

Do Scratch ao Python: utilizando metodologias ativas no ensino de programação no projeto Meninas Digitais do Acre

Sthefany Ferreira¹, Victoria Cavalcante¹, Alessandreia Oliveira², Catarina Costa¹

¹Centro de Ciências Exatas e Tecnológicas - Universidade Federal do Acre (UFAC)
Rio Branco – AC – Brasil

²Depto de Ciência da Computação - Universidade Federal de Juiz de Fora (UFJF)
Juiz de Fora - MG - Brasil.

sthefany.ferreira@sou.ufac.br, victoria.cavalcante@sou.ufac.br

alessandreia.oliveira@ufjf.br, catarina.costa@ufac.br

Abstract. *Teaching programming to beginners is challenging and can lead to demotivation. This work reports on the experience of courses from the Meninas Digitais do Acre project, a partner of Meninas Digitais [SBC 2025], which employed Problem-Based Learning (PBL), Gamification, Scratch, and Python. The objective was to evaluate the impact of these approaches on student engagement and learning, especially during the transition between programming languages. The analysis revealed that methodological adjustments were essential to mitigate difficulties, resulting in a gradual improvement in performance. The results indicate that active methodologies and a welcoming environment promote participation and permanence. The study reinforces the importance of inclusive initiatives for women in Computing.*

Resumo. *Ensinar programação para iniciantes é desafiador e pode levar à desmotivação. Este trabalho relata a experiência de cursos do projeto Meninas Digitais do Acre, parceiro do programa Meninas Digitais [SBC 2025], utilizando Aprendizagem Baseada em Problemas (ABP), gamificação, Scratch e Python. O objetivo foi avaliar o impacto dessas abordagens no engajamento e aprendizagem das alunas, especialmente na transição entre linguagens. A análise revelou que ajustes metodológicos foram essenciais para mitigar dificuldades, resultando em melhoria gradual no desempenho. Os resultados indicam que metodologias ativas e um ambiente acolhedor favorecem a participação e permanência. O estudo reforça a importância de iniciativas inclusivas para mulheres na Computação.*

1. Introdução

Os cursos superiores de tecnologia no Brasil apontam um público majoritariamente masculino. Segundo Lopes et al. [2023], as mulheres representam 20% do público ingressante e apenas 16% dos graduandos em cursos de tecnologia. As mulheres frequentemente se sentem isoladas, o que as leva a desistir ou até mesmo a não se interessar em ingressar na área. Apesar de já existirem muitas iniciativas voltadas para mulheres com o objetivo de incentivá-las, essas ações ainda não têm sido suficientes para aumentar significativamente o percentual de mulheres na área. Isso reforça a importância de promover iniciativas exclusivas para mulheres.

A programação é uma habilidade importante no contexto atual, sendo sua inclusão no currículo acadêmico fundamental. Além de preparar os alunos para as demandas da área, contribui para o desenvolvimento de competências cognitivas, lógicas, criativas e de resolução de problemas [Bié et al. 2023]. No entanto, aprender programação pode ser desafiador, com dificuldades que vão desde a compreensão da lógica até o entendimento de estruturas de dados complexas [Arimoto and Oliveira 2019].

De acordo com Sousa et al. [2024] , a disciplina de programação tem sido um desafio para iniciantes, levando a altos índices de desistência, reprovação e evasão, além de desmotivação. Por outro lado, métodos tradicionais de ensino, baseados na transmissão de informações e memorização, mostram-se ineficazes para a aprendizagem significativa dos conteúdos [Ferreira et al. 2020].

Para enfrentar esses desafios, identificou-se que metodologias ativas se mostram como boas opções para que os alunos aprendam de forma mais eficaz e significativa [Alves 2023]. A ABP demonstrou reduzir a desistência e reprovação, além de promover maior autonomia aos estudantes [Finger et al. 2021]. Outra abordagem relevante é a gamificação, que tem um impacto positivo no ensino de programação, aumentando o engajamento dos alunos [Calderon et al. 2021].

Além das metodologias ativas, Scratch¹ e Python² também auxiliam no ensino de programação. O Scratch se destaca como uma abordagem lúdica e dinâmica, facilitando a compreensão dos conceitos básicos [Brenda et al. 2024]. Python é considerado uma linguagem de fácil aprendizado devido à sua sintaxe simples e acessível, sendo, assim, adequada para o ensino de programação para iniciantes.

Diante disso, este trabalho relata a experiência de planejamento, execução e avaliação de cursos introdutórios de programação para três turmas do projeto Meninas Digitais do Acre, combinando Scratch, Python, ABP e gamificação. Essas estratégias foram utilizadas para o ensino de conceitos iniciais de programação, visando promover a aprendizagem da lógica de programação de forma lúdica e interativa. Esses cursos foram aplicados na Universidade Federal do Acre (UFAC), em abril, junho e novembro de 2024, no contexto do projeto Meninas Digitais do Acre, projeto este parceiro do Programa Meninas Digitais. Os cursos foram ministrados por uma aluna de graduação do curso de Sistemas de Informação da UFAC, sob a orientação de uma professora do curso de graduação e coordenadora do projeto e o apoio de outras monitoras do projeto.

Como resultado, pode-se verificar que a adaptação das metodologias ao perfil das alunas foi essencial para garantir um aprendizado mais equitativo e eficiente. A introdução de Python representou um desafio, refletido na queda inicial de desempenho em todas as turmas, mas a adoção de abordagens mais flexíveis, como o uso de compiladores alternativos e ajustes no nível das atividades, contribuiu para uma melhora gradual na assimilação dos conceitos. Além disso, o sistema de *ranking* e os *quizzes* incentivaram a participação ativa, promovendo engajamento e motivação ao longo do curso. Esses achados reforçam a importância da personalização no ensino de programação para diferentes públicos, destacando o potencial das metodologias ativas para mitigar dificuldades comuns na aprendizagem inicial de lógica de programação e transição entre linguagens.

¹<https://scratch.mit.edu/>

²<https://www.python.org/psf-landing/>

Este trabalho está organizado como a seguir. A Seção 2 discute trabalhos relacionados, explorando pesquisas anteriores sobre metodologias ativas e o ensino de programação com Scratch e Python. A Seção 3 descreve o planejamento dos cursos, detalhando a definição do público, a estrutura das aulas e a distribuição dos conteúdos. A Seção 4 explica como os cursos foram conduzidos, abordando as dinâmicas aplicadas em cada turma. A Seção 5 analisa o desempenho das alunas, destacando desafios e adaptações feitas ao longo do processo. A Seção 6 apresenta as lições aprendidas, ressaltando os principais achados e insights obtidos. Por fim, a Seção 7 traz as conclusões do estudo e sugestões para trabalhos futuros.

2. Trabalhos Relacionados

A adoção de metodologias ativas tem sido amplamente explorada no ensino de programação para aumentar o engajamento dos estudantes e reduzir taxas de evasão e reaprovação. Diversos estudos têm investigado o impacto de abordagens como ABP, gamificação e desenvolvimento de jogos na introdução à lógica de programação. O estudo de Andrade [2024], realizado no IFPE - Campus Jaboatão dos Guararapes, investigou o impacto de metodologias ativas no ensino de Lógica de Programação. A pesquisa aplicou ABP, aprendizado baseado em equipes e sala de aula invertida para reduzir reaprovação e evasão. O Scratch foi adotado como ferramenta introdutória para facilitar a compreensão dos conceitos básicos de programação, antes da transição para JavaScript. Os resultados mostraram uma melhora no engajamento dos alunos e uma queda significativa na evasão, especialmente no turno noturno.

De forma semelhante, Detoni [2022] apresenta uma intervenção pedagógica baseada em ABP no Instituto Federal do Espírito Santo (IFES) – Campus Santa Teresa. O estudo foi aplicado na disciplina de programação de um curso técnico e teve como objetivo tornar o ensino mais dinâmico e atrativo. O Scratch foi utilizado como ferramenta introdutória para ajudar na assimilação dos conceitos básicos e os alunos participaramativamente da resolução de problemas. Os resultados indicaram maior motivação e participação dos estudantes, evidenciando que a ABP pode ser uma alternativa eficiente ao ensino tradicional.

Já o estudo de Costa et al. [2023], realizado no Instituto Federal do Espírito Santo (IFES) - Campus de Alegre, utilizou a gamificação como suporte para o ensino de lógica de programação. Os alunos participaram de atividades com o software educacional CodeCombat³, que combina a ABP com desafios de programação em um jogo estilo masmorra. A pesquisa foi aplicada em cursos técnicos e no ensino superior, utilizando Python e JavaScript. Os resultados indicaram alto nível de motivação e engajamento dos estudantes, além de um bom equilíbrio entre teoria e prática no aprendizado da lógica de programação.

Costa et al. [2024] apresentam um relato de experiência sobre o ensino de programação utilizando o MIT App Inventor para alunos do ensino médio de uma escola pública. O estudo, conduzido pelo grupo GET Engenharia Computacional da UFJF, teve como objetivo estimular o desenvolvimento do Pensamento Computacional por meio de oficinas práticas. A metodologia combinou aulas teóricas e práticas, explorando conceitos de lógica de programação e criação de aplicativos. Os resultados indicam que o uso

³<https://br.codecombat.com/about>

de ferramentas visuais facilitou a compreensão dos conceitos, promovendo maior engajamento dos alunos. O trabalho destaca a importância de abordagens interativas no ensino de programação e sugere a replicabilidade da experiência em outras instituições.

Honda et al. [2020] investigaram o uso da Aprendizagem Baseada no Desenvolvimento de Jogos (Game Development-Based Learning – GDBL) no ensino de Computação. O estudo utiliza o método qualitativo de narrativa de vida para relatar a experiência de um estudante de Licenciatura em Computação na criação de um jogo educacional sobre autômatos finitos determinísticos. O processo envolveu concepção, prototipação, desenvolvimento e testes, promovendo aprendizado prático em estrutura de dados, programação orientada a objetos e design instrucional. Os resultados indicam que a criação de jogos educacionais favorece o desenvolvimento de competências computacionais e pedagógicas, destacando a abordagem GDBL como uma estratégia motivadora e eficaz no ensino de Computação.

Alves [2023] conduziu um estudo no Instituto Federal de Pernambuco (IFPE) - Campus Jaboatão dos Guararapes, no qual analisou o impacto da aplicação de metodologias ativas na disciplina de Lógica de Programação. A pesquisa combinou ABP, Aprendizado Baseado em Equipes e sala de aula invertida, buscando tornar o ensino mais interativo e envolvente. Inicialmente, os alunos utilizaram o Scratch para se familiarizarem com conceitos fundamentais da programação, antes de avançarem para linguagens textuais. Os resultados evidenciaram uma redução na evasão e reprovação, além do aumento da participação dos estudantes. A introdução das metodologias ativas proporcionou um ambiente de aprendizado mais dinâmico, onde os alunos puderam aplicar os conceitos de forma prática, favorecendo a compreensão e retenção do conteúdo.

3. Planejamento dos Cursos

Esta seção apresenta a organização dos cursos, contendo a definição do público, o planejamento das aulas utilizado para as três turmas e a distribuição do conteúdo. Mais informações sobre os conteúdos trabalhados durante as aulas e atividades realizadas podem ser acessados⁴ e aplicados em outras iniciativas.

3.1. Definição do Público

Os cursos foram direcionados exclusivamente ao público feminino, visando estimular a participação de mulheres na Computação. As iniciativas descritas a seguir ocorreram no âmbito do projeto de extensão e pesquisa Meninas Digitais do Acre, criado em 2024. As turmas foram compostas por mulheres com diferentes perfis, incluindo alunas do ensino fundamental e médio, ingressantes no curso de Sistemas de Informação da UFAC, estudantes de outras graduações e mulheres que já concluíram o ensino médio, mas ainda não ingressaram no ensino superior.

3.2. Planejamento das Aulas

Com o público definido, as aulas foram planejadas em reuniões com a orientadora para definir metodologias, abordagens e organização do conteúdo. Foram estabelecidas como metodologias aplicadas e avaliadas: a ABP e a gamificação. Além disso, o Scratch foi adotado como uma primeira linguagem de contato mais visual, facilitando a compreensão

⁴<https://doi.org/10.5281/zenodo.15106890>

da lógica de programação, seguido pelo ensino da codificação na linguagem Python. Essas metodologias, juntamente com o uso do Scratch e do Python, foram selecionadas para proporcionar às estudantes uma experiência de aprendizado diferenciada, distanciando-se dos métodos tradicionais de ensino.

A organização de como o conteúdo e as abordagens seriam aplicados foi uma etapa fundamental do planejamento. Para incorporar a metodologia ABP em todas as aulas, as alunas realizaram atividades práticas que permitiram aplicar o que foi ensinado. Além disso, foi implementado um *ranking* que gerava uma classificação das três primeiras posições de acordo com as pontuações obtidas pelo acerto nas atividades em cada aula, promovendo o uso da gamificação. Ao final do curso, as três primeiras colocadas receberam um brinde.

A Tabela 1 apresenta a distribuição do conteúdo ao longo dos dias dos cursos. O primeiro dia foi dedicado à introdução de conceitos básicos de programação, permitindo que as alunas se familiarizassem com o tema. No segundo dia, a plataforma Scratch foi apresentada, com a explicação de como utilizá-la e a introdução aos blocos básicos. No terceiro dia, foram introduzidos conceitos de lógica de programação utilizando os blocos do Scratch, com a continuidade desses conceitos no quarto dia, onde as alunas aprenderam sobre operadores lógicos e variáveis.

Após o conteúdo de Scratch ser concluído, foi introduzida a linguagem de programação Python, iniciando no quinto dia do curso com a apresentação do Python e seus comandos básicos. No sexto dia, a aula abordou o uso de condicionais, e o curso foi encerrado no sétimo e oitavo dias com o estudo de estruturas de controle. A organização desse conteúdo foi planejada para que as alunas aprendessem de forma linear e contínua, permitindo que evoluíssem a cada etapa do aprendizado.

Tabela 1. Conteúdo das Aulas

Dia	Conteúdo	Objetivo
1	Introdução à Programação	Apresentar o que é programação com exemplos e explicar os conceitos básicos como variáveis, tipos de dados e estruturas de controle.
2	Introdução ao Scratch	Apresentar o Scratch, ensinando como ele funciona e explicando o funcionamento geral dos blocos de código.
3	Aprendendo Scratch	Ensinar o Scratch na prática, focando nos blocos de código de eventos e controle.
4	Aprendendo Scratch	Ensinar o Scratch na prática, focando nos blocos de código de sensores, operadores e variáveis.
5	Apresentação e comandos básicos do Python	Apresentar a linguagem e ensinar conceitos básicos como variáveis, tipos de dados, entrada e saída de dados, operações matemáticas e operadores de atribuição.
6	Estruturas de Controle	Ensinar as estruturas de controle ‘if’, ‘else’ e ‘elif’.
7	Estruturas de Controle	Ensinar a estrutura de controle ‘for’.
8	Estruturas de Controle	Ensinar a estrutura de controle ‘while’.

4. Condução dos Cursos

Nesta seção, são apresentados mais detalhes da aplicação dos cursos, levando em consideração que as metodologias ativas ABP e gamificação foram aplicadas em conjunto, com ajustes conforme o perfil das alunas de cada turma. Durante as aulas, as participantes realizavam atividades práticas e *quizzes* para fixar o conteúdo ministrado. Além disso, um sistema de *ranking* foi utilizado para estimular o engajamento, onde a pontuação era calculada e acumulada ao longo do curso e as três primeiras colocadas recebiam brindes ao final.

Cada curso teve 20 horas de duração e ocorreu no laboratório N.A.V.E Lab, na UFAC, com 26 alunas concluintes. A Tabela 2 apresenta a quantidade de alunas em cada turma, o período de realização das atividades e o público participante. A primeira turma foi composta por alunas do 1º, 2º e 3º ano do Ensino Médio. A segunda turma era formada exclusivamente por calouras do curso de Sistemas de Informação da UFAC. A terceira turma teve um perfil multidisciplinar, com a participação de meninas do Ensino Fundamental, Médio e da graduação, oriundas de diversos cursos.

Tabela 2. Resumo das Turmas

Turma	Qtde	Período	Público
1	7	09/04/2024 à 02/05/2024	Ensino Médio
2	8	04/06/2024 à 27/06/2024	Calouras de SI
3	11	05/11/2024 à 28/11/2024	Fundamental, médio e graduação
Total de alunas concluintes: 26			

Na Turma 1, cada aula tinha atividades práticas ou questionários que valiam, no máximo, 25 pontos, totalizando até 200 pontos ao final do curso. No primeiro dia, um *quiz* foi aplicado para reforçar os conceitos introdutórios de programação. Nos dias seguintes, as atividades tiveram seu foco na prática com Scratch, onde as alunas seguiam instruções para desenvolver pequenos programas, aplicando blocos de controle, eventos e variáveis. A partir do quinto dia, iniciou-se a transição para Python, com atividades que envolviam a escrita de código. Inicialmente, foi utilizada a plataforma Beecrowd⁵, mas, devido às dificuldades das alunas, a abordagem foi ajustada para o compilador online gdb⁶, permitindo maior flexibilidade na resolução dos exercícios. No último dia, um *quiz* final avaliou o conhecimento adquirido sobre Python.

A Turma 2 seguiu a mesma estrutura de ensino da Turma 1, porém, as atividades foram ajustadas para o perfil das alunas, que já tinham alguma familiaridade com tecnologia. A pontuação variava conforme a quantidade de questões em cada exercício, sem um limite fixo diário. Nos primeiros dias, as alunas responderam *quizzes* sobre Scratch, consolidando os conceitos antes da prática. Diferente da Turma 1, essa teve mais autonomia nas atividades, tornando possível manter a plataforma Beecrowd para a resolução dos desafios de Python. Ao longo dos últimos dias, foram propostos exercícios práticos que exigiam a interpretação dos problemas e a escrita de código. No final, a realização de um *quiz* avaliou o aprendizado.

⁵<https://beecrowd.com/pt/sobre-nos/>

⁶<https://www.onlinegdb.com/>

A Turma 3 foi a mais diversificada, composta por alunas de diferentes níveis de conhecimento. Nessa turma, a pontuação foi baseada apenas em *quizzes* diários, sem atividades práticas avaliativas. As práticas com Scratch e Python foram realizadas, mas sem atribuição de pontuação no *ranking* final, apenas para exercitar os conteúdos aprendidos. Nos primeiros quatro dias, os *quizzes* abordaram os fundamentos da programação e os blocos do Scratch. A partir do quinto dia, iniciou-se Python, incluindo atividades práticas complementares que não valiam pontos. A avaliação final também foi realizada por meio de um *quiz*, cobrindo todo o conteúdo ensinado.

As Figuras 1, 2 e 3 apresentam o registro fotográfico do último dia de aula de cada uma das três turmas.



Figura 1. Turma 1



Figura 2. Turma 2



Figura 3. Turma 3

5. Análise de Desempenho das Alunas

Apesar de o conteúdo ter sido o mesmo para todas as turmas, a abordagem foi ajustada conforme o nível que cada turma apresentou. A Turma 1 realizou mais atividades práticas guiadas. A Turma 2 manteve desafios mais estruturados com Beecrowd. A Turma 3 utilizou majoritariamente *quizzes* para obter a pontuação no *ranking*, mas também realizou atividades práticas sem pontuação, apenas para exercitar os conteúdos aprendidos. Essas adaptações garantiram que todas as alunas conseguissem acompanhar as atividades de forma equitativa e eficiente.

Na Turma 1, no primeiro dia, 100% das alunas obtiveram uma pontuação acima de 12,5 pontos (50% da pontuação máxima de 25). Esse resultado indica que a maioria da turma teve um desempenho satisfatório nas atividades introdutórias. No segundo e terceiro dias, o desempenho permaneceu alto, com todas as alunas atingindo mais de 12,5 pontos. Já no quarto dia, houve uma leve queda: apenas 66,67% superaram essa pontuação, o que pode indicar uma dificuldade pontual nos conceitos abordados. A transição para Python no quinto dia representou um desafio mais significativo: apenas 42,86% das alunas atingiram ao menos 12,5 pontos, sugerindo a necessidade de uma abordagem mais gradual para a nova linguagem. No sexto e sétimo dias, observou-se uma recuperação no desempenho, com 100% das alunas pontuando acima de 12,5 pontos. Contudo, no oitavo e último dia, novamente apenas 42,86% atingiram essa meta, o que sugere que o conteúdo final ou o formato da atividade exigiu maior retenção ou domínio dos conceitos trabalhados ao longo do curso.

Na Turma 2, no primeiro dia, 100% das alunas obtiveram uma pontuação acima de 12,5 pontos, demonstrando um ótimo desempenho inicial. No segundo dia, a pontuação máxima era 10 pontos, e 33,33% das alunas pontuaram acima de 5 pontos, sugerindo uma leve dificuldade inicial com as atividades. No terceiro dia, com pontuação máxima de

15 pontos, 55,56% das alunas alcançaram mais de 7,5 pontos, demonstrando uma melhoria no desempenho. No quarto dia, com pontuação máxima de 10 pontos, 44,44% das alunas obtiveram mais de 5 pontos, indicando que estavam assimilando os conceitos ensinados. A partir do quinto dia, com a introdução da linguagem Python e uma pontuação máxima de 5 pontos, 55,56% das alunas pontuaram acima de 2,5 pontos. No sexto dia, com pontuação máxima de 3 pontos, 33,33% das alunas ultrapassaram 1,5 pontos, demonstrando uma adaptação inicial ao Python. No sétimo dia, com pontuação máxima de 5 pontos, 55,56% das alunas pontuaram acima de 2,5 pontos, sugerindo um avanço no aprendizado. No oitavo e último dia, com pontuação máxima de 30 pontos, 66,67% das alunas pontuaram acima de 15 pontos, mostrando que a maioria conseguiu consolidar os conhecimentos adquiridos ao longo do curso.

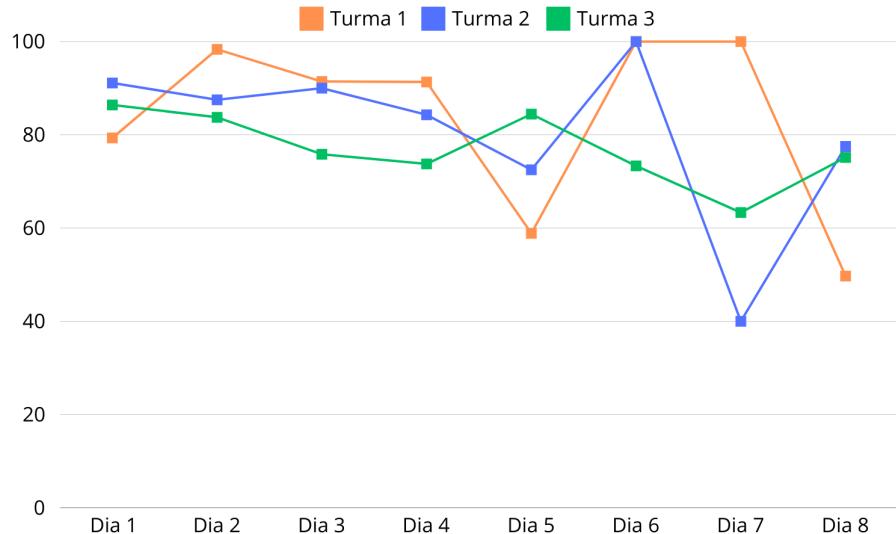
Na Turma 3, no primeiro dia, 72,73% das alunas obtiveram uma pontuação acima de 12,5 pontos, indicando um bom desempenho inicial. No segundo dia, com pontuação máxima de 10 pontos, 27,27% das alunas conseguiram mais de 5 pontos, indicando um desafio maior nessa fase. No terceiro dia, com pontuação máxima de 15 pontos, 36,36% das alunas alcançaram mais de 7,5 pontos, mostrando um desempenho abaixo do esperado. No quarto dia, com pontuação máxima de 10 pontos, 36,36% das alunas pontuaram acima de 5 pontos, sugerindo uma assimilação gradual dos conceitos. No quinto dia, com pontuação máxima de 10 pontos, 45,45% das alunas ultrapassaram 5 pontos, demonstrando um progresso no aprendizado de Python. No sexto dia, com pontuação máxima de 10 pontos, 36,36% das alunas pontuaram acima de 5 pontos, reforçando que algumas alunas ainda enfrentavam dificuldades. No sétimo dia, com pontuação máxima de 10 pontos, 36,36% das alunas ultrapassaram 5 pontos, consolidando o aprendizado. No oitavo e último dia, com pontuação máxima de 30 pontos, 54,55% das alunas pontuaram acima de 15 pontos, demonstrando que uma parte significativa conseguiu consolidar os conhecimentos adquiridos ao longo do curso.

Essa análise baseada em 50% da pontuação máxima permitiu uma visão mais equilibrada do desempenho das alunas ao longo das atividades, destacando os momentos de maior dificuldade e adaptação ao conteúdo ensinado. Com base na aplicação das metodologias ABP e gamificação, observou-se que, de forma geral, as alunas tiveram um bom desempenho e engajamento ao longo das atividades. O uso do *ranking* motivou a participação ativa, incentivando as alunas a buscarem melhores resultados a cada dia.

Comparando as três turmas, nota-se algumas semelhanças e diferenças significativas. Em todas elas, o primeiro dia de atividades foi marcado por um bom desempenho, com as alunas conseguindo acompanhar bem os conteúdos introdutórios. No entanto, à medida que as atividades avançaram e os desafios se tornaram mais complexos. Especialmente com a introdução de Python, foi possível identificar um impacto na pontuação das participantes. Essa variação pode ser observada na Figura 4, onde é possível notar um bom desempenho geral na média de acertos por turma nos primeiros dias. Porém, no quinto dia, quando se iniciaram as aulas de Python, houve uma queda na pontuação das turmas 1 e 2. Já na turma 3, observou-se um aumento em relação aos dias 3 e 4. Após o quinto dia, ocorreram variações entre as turmas quanto à média de acertos por dia.

Apesar de todas as turmas serem compostas exclusivamente por mulheres, elas apresentavam níveis de conhecimento e faixas etárias diferentes. Por esse motivo, as atividades foram adaptadas para se adequar ao nível de cada turma, garantindo um apren-

Figura 4. Percentual médio de acertos por dia em cada turma



dizado linear para todas. Inicialmente, as atividades práticas de Python seriam realizadas na plataforma Beecrowd. Contudo, a Turma 1 enfrentou dificuldades significativas, levando à adoção de um compilador online mais acessível. Essa adaptação permitiu maior flexibilidade e facilitou a aprendizagem.

Já na Turma 2, como todas as alunas já estavam direcionadas para a área de tecnologia, matriculadas no curso de Sistemas de Informação e algumas egressas de curso técnico integrado ao ensino médio na área de informática, foi possível manter todas as atividades práticas de Python na plataforma Beecrowd. Elas não tiveram dificuldades com a plataforma e, além disso, como já estavam prestes a iniciar o curso de Sistemas de Informação, o uso dessa ferramenta foi considerado relevante, pois fariam uso dela ao longo da graduação.

Por fim, na Turma 3, com diferentes perfis de participantes, o nível de conhecimento das alunas era bastante diversificado. Com base na experiência da Turma 1, percebeu-se que algumas conseguiriam utilizar a plataforma Beecrowd, enquanto outras teriam dificuldades. Diante disso, para garantir que todas pudessem acompanhar, aprender e resolver as atividades, foi adotada uma abordagem alternativa. No entanto, em relação aos questionários e às atividades no Scratch, nenhuma das turmas apresentou dificuldades significativas.

6. Lições aprendidas

Durante a implementação dos cursos, diversas lições foram aprendidas ao longo do processo. A adaptação da metodologia de ensino ao perfil das turmas revelou-se essencial, uma vez que as participantes apresentavam diferentes níveis de conhecimento, e algumas enfrentavam dificuldades com recursos básicos do computador. Diante desse cenário, foi necessário ajustar as atividades de acordo com o nível geral da turma, garantindo que todas pudessem acompanhar o conteúdo. Essa experiência demonstrou a importância da flexibilização no ensino, exigindo uma observação atenta das dificuldades apresentadas para que fossem feitas as adaptações necessárias visando o aprimoramento do aprendizado.

Mesmo com esses ajustes, algumas alunas ainda encontraram desafios para acompanhar os conteúdos, especialmente nas aulas envolvendo a linguagem Python. Essa diferença no ritmo de aprendizagem reforçou a necessidade de oferecer suporte individualizado sempre que possível, permitindo que todas consolidassem o aprendizado de forma mais eficaz.

Além das adaptações e dos desafios enfrentados, a experiência como um todo proporcionou crescimento e resultados positivos. O *feedback* das alunas indicou que os recursos utilizados no curso foram adequados, demonstrando a importância das escolhas metodológicas adotadas. Embora o estudo tenha se concentrado nos resultados quantitativos, as análises qualitativas, com base nas opiniões das participantes, evidenciaram um alto nível de concordância em relação à definição das metodologias, linguagens e abordagens aplicadas ao longo das atividades.

7. Conclusões

Este estudo apresentou a experiência do projeto Meninas Digitais do Acre na implementação de cursos introdutórios de programação utilizando metodologias ativas, como ABP e gamificação, integradas ao ensino de Scratch e Python. O objetivo foi proporcionar uma experiência de aprendizado dinâmica e interativa para alunas com diferentes perfis, buscando minimizar desafios comuns ao ensino de programação e promover um ambiente inclusivo e acolhedor.

A análise dos resultados evidenciou que a adaptação das metodologias ao perfil das turmas foi essencial para o aprendizado. A introdução de Python representou um desafio significativo, resultando em uma queda inicial no desempenho das alunas. No entanto, a implementação de ajustes, como o uso de compiladores alternativos e atividades adaptadas, contribuiu para uma recuperação gradual na assimilação dos conceitos. Além disso, a gamificação, por meio de *quizzes* e um sistema de *ranking*, mostrou-se uma estratégia eficaz para estimular o engajamento e a motivação das participantes nas aulas.

Os resultados deste estudo reforçam a importância da personalização no ensino de programação para iniciantes e da criação de iniciativas voltadas para a inclusão de mulheres na computação. A experiência relatada demonstrou que um ambiente de aprendizado estruturado e adaptável pode contribuir para reduzir dificuldades na transição entre linguagens e aumentar a retenção das alunas.

Como trabalhos futuros, sugere-se a exploração de novas metodologias ativas combinadas ao ensino de Scratch e Python, investigando seu impacto no aprendizado e na motivação das alunas. Além disso, futuras edições dos cursos podem incluir a realização de competições de programação, como maratonas e desafios práticos, para reforçar os conceitos e estimular a autonomia das participantes no processo de aprendizado.

Referências

- Alves, H. D. (2023). Análise da aplicação de metodologias ativas em disciplina de lógica de programação. In *Encontro Nacional de Computação dos Institutos Federais (EN-CompIF)*, pages 85–92. SBC.
- Andrade, H. D. (2023). Aplicação de metodologias ativas em disciplina de lógica de programação no ifpe jaboatão dos guararapes. *Revista de Sistemas e Computação-RSC*, 13(3).

- Arimoto, M. and Oliveira, W. (2019). Dificuldades no processo de aprendizagem de programação de computadores: um survey com estudantes de cursos da área de computação. In *Anais do XXVII Workshop sobre Educação em Computação*, pages 244–254. SBC.
- Bié, E. P., Souto, E., Braga, D., Oliveira, E., and Carvalho, L. (2023). Ensino de programação para alunos nos anos escolares entre ensino fundamental ii e ensino médio: Um mapeamento sistemático. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 414–427. SBC.
- Brenda, F., Venâncio, C., Pedrosa, C., Miguel, J., Martins, W., Honda, F., Macena, J., Pires, F., and Pessoa, M. (2024). Um curso de programação para calouros com scratch: um relato de licenciandos em computação. In *Workshop sobre Educação em Computação (WEI)*, pages 251–262. SBC.
- Calderon, I., Silva, W., and Feitosa, E. (2021). Um mapeamento sistemático da literatura sobre o uso de metodologias ativas durante o ensino de programação no brasil. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 1152–1161.
- Costa, C. F., Ferreira, O. S., Ditz, Á. J. M., Tosta, C. S. V., da Silva, C. A. S., Simão, F. P., and Silveira, P. D. N. (2023). Gamificação como auxílio ao desenvolvimento da lógica de programação para alunos de cursos de informática. *Cuadernos de Educación y Desarrollo*, 15(4):3756–3773.
- Costa, H. H. B., Maurício, J. S. S., de Filippo Cavalini, A., de Souza, G. H., Mattos, G., de Faria Rocha, D. L., Caçador, J. Z., de Souza, R. G. V., Nazareth, L. G. F., Menezes, M. I. R., et al. (2024). Ensino de conceitos básicos de programação usando mit app inventor para alunos de escolas públicas: um relato de experiência. In *Workshop sobre Educação em Computação (WEI)*, pages 297–306. SBC.
- Detoni, A. A. (2022). Intervenção pedagógica apoiada em aprendizagem baseada em problemas: estudo de caso em disciplina inicial de programação para curso técnico de nível médio da área de informática.
- Ferreira, D. H. L., Branchi, B. A., Sugahara, C. R., et al. (2020). Processo de ensino e aprendizagem no contexto das aulas e atividades remotas no ensino superior em tempo da pandemia covid-19. *Revista práxis*, 12(1sup).
- Finger, A. F., da Silva, J. P. S., and Ecar, M. (2021). Utilizando aprendizado baseado em problemas para o ensino de paradigmas de programação. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 135–144. SBC.
- Honda, F., Pires, F., Pessoa, M., and Oliveira, E. (2020). Lições aprendidas em computação através da criação de um jogo educacional: entre autômatos e design de aprendizagem. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pages 1753–1762, Porto Alegre, RS, Brasil. SBC.
- Lopes, R., Maciel, B., Soares, D., Figueiredo, L., and Carvalho, M. (2023). Análise e reflexões sobre a diferença de gênero na computação: podemos fazer mais? In *Anais do XVII Women in Information Technology*, pages 68–79, Porto Alegre, RS, Brasil. SBC.
- SBC (2025). Meninas Digitais. Acessado em: 26 mar. 2025.

Sousa, R. R., Leite, F. T., de Oliveira Guimarães, Á., and de Oliveira, A. R. (2020). Pré-algoritmos–ações de apoio à melhoria do ensino de graduação. *Brazilian Journal of Development*, 6(3):12625–12635.