidade, mas também aproximar jovens de escolas públicas e estudantes do curso de Engenharia de Computação da Universidade Estadual de Feira de Santana. A interação entre os dois grupos de estudantes torna o sonho da universidade mais concreto para os estudantes do ensino médio de escolas públicas, ao mesmo tempo que permite aos estudantes de graduação desenvolver *soft skills* importantes em relação ao trabalho em grupo e liderança.

A seguir, apresentamos em detalhes o formato do curso de extensão oferecido. Em resultados, trazemos as observações colhidas através de questionários respondidos tanto pelos estudantes do Ensino Médio quanto de Engenharia de Computação. Por fim, a seção de conclusões sintetiza as contribuições e limitações deste projeto de extensão.

2. Metodologia

Descrevemos aqui o cenário que originou o curso, além do curso propriamente dito, os participantes, sua abordagem de ensino-aprendizagem e sua organização.

2.1. Cenário

A Resolução MEC/CNE/CES nº 7 de 2018 estabelece as diretrizes para a Extensão na Educação Superior, destinando ao menos 10% da carga horária total dos cursos de graduação às atividades de extensão [Brasil 2018]. No curso de Engenharia de Computação da Universidade Estadual de Feira de Santana (com carga horária total de 3600 h), implementamos a curricularização da extensão em quatro partes de 90 horas cada: criando três novas disciplinas obrigatórias puramente extensionistas – uma na área de hardware, outra na área de software e outra atrelada a projeto de extensão (denominada Unidade Curricular de Extensão, UCE) – e atribuindo caráter extensionista a algumas disciplinas. Fica a cargo do professor decidir o formato das atividades extensionistas em cada oferta.

Dentre os possíveis formatos, focamos em *serviços à comunidade* através da disciplina extensionista na área de software e em *cursos* através da UCE, visto que já coordenávamos um projeto de extensão na área de ensino anteriormente. Oferecemos à comunidade o desenvolvimento de software, sobretudo para Organizações da Sociedade Civil (OSC) da região. O primeiro produto entregue foi um aplicativo e um sistema Web para uma OSC do litoral do estado dedicada à preservação da Mata Atlântica.

Na área de ensino, temos oferecido dois cursos: Algoritmos e Programação em Python (desde 2023) e Desenvolvimento de Aplicativos (desde 2024). Em 2025, além de prosseguirmos com o oferecimento de novas turmas, iniciaremos a oferta de mais um curso: Treinamento para a OBI¹, que, assim como o último curso, tem como pré-requisito o conhecimento de programação em Python. A experiência com o oferecimento da primeira turma do curso de Desenvolvimento de Aplicativos foi detalhada em [de Jesus and Cortez 2024].

2.2. Curso EaD de Algoritmos e Programação

O curso EaD de Algoritmos e Programação em Python destinado aos estudantes, professores e egressos do ensino médio de escolas públicas é composto por videoaulas gravadas por professores do curso de Engenharia de Computação da Universidade Estadual de Feira de Santana, dois encontros síncronos semanais e um exercício avaliativo desenvolvido ao longo do curso. Seguimos ementa similar ao da disciplina Algoritmos e Programação I², que inclui: noção de algoritmo; dado; variável; operadores e precedência; estruturas condicionais e de repetição; funções; listas e matrizes; arquivos. Incluímos também interface gráfica (empregando o *framework pyGame*), com o intuito de motivá-los ainda mais a desenvolver o trabalho avaliativo proposto. Todo o material, incluindo cronograma detalhado, está disponível em nosso site³.

2.2.1. Participantes

Em 2023, recebemos exatamente 500 inscrições, das quais 431 eram de fato estudantes de escolas públicas, sendo que metade respondeu não ter nenhum conhecimento prévio em programação. Destes, selecionamos 60 jovens entre egressos e estudantes do 3º ano

¹<https://olimpiada.ic.unicamp.br>

²<https://www.ecomp.uefs.br/academico/curriculo>

³<https://sites.google.com/view/jeditemple/>

do ensino médio de escolas públicas do estado (27% do sexo feminino e 73% do sexo masculino). Apenas 18 foram aprovados ao final do curso (30% dos ingressantes, sendo 11% do sexo feminino e 89% do sexo masculino). Em 2024, expandimos a quantidade de vagas para 120, oferecidas semestralmente, para poder contemplar estudantes do 2º ano do ensino médio. Recebemos 235 e 355 inscrições no início e meio do ano, respectivamente. Como o semestre 2024.2 ainda não está concluso, não temos dados concretos sobre aprovações e desistências e, portanto, os resultados apresentados neste trabalho são referentes a 2023 (primeira oferta).

Participam também deste projeto, no papel de mentores, os estudantes cumprindo carga horária obrigatória de extensão. São eles que efetivamente acompanham e facilitam as atividades de ensino de programação.

2.2.2. Abordagem de Ensino-Aprendizagem

Nossa escolha por metodologias ativas para o ensino de programação se dá pela própria estrutura do nosso curso de Engenharia de Computação, onde empregamos a metodologia PBL com sucesso desde a criação do curso em 2003 [Ribeiro and Bittencourt 2019].

A Metodologia de Aprendizagem Baseada em Problemas (PBL – *Problem-Based Learning*) é uma abordagem pedagógica ativa que promove a construção do conhecimento por meio da resolução de problemas reais ou simulados. Diferente do modelo tradicional, no qual o conteúdo é transmitido de forma passiva, o PBL incentiva a autonomia do aluno, o trabalho em equipe, o pensamento crítico e a aplicação prática dos conceitos teóricos [Savery and Duffy 1995]. Segundo Barrows [Barrows 1996], essa metodologia é centrada no aluno, que assume um papel ativo em seu processo de aprendizagem, enquanto o professor atua como facilitador, orientando e estimulando a investigação e a reflexão. Estudos apontam que o PBL melhora a retenção do conhecimento e desenvolve habilidades socioemocionais essenciais para a prática profissional [Schmidt et al. 2011].

A Metodologia de Aprendizagem Baseada em Problemas tem sido amplamente aplicada em diversos contextos acadêmicos, incluindo a extensão universitária. Estudos recentes demonstram que a aplicação do PBL em projetos de extensão universitária fortalece a formação cidadã e o desenvolvimento de competências como a autonomia e a resolução de problemas [Silva and Oliveira 2022]. Iniciativas como o uso do PBL em programas de educação em saúde e em práticas de empreendedorismo social reforçam a eficácia dessa metodologia ao aproximar a universidade da sociedade e ao estimular o protagonismo discente [Souza and Mendes 2021, Almeida and Costa 2023]. Além disso, evidências apontam que a utilização do PBL em cursos de extensão favorece a aprendizagem significativa e interdisciplinar, tornando-se uma estratégia pedagógica eficaz [Pereira and Santos 2023].

A metodologia PBL, aplicada já com os calouros do curso, contribui com a motivação dos estudantes, sobretudo em relação à autoconfiança e satisfação nos estudos [Souza and Bittencourt 2019]. Por essa razão, a atividade avaliativa é o desenvolvimento de um pequeno jogo construído ao longo do curso, utilizando-se da metodologia PBL. A opção por sempre propor o desenvolvimento de um joguinho se dá pela nossa experiência na docência da disciplina introdutória de programação na graduação, onde observamos que os estudantes se sentem bem mais motivados a desenvolver o problema proposto

quando este inclui desenvolvimento de jogos. As discussões para elucidar como resolver o problema seguem o *framework* de PBL, onde os próprios estudantes assumem diversos papéis (e.g., coordenador, secretário, membro regular), assumindo assim protagonismo nos estudos, promovendo a colaboração e o pensamento lógico ao liderarem a solução do problema em equipe.

Além do PBL, empregamos também a metodologia de classes invertidas, que é um modelo pedagógico que inverte a lógica tradicional de ensino, transferindo a parte expositiva para o ambiente virtual e reservando o tempo em sala para atividades práticas e colaborativas. Nesse modelo, os alunos acessam previamente os conteúdos teóricos por meio de vídeos, leituras ou outros materiais digitais, permitindo que o tempo presencial seja dedicado a discussões, resolução de problemas e aplicação do conhecimento [Bergmann and Sams 2012]. De acordo com Bishop e Verleger [Bishop and Verleger 2013], essa abordagem aumenta o engajamento dos estudantes, promove a aprendizagem ativa e permite uma personalização mais eficaz do ensino. Dessa forma, os estudantes têm um primeiro contato com os conceitos teóricos antes da aula síncrona [Valente 2014], permitindo assim que o tempo em sala (virtual) seja empregado para a realização de atividades mais complexas, como a aplicação dos conceitos estudados [National Research Council 2000].

Em consonância com as regras de ouro da metodologia de classes invertidas [Flipped Classroom Field Guide 2014], as atividades em sala de aula envolvem resolução de exercícios e estes são corrigidos na mesma aula para que recebam *feedback* imediato. Assim como as videoaulas, as listas de exercícios foram cuidadosamente preparadas pelos professores para garantir que o conteúdo fosse acessível e relevante, facilitando a compreensão e promovendo a consolidação dos conceitos ensinados através da prática.

Através das listas de exercícios, os encontros síncronos são direcionados para as atividades práticas de programação. Dado o caráter prático das aulas, os estudantes reconhecem a necessidade de estudar as videoaulas antes da respectiva aula síncrona para estarem aptos a efetivamente programar junto com os colegas e mentor. Através da metodologia empregada, asseguramos que os estudantes estão constantemente colocando o que aprendem em prática e, portanto, aprendendo de fato a programar. Além disso, como os estudantes vêm para os encontros síncronos com algum conhecimento prévio e certo grau de nivelamento entre eles, as dúvidas trazidas sobre os conceitos ensinados nas videoaulas podem ser dirimidas via discussão entre os presentes, de forma a fomentar a participação dos estudantes e contribuir com o desenvolvimento do raciocínio lógico.

Durante a primeira turma do curso, experimentamos garantir a realização das listas de exercícios através de um ambiente juiz online [Galvão et al. 2016], mas alguns participantes relataram que o sistema acusou erro por qualquer diferença na saída os desmotivava. Já com o emprego da metodologia ativa, os estudantes realizam as atividades práticas em um contexto mais colaborativo, tornando o aprendizado mais leve e divertido. Além das aulas síncronas para resolução das listas de exercícios e elucidação das dúvidas, distribuímos algumas sessões PBL ao longo do semestre para que possam, também colaborativamente, solucionar o problema PBL proposto.

2.2.3. Organização do Curso

Utilizamos o universo *Star Wars* para contextualizar o ambiente com um pouco de humor: os estudantes são chamados de *Padawans* e os mentores, *Jedi Masters*. O acompanhamento das atividades se dá através de reuniões semanais entre professores e mentores, contando ainda com a figura de um *Yoda* (estudante encarregado de organizar o ambiente Discord, responder dúvidas nos canais gerais e receber as dúvidas dos mentores). Ao *Yoda* é disponibilizado o contato telefônico dos professores para que as dúvidas dos mentores sejam dirimidas imediatamente. Os estudantes (*Padawans*) se dividem em turmas com número máximo de 10 estudantes de acordo com suas preferências de horário (cada mentor define livremente os horários nos quais deseja dar aula). A cada turma é associado o nome de um planeta no universo *Star Wars* à escolha de cada *Jedi Master*. Pode parecer irrelevante, mas isso cria um espírito de comunidade e pertencimento, eles chegam a expressar certo carinho pelo antigo planeta quando, por motivos de evasão, precisamos unir duas turmas em uma única.

Os encontros síncronos são conduzidos pelos estudantes matriculados na disciplina UCE, que tem como pré-requisito a disciplina Algoritmos e Programação I. Nenhum encontro síncrono é conduzido pelos próprios docentes, exceto a aula inicial, na qual explicamos as metodologias ativas empregadas no curso e enfatizamos a necessidade dos estudantes assistirem as videoaulas antes das aulas síncronas. As aulas síncronas são realizadas através da plataforma Discord, onde também indicamos semanalmente os assuntos que devem ser estudados, conforme a Figura 1.

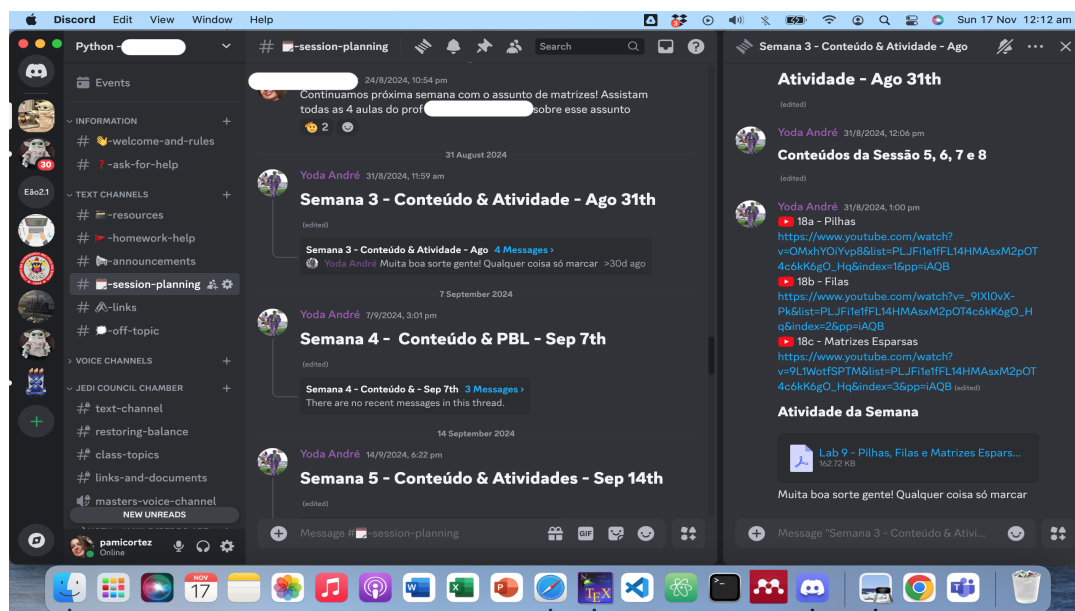


Figura 1. Organização do servidor Discord

Iniciamos este projeto de extensão em 2023 oferecendo apenas 60 vagas em um curso anual de programação cuja ementa era a mesma da disciplina inicial do curso de Engenharia de Computação da Universidade Estadual de Feira de Santana. A partir de 2024.2, adequamos o curso de 180 para 90 horas (60 h síncronas, 15 h assíncronas e 15 h de atividades práticas extra-classe), apenas reduzindo o tamanho de cada lista de

exercícios e retirando da ementa os assuntos de recursão, busca e ordenação e classes como contêineres (registros). Isso se deu tanto pelo *feedback* fornecido pelos estudantes quanto pela introdução de um novo curso a partir de 2025: Treinamento para a OBI (Olimpíada Brasileira de Informática).

Como dito anteriormente, a avaliação se dá através do código entregue individualmente para o problema PBL proposto. Nas aulas práticas cujo assunto é a interface gráfica, os estudantes podem inclusive compartilhar a tela caso queiram receber ajuda ou então ensinar os colegas sobre a implementação desta funcionalidade. Os certificados do curso são emitidos pela Pró-reitoria de Extensão da instituição apenas para os estudantes aprovados, i.e., aqueles com frequência acima de 70% nos encontros síncronos e nota igual ou superior a 7,0 no trabalho prático.

2.3. Avaliação da Experiência

Estudantes e mentores são convidados a responder questionários visando avaliar como adequar o curso para melhor aproveitamento e aprendizado. Por exemplo, através da análise dos questionários respondidos pelos estudantes da primeira turma do curso, pudemos repensar a complexidade das listas de exercícios e abandonar o emprego de juiz online como ambiente virtual de realização das atividades práticas. Na próxima seção, discutimos alguns pontos observados nesses questionários.

3. Resultados e Discussão

Nesta seção, antes de discutir os resultados observados objetivamente através dos questionários, elencamos algumas observações que nos permitiram adequar o curso ou permanecer confiantes acerca da abordagem de ensino-aprendizagem adotada:

1. Os estudantes da primeira e segunda turma nos mostraram ser desnecessário dividir o curso em dois semestres, pois eles aprendem tão rápido quanto os estudantes que acabaram de ingressar no curso de graduação. Como relataram que a parte inicial do curso foi demasiado lenta, a partir de 2024.2, o curso passou a ser ofertado em apenas um semestre.
2. Acerca do PBL, apesar da maioria relatar que o problema proposto parecia muito difícil em um primeiro momento, o *feedback* recebido indica que eles se sentem muito satisfeitos por produzir algo que inicialmente julgavam bastante desafiador. Isso contribui para desmistificar a ideia de que programação é apenas para gênios.

A seguir, apresentamos algumas estatísticas e relatos dos participantes.

3.1. Relatos de experiência dos estudantes

Nesta seção, apresentamos as respostas ao questionário respondido pelos primeiros 60 estudantes após quase três meses de curso. Em relação à expectativa dos estudantes, 61,7% (37 estudantes) disseram que o curso superou suas expectativas, 35% (21 estudantes) responderam que o curso foi o que esperavam de uma primeira disciplina de programação, enquanto apenas dois estudantes relataram esperar muito mais do curso. Em relação à didática do mentor, 76,7% consideraram ótima, 21,7% boa, 1,7% responderam regular. Nenhum estudante considerou a didática do mentor (estudante de Engenharia de Computação) ruim ou péssima. Em relação à qualidade das interações e/ou mediações do

mentor e sua disponibilidade, 85% relataram ser ótima enquanto 15% responderam ser de boa qualidade. Como esperávamos, o mentor é uma referência positiva para a grande maioria dos estudantes (85%), que o consideram essencial no seu aprendizado e os mantêm entusiasmados com o curso (93,3%).

Também fizemos diversas perguntas sobre a percepção dos estudantes em relação à importância do curso em seu aprendizado, metodologia e motivação pessoal. Devido à quantidade de questões aplicadas, detalhamos as respostas através da Figura 2. 73,3% dos estudantes responderam que a metodologia cria senso de comunidade e 68,3% veem uma influência positiva do curso em outras disciplinas. A importância do curso é clara para 90% dos estudantes, que concordam total ou parcialmente com a afirmação “as coisas que estou aprendendo serão úteis para mim”. Apenas 3,3% não veem benefício no curso oferecido. 80% dizem gostar de estudar programação, 78,3% sentem satisfação com o curso e 88,3% se sentem satisfeitos com o aprendizado proporcionado pela oportunidade. Em relação ao nível de dificuldade e volume de trabalho, 71,7% acredita que o nível seja razoável e que a quantidade de trabalho é apropriada para um curso de programação.



Figura 2. Respostas dos estudantes ao questionário de avaliação

Sobre o nível de dificuldade em relação a cada conceito aprendido, podemos observar na Figura 3 que o assunto passou a ser considerado difícil ou muito difícil para uma quantidade razoável de estudantes a partir de estruturas condicionais aninhadas. *Soft skills* como análise dos enunciados e autogerenciamento das atividades também não são consideradas atividades fáceis pelos estudantes, o que reforça a importância de preparar melhor os jovens para situações que demandam autonomia e responsabilidade.

Em suma, os estudantes gostaram da oportunidade de estudar em pequenos grupos com a ajuda de um estudante de Engenharia de Computação. Alguns dos *feedbacks* orgânicos recebidos sobre a metodologia de estudo em grupo com o acompanhamento de um mentor foram:

- “Um aprende com o outro, troca opiniões e modos diferentes de resolver um problema ajuda muito, eu aprendi muito com meus colegas e eles comigo.”

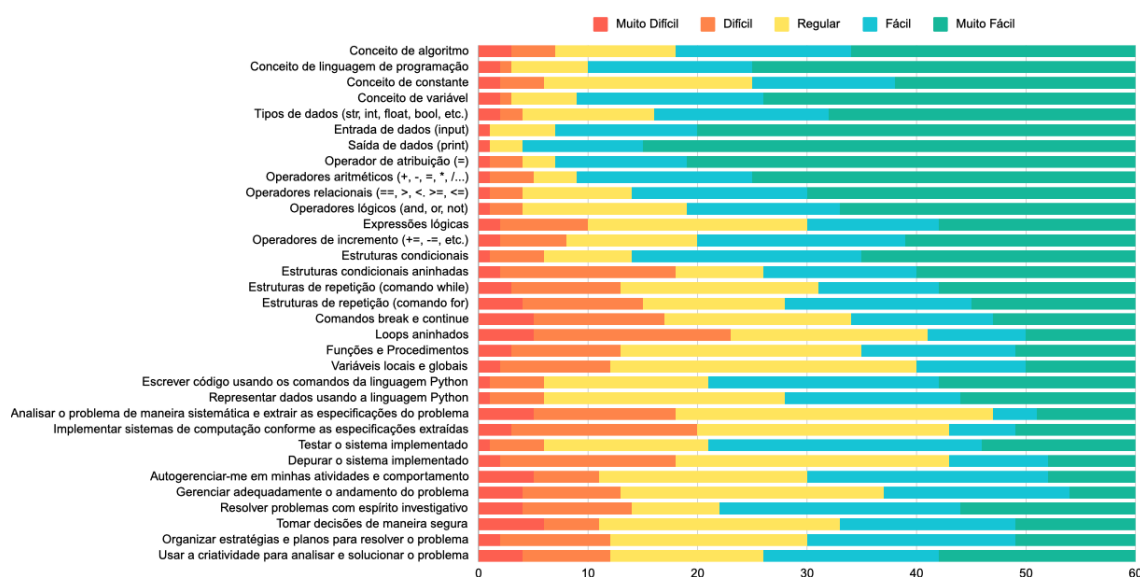


Figura 3. Facilidade em aprender os conceitos/recursos e habilidades

- “Ajuda a socializar, é mais fácil de tirar dúvidas (dá pra tirar dúvida tanto com o Jedi quanto com outros colegas), todos se ajudam, e as aulas são muito divertidas e didáticas.”
- “Ensina de fato e incentiva a ser autodidata.”
- “O aprendizado é bem mais otimizado, muitas vezes eu consigo aprender até com os próprios colegas, com formas diferentes que eles usaram pra resolver determinado problema. A metodologia também tem me ajudado a pensar mais analiticamente, e fragmentar os problemas em outros menores, a interpretar melhor.”
- “Os professores [mentores] são como amigos, e seu jeito de conversar e ensinar é muito bom por isso.”

Mas nem sempre os estudantes estão satisfeitos com o comprometimento dos colegas da turma, como demonstrado neste comentário: “A questão da interação com os colegas no meu planeta é o que mais precisa melhorar, são poucas pessoas, e a grande maioria não interage tanto, o que atrapalha toda a aula em grupo e até mesmo o Jedi, que tenta uma interação que as vezes não é respondida, a inatividade de alguns membros também atrapalha, por isso uma distribuição diferente seria interessante”. Uma possível solução é atribuir uma nota (de 0 a 4, por simplicidade) à participação dos estudantes a cada aula para deixar evidente a obrigatoriedade de uma postura ativa durante os encontros síncronos. Testaremos também alguns encontros nos fins de semana apenas para diversão, com jogos online, visando fomentar amizade entre os membros, pois observamos que as turmas com melhor interação são compostas por estudantes que tem laços afetivos já consolidados (são colegas da mesma escola).

3.2. Análise dos motivos de desistência

Dos quase 40 desistentes em 2023, 23 estudantes (48% do sexo feminino e 52% do sexo masculino) responderam ao questionário específico sobre os motivos da desistência. A Figura 4 mostra as opiniões dos estudantes em relação a 11 possíveis motivos.

As principais causas apontadas foram a falta de tempo e conflito de horário com outras atividades (indicadas por 48% dos entrevistados). Em seguida, com apenas 17%

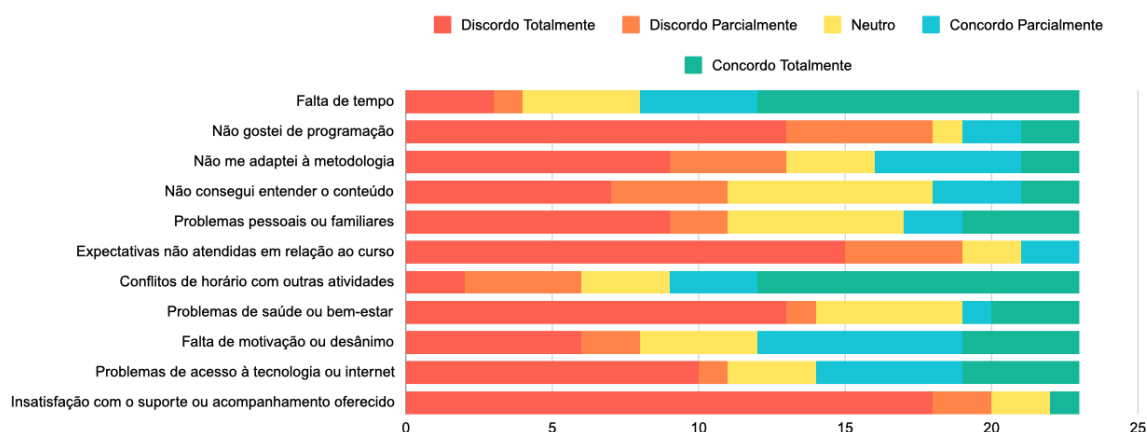


Figura 4. Possíveis motivos de desistência do curso

dos votos, problemas pessoais, técnicos e falta de motivação. Apenas quatro estudantes citaram não ter gostado de programação e cinco não se adaptaram às metodologias ativas, o que pode ser a causa de não conseguir entender o conteúdo (cinco estudantes) e relatar que o curso não atendeu às expectativas (dois estudantes). Apenas um estudante relatou insatisfação com o suporte oferecido.

3.3. Relatos de experiência dos mentores

Além de cumprir 25% da carga horária obrigatória de extensão, os estudantes de Engenharia de Computação tem a oportunidade de desenvolver *soft skills* importantes tanto para aqueles que optarem pela vida acadêmica (experiência docente), quanto aqueles que seguirem carreira em empresas privadas, uma vez que a interação com alunos pode ajudá-los a desenvolver empatia e liderança. De fato, termos relacionados a “habilidades sociais e comunicativas” e/ou liderança foram citados por todos os mentores. Um estudante destacou que o projeto “*me ajudou não só a reforçar meu conhecimento técnico, mas também desenvolver habilidades interpessoais e de comunicação, áreas nas quais sempre senti dificuldade.*”.

Alguns mentores demonstraram o reconhecimento de que sempre aprendemos algo a mais através da experiência docente com relatos como: “Você também aprende a medida que ensina”. Alguns expressaram ainda certa realização pessoal pelo impacto positivo que puderam causar: “*A conexão com os padawans e saber que, mesmo que pouco, o que eu fiz talvez [os] tenha ajudado a decidir o que querem para o futuro.*”.

Eles evidenciaram entender a proposta de mentoria versus monitoria. Um deles resumiu da seguinte forma: “*A mentoria é um relacionamento mais profundo, em que um indivíduo com mais experiência e conhecimento orienta e apoia outra pessoa no desenvolvimento de suas habilidades, conhecimentos e perspectivas. A mentoria é mais ampla em termos de objetivos e oferece um espaço seguro para discutir questões profissionais e pessoais.*”. Nenhum estudante relatou não ter gostado da experiência e dois terços afirmaram ser provável ou muito provável a participação no projeto ainda que não tivessem que cumprir carga horária obrigatória de extensão.

4. Considerações Finais

Apesar da evasão elevada (70%), consideramos que a iniciativa tem sido bem-sucedida. Os resultados mostram que os benefícios são claros tanto para os estudantes do curso quanto para os mentores (nossos estudantes de extensão). Alguns dos estudantes da turma de 2023 são hoje estudantes do nosso curso de Engenharia de Computação da Universidade Estadual de Feira de Santana. Todos já tinham interesse por TI antes do curso, mas relataram que o curso aumentou significativamente a confiança na decisão pela área.

Consideramos ainda que a interação entre estudantes do ensino médio de escolas públicas e estudantes de engenharia é um dos pontos fortes do projeto. Um jovem relatou: “É um curso realmente excelente, me deu um objetivo de vida, me mostrou algo que eu não tinha... um sonho, agora com força e determinação sinto que um dia posso me tornar aquilo que desejo, muito obrigado por me dar essa chance de sonhar”. Permitir que esses sonhos surjam é, ao nosso ver, mais importante que o próprio conteúdo ensinado.

A partir do segundo semestre de 2024, passamos a oferecer vagas também para professores do ensino médio interessados em aprender programação e, em 2025, iremos expandir a oferta para incluir os estudantes dos cursos de licenciatura da nossa Instituição. Acreditamos que essas iniciativas podem promover ainda mais o interesse dos jovens em TI ao engajá-los em competições da área de programação e preparar os professores atuais e do futuro para fomentar a participação dos jovens nesses eventos.

Em trabalhos futuros, desejamos também investigar o impacto de dados demográficos como disponibilidade de equipamentos adequados (computador ou notebook) na evasão, além de fazer um estudo do impacto do pensamento computacional no desempenho dos jovens em assuntos correlatos. Através de testes específicos em matemática e interpretação de textos aplicados antes, durante e ao final do curso, tentaremos medir a influência da lógica de programação no aprimoramento do raciocínio lógico.

Referências

- Almeida, B. R. and Costa, F. M. (2023). Aprendizagem baseada em problemas no empreendedorismo social: Um relato de experiência em extensão universitária. *Revista de Inovação e Extensão*, 10(1):89–102.
- Barrence, A. (2023). A escassez dos profissionais de tecnologia no Brasil e seu consequente impacto no ecossistema de startups. Blog do Google Brasil. Disponível em: <https://blog.google/intl/pt-br/produtos/a-escassez-dos-profissionais-de-tecnologia-no-brasil-e-seu-consequente-impacto-no-ecossistema-de-startups>. Acesso em: 27 de outubro de 2024.
- Barrows, H. S. (1996). Problem-based learning in medicine and beyond: A brief overview. *New Directions for Teaching and Learning*, 1996(68):3–12.
- Bergmann, J. and Sams, A. (2012). *Flip Your Classroom: Reach Every Student in Every Class Every Day*. International Society for Technology in Education.
- Bishop, J. L. and Verleger, M. A. (2013). The flipped classroom: A survey of the research. *ASEE National Conference Proceedings*, 30(9):1–18.
- Blikstein, P. and Moghadam, S. H. (2019). Computing education. *The Cambridge handbook of computing education research*, pages 56–78.

- Brasil (2018). *Base Nacional Comum Curricular*. Ministério da Educação, Brasília, DF. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518-versaofinal_site.pdf. Acesso em: 19 de setembro de 2024.
- Brasil (2018). *Resolução MEC/CES/CNE nº 7, de 18 de dezembro de 2018*. Ministério da Educação, Conselho Nacional de Educação, Câmara de Educação Superior, Brasília, DF. Disponível em: <https://www.gov.br/mec/pt-br/cne/resolucoes/resolucoes-cne-ces-2018>. Acesso em: 27 de outubro de 2024.
- Brasil (2022). *Parecer nº 2, de 17 de fevereiro de 2022. Normas sobre Computação na Educação Básica – Complemento à Base Nacional Comum Curricular (BNCC)*. Ministério da Educação, Conselho Nacional de Educação, Câmara de Educação Superior, Brasília, DF. Disponível em: <http://portal.mec.gov.br/docman/fevereiro-2022-pdf/236791-anexo-ao-parecer-cneceb-n-2-2022-bncc-computacao/file>. Acesso em: 27 de outubro de 2024.
- de Jesus, D. O. and Cortez, P. M. C. (2024). Metodologias ativas no ensino de frameworks para o desenvolvimento de aplicativos mobile: uma experiência com estudantes do ensino médio. In *Anais da XXIV Escola Regional de Computação Bahia, Alagoas e Sergipe*, Salvador, BA. SBC.
- Flipped Classroom Field Guide (2014). Portal Flipped Classroom Field Guide. Disponível em: <https://www.shortcutstv.com/wp-content/uploads/2023/06/Flipped-Classroom-Field-Guide.pdf>. Acesso em: 10 de outubro de 2024.
- Freire, F., Ramos, R., Morelli, K. C., and Morelli, M. H. (2023). Mapa do Ensino Superior no Brasil. Instituto Semesp, São Paulo, SP. Disponível em: <https://www.semesp.org.br/wp-content/uploads/2023/06/mapa-do-ensino-superior-no-brasil-2023.pdf>. Acesso em: 15 de outubro de 2024.
- Galvão, L., Fernandes, D., and Gadelha, B. (2016). Juiz online como ferramenta de apoio a uma metodologia de ensino híbrido em programação. *Simpósio Brasileiro de Informática na Educação (SBIE)*, 27(1):140.
- MEC/CNE/CEB (2022). Resolução CNE/CEB 1/2022 - Normas sobre Computação na Educação Básica – Complemento à Base Nacional Comum Curricular (BNCC).
- Menezes, G., Theodoro, L., Araújo, R., and Pereira, J. (2023). Ensino de programação para estudantes do ensino médio de escolas públicas com igualdade de gênero. In *Anais Estendidos do III Simpósio Brasileiro de Educação em Computação*, pages 16–17, Recife, PE. SBC.
- National Research Council (2000). *How People Learn: Brain, Mind, Experience, and School*. The National Academies Press, Washington, DC. Disponível em: <http://nap.edu/9853>. Acesso em: 10 de outubro de 2024.
- NIC.BR (2020). Pesquisa sobre o uso das tecnologias de informação e comunicação nas escolas brasileiras. Núcleo de Informação e Coordenação do Ponto BR. Disponível em: https://cetic.br/media/docs/publicacoes/2/20211124200326/tic_educacao_2020_livro_eletronico.pdf. Acesso em: 19 de setembro de 2024.
- Pereira, L. V. and Santos, J. R. (2023). A aprendizagem baseada em problemas como ferramenta interdisciplinar em projetos de extensão. *Revista Interdisciplinar de Extensão*, 9(4):67–80.

- Pimentel, M. d. G. C., Eusébio, J., Goularte, R., Leite, U. V., and Picoli, H. S. (2024). Cursos curtos online síncronos para meninas do ensino médio e concluintes e para professoras do ensino básico. In *Anais Estendidos do XXX Simpósio Brasileiro de Sistemas Multimídia e Web*, pages 203–209, Juiz de Fora, MG. SBC.
- Ribeiro, A. L. and Bittencourt, R. A. (2019). A case study of an integrated programming course based on PBL. In *2019 IEEE Frontiers in Education Conference (FIE)*.
- Savery, J. R. and Duffy, T. M. (1995). Problem-based learning: An instructional model and its constructivist framework. *Educational Technology*, 35(5):31–38.
- Schmidt, H. G., Van Der Molen, H. T., Te Winkel, W. W. R., and Wijnen, W. J. (2011). Constructivist, problem-based learning does work: A meta-analysis of curricular comparisons involving a single medical school. *Educational Psychologist*, 46(3):227–249.
- Silva, M. F. and Oliveira, J. P. (2022). Aprendizagem baseada em problemas como estratégia na extensão universitária: Um estudo de caso em projetos socioculturais. *Revista Brasileira de Extensão Universitária*, 14(2):45–58.
- Souza, A. C. and Mendes, R. L. (2021). Extensão universitária e pbl: Formação em saúde a partir de problemas reais da comunidade. *Cadernos de Educação e Saúde*, 17(3):123–135.
- Souza, S. M. and Bittencourt, R. A. (2019). Motivation e engagement with PBL in an introductory programming course. In *2019 IEEE Frontiers in Education Conference (FIE)*.
- Valente, J. A. (2014). Blended learning e as mudanças no ensino superior: a proposta da sala de aula invertida. *Educar em Revista*, page 79–97. Disponível em: <https://doi.org/10.1590/0104-4060.38645>. Acesso em: 19 de setembro de 2024.