REALIZAÇÃO ABENGE Associação Brasileira de Educação em Engenharia
COBENGE 2026
14 A 18 DE SETEMBRO DE 2026
BELO HORIZONTE -MG
ORGANIZAÇÃO PUC ENGENHARIAS PUC Minas

PRÁTICAS DE PROGRAMAÇÃO E MODELAGEM NA FORMAÇÃO INICIAL DE ALUNOS DO CURSO DE ENGENHARIA COMPUTACIONAL: UMA APLICAÇÃO DE METODOLOGIAS ATIVAS

DOI: 10.37702/2175-957X.COBENGE.2026.0331

Autores: *ROBERT GONÇALVES VIEIRA DE SOUZA, JOÃO ARTHUR VIEIRA, LARA OLIVEIRA ESTEVES, JOVENTINO DE OLIVEIRA CAMPOS, HEITOR COELHO COSTA, LEONARDO BECK CARLOS*

Resumo:

Nos cursos de graduação, especialmente nos períodos iniciais, é comum que os ingressantes apresentem dificuldades para compreender o escopo prático de sua futura atuação profissional, tornando as disciplinas introdutórias essenciais para o nivelamento e o acolhimento acadêmico. Na Engenharia Computacional, esse primeiro contato ocorre por meio do estudo de modelagem computacional, uma área abstrata que costuma gerar resistência inicial. Para mitigar essa dificuldade, o presente artigo relata a intervenção de um grupo de educação tutorial que, em parceria com a docência, aplicou os princípios da Aprendizagem Baseada em Projetos e da mentoria de pares na disciplina. A metodologia consistiu na estruturação de aulas práticas de Python, LaTeX e comunicação, culminando no desenvolvimento de um projeto final de simulação visual utilizando autômatos celulares. A análise dos dados referentes às edições entre 2024 e 2026 demonstrou a eficácia da abordagem, registrando um índice de aprovação discente de 87,6% quanto à contribuição da iniciativa para o aprendizado prático. O processo de avaliação contínua também evidenciou a necessidade de readequação no ritmo das aulas de programação, gerando um plano de ação focado em reduzir a sobrecarga cognitiva e aprimorar as futuras edições do curso.

Palavras-chave: Engenharia Computacional; Programa de Educação Tutorial; Modelagem Computacional; Python; Educação na Engenharia; Autômatos Celulares

PRÁTICAS DE PROGRAMAÇÃO E MODELAGEM NA FORMAÇÃO INICIAL DE ALUNOS DO CURSO DE ENGENHARIA COMPUTACIONAL: UMA APLICAÇÃO DE METODOLOGIAS ATIVAS

1 INTRODUÇÃO

O grupo de educação tutorial da Universidade Federal de Juiz de Fora (UFJF) tem como um de seus principais compromissos organizar projetos de ensino para alunos de graduação. A formação do curso de Engenharia Computacional visa preparar profissionais capazes de formular, analisar e implementar modelos matemáticos e computacionais para a solução de problemas científicos e de engenharia, por meio de um currículo interdisciplinar e flexível com definido por (Lobosco et al., 2018). Nesse contexto, as iniciativas desenvolvidas pelo grupo se inserem como um importante complemento às estratégias institucionais de acolhimento e orientação dos ingressantes.

Durante as fases iniciais do ensino superior, na maior parte dos casos, o propósito e os objetivos da área não são bem compreendidos pelos calouros. Por isso, é comum a oferta de disciplinas de caráter introdutório, que visam apresentar ao estudante as expectativas e realidades da profissão escolhida, inserindo-o no contexto acadêmico, o que permite enriquecer a sua experiência (Marinho et. al., 2018). No caso específico da Engenharia Computacional, existe a disciplina de Introdução à Engenharia Computacional, que cumpre esse papel. Ela atua em uma temática, a modelagem computacional, ainda pouco difundida fora da universidade e que costuma gerar certa resistência inicial na assimilação de seus conceitos.

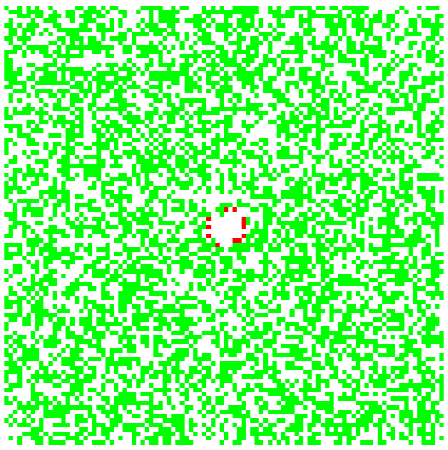
Nessa perspectiva, notou-se uma grande oportunidade de troca de experiências entre os membros do Grupo de Educação Tutorial (GET) de Engenharia Computacional, os discentes e os professores da disciplina, já que é função do grupo auxiliar e engajar os alunos em sua formação, cumprindo seu papel no ensino. Assim, em conjunto com o responsável pela matéria, foi elaborado um plano de curso que inclui cinco aulas, atividades em sala e extraclasse, além de um trabalho final.

Esses encontros são organizados em etapas de capacitação e prática, de forma que em cada aula seja aprendida uma habilidade necessária para a realização do projeto e essencial para um engenheiro computacional. O trabalho proposto consiste na aplicação de um método de modelagem básico. Os alunos devem se organizar em grupos, para estimular o trabalho em equipe (Bender, 2014), e programar em Python o método aplicado a algum fenômeno empírico, como incêndio florestal ou propagação de bactérias. Além disso, devem entregar um relatório escrito em LaTeX e realizar uma apresentação sobre o projeto ao fim da disciplina.

Nas últimas três edições, o método abordado foi o dos autômatos celulares, considerado lúdico e visualmente bem representativo do que é a modelagem. Em síntese, um autómato celular é um modelo matemático e computacional discreto composto por uma malha de células que evoluem ao longo do tempo segundo regras de transição específicas (Shiflet, 2014). O estado de cada célula numa determinada iteração é definido com base no seu estado anterior e no estado das suas células vizinhas. Pela sua capacidade de simular o surgimento de comportamentos macroscópicos complexos a partir de regras locais simples, o método revela-se uma excelente ferramenta pedagógica para introduzir os ingressantes aos fundamentos da simulação de sistemas dinâmicos (Oliveira et al., 2021). A aplicação visual deste método pode ser observada na Figura 1, que ilustra o estado inicial de uma

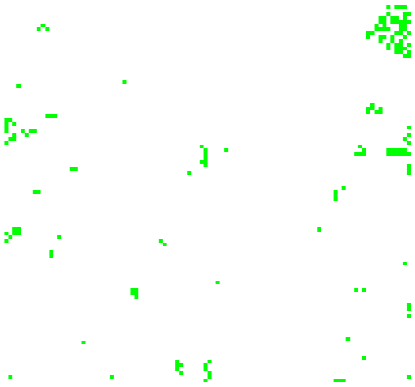
simulação básica de incêndio florestal, e na Figura 2, que apresenta o estado final da malha após a propagação do fogo.

Figura 1 – Estado inicial da implementação de autômatos celulares em incêndio florestal



Fonte: Os autores (2026)

Figura 2 – Estado final da implementação de autômatos celulares em incêndio florestal



Fonte: Os autores (2026)

Embora a colaboração do grupo nessa disciplina ocorra há cinco anos, o presente artigo tem como objetivo relatar e analisar a experiência e os resultados obtidos com essa metodologia nas três últimas edições, compreendendo o período letivo entre 2024 e o momento atual de 2026.

2 METODOLOGIA

Neste tópico, detalha-se a construção e a aplicação da metodologia utilizada pelo GET Engenharia Computacional nas aulas e atividades da disciplina.

2.1 Plano de Curso

O ponto central do planejamento estruturado pelo GET é o desenvolvimento contínuo do trabalho final ao longo dos encontros. Para isso, os conceitos teóricos e práticos são apresentados seguindo exatamente a ordem cronológica exigida para a confecção do projeto. Essa estruturação guiada alinha-se aos princípios da Aprendizagem Baseada em Projetos (ABP) defendidos por Bacich e Moran (2018), que ressaltam a importância de fragmentar desafios complexos para que a teoria ganhe sentido imediato na prática do aluno. Especialmente no ensino de engenharia, como destacam Barbosa e Moura (2014), organizar a construção do conhecimento em etapas práticas sequenciais reduz a sobrecarga cognitiva do ingressante e garante que o estudante assuma um papel ativo, compreendendo o propósito de cada ferramenta metodológica apresentada antes de avançar para a próxima fase do projeto.

2.2 Organização das Aulas

As aulas são estruturadas sob um formato padronizado, combinando exposições teóricas, por meio de slides, com atividades práticas. Como os encontros são ministrados em laboratórios de informática, garante-se que cada discente tenha acesso individual a um computador para a execução das tarefas em tempo real. Além disso, a fim de assegurar um suporte personalizado e atender às diversas necessidades de aprendizagem da turma, estabeleceu-se a diretriz de que ao menos três membros do GET estejam sempre presentes em sala. Essa prática de mentoria de pares reduz a sobrecarga cognitiva e é cientificamente comprovada como uma estratégia eficaz para melhorar o rendimento e sanar dificuldades individuais no ensino de programação em cursos de engenharia (Figuerêdo et al., 2021).

Na fase inicial, é feita uma introdução à programação em Python. Como a disciplina ocorre no segundo período, os alunos já possuem conhecimentos prévios em programação (C++), pensamento computacional e matemática (cálculo e geometria analítica). Isso permite introduzir o Python, uma linguagem de alto nível e rápido aprendizado (Van Rossum et al., 2009) sem grande resistência. Esses conhecimentos prévios são, então, utilizados para criar os exemplos práticos da disciplina, unindo programação à resolução de problemas matemáticos já familiares para introduzir os conceitos de modelagem computacional.

A introdução ao Python é dividida em duas aulas focadas em funções básicas e bibliotecas. A primeira inicia-se apresentando as vantagens e diferenças do Python em relação a outras linguagens base, visando engajar os alunos. Em seguida, abordam-se os ambientes de desenvolvimento integrado, com foco na justificativa de uso do Google Colaboratory. Após essa etapa, exploram-se os conceitos da linguagem, como variáveis e estruturas condicionais, sempre comparando as particularidades em relação aos conhecimentos prévios da turma. Ao fim de cada conceito, realizam-se exercícios de fixação. A segunda aula aborda as funções essenciais das bibliotecas NumPy, SymPy, Pandas e Matplotlib. Para contornar a resistência dos alunos em compreender a relevância desse tema, foram inseridas duas atividades avaliativas de baixa complexidade na aula para fixação prática.

Encerrada a etapa de introdução, ministra-se uma aula focada no detalhamento do projeto final. Neste momento, ocorre a formação das equipes, a apresentação das diretrizes da atividade principal e o início de sua implementação prática. A exposição dedicada aos autômatos celulares oferece uma visão abrangente sobre a origem e as aplicações do método. Em seguida, introduzem-se os conceitos técnicos fundamentais para a modelagem: a definição da malha espacial, o conjunto finito de estados possíveis para cada célula, as configurações de vizinhança, como os modelos de Moore ou von Neumann, e as regras de transição locais. Essa fundamentação garante que os alunos compreendam a lógica

matemática e dinâmica do sistema antes de transcrevê-lo para a linguagem de programação.

Nesse momento, parte-se para as aulas voltadas aos conceitos de soft skills, habilidades comportamentais e de comunicação que são fundamentais para um engenheiro computacional e, principalmente, para um aluno de graduação (Rocha et al., 2021). A etapa se inicia com uma aula de conceitos básicos de LaTeX utilizando a ferramenta colaborativa Overleaf. Sendo o LaTeX um sistema de composição tipográfica de alta qualidade amplamente utilizado em trabalhos acadêmicos de engenharia e computação (Lamport, 1994), o GET Engenharia Computacional julga fundamental que o estudante tenha o domínio básico desta ferramenta. Esse conhecimento será avaliado de forma prática por meio de um relatório obrigatório, escrito em LaTeX, sobre a implementação do código desenvolvido pelo grupo.

Encerrando as aulas conteudistas, ministra-se uma aula de boas práticas em apresentações, que segue as diretrizes de defesas acadêmicas definidas por Severino (2016). O final da matéria consiste, então, na entrega do código do autômato celular sobre o tema escolhido, do relatório do trabalho e de uma apresentação oral, que visa trabalhar ainda mais as soft skills dos alunos. A depender do calendário da disciplina, essa apresentação pode ser feita presencialmente ou por meio de vídeos enviados pelas equipes.

2.3 Recursos Didáticos e Tecnológicos

Como as aulas sobre conceitos matemáticos e computacionais exigem explicações apoiadas em representações gráficas, todas as exposições teóricas utilizam slides como recurso pedagógico (Bartsch, 2003). Esse material é desenvolvido pela própria equipe utilizando a ferramenta Figma, o que garante uma identidade visual padronizada a partir de reuniões prévias de planejamento. Além disso, os encontros ocorrem em laboratórios de informática, assegurando aos alunos acesso individual aos computadores. Para a execução prática das atividades, o ecossistema tecnológico da disciplina apoia-se em plataformas de funcionamento em nuvem: adota-se o Google Colaboratory para a programação e o Overleaf para a formatação dos relatórios (Tock, 2019). O emprego dessas ferramentas viabiliza a continuidade do projeto em qualquer máquina e facilita a edição simultânea, estimulando a dinâmica de trabalho em equipe exigida na disciplina.

2.4 Avaliação

O GET busca, a cada edição, adaptar e aprimorar o projeto conforme as lacunas e dificuldades percebidas. Para quantificar as abordagens a serem modificadas, aplicou-se, ao fim da disciplina, um formulário eletrônico de avaliação. O instrumento foi estruturado com base em itens mensurados por meio de uma escala Likert de cinco pontos, permitindo a análise da percepção dos alunos quanto aos seguintes aspectos:

- Como você avalia a contribuição do GET Engenharia Computacional para a disciplina?
- Os conteúdos abordados contribuíram para o seu aprendizado?
- A forma de apresentação dos conteúdos foi clara?
- Você compreendeu os temas trabalhados ao longo da disciplina?
- Você se sente capaz de aplicar os conhecimentos adquiridos?

No final do formulário também há um espaço para sugestões e opiniões gerais.

3 RELATO DE EXPERIÊNCIA

O presente relato analisa a aplicação da metodologia nas três últimas edições da disciplina, compreendendo os períodos letivos de 2024 a 2026. A observação do

engajamento dos discentes e a avaliação dos trabalhos finais permitiram verificar a eficácia das intervenções realizadas pelo grupo. O foco na modelagem computacional por meio de autômatos celulares exigiu dos alunos a compreensão lógica do método, a capacidade de transcrição para a linguagem Python e a estruturação dos resultados. A escolha dessa abordagem, aplicada à simulação visual de fenômenos físicos ou biológicos, facilitou a interpretação dos dados e mitigou a dificuldade inicial com a abstração matemática comum nos primeiros semestres.

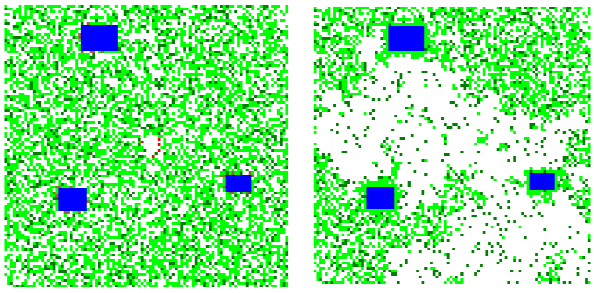
Ao longo das edições, notou-se uma evolução na qualidade técnica e na organização das entregas. Em momentos anteriores, a ausência de um direcionamento estruturado para a elaboração de relatórios e apresentações resultava em trabalhos com deficiências de comunicação técnica. Com a inserção das aulas de boas práticas e a capacitação na ferramenta LaTeX, os discentes passaram a entregar documentos alinhados aos padrões acadêmicos esperados para cursos de engenharia. O suporte direcionado durante as aulas práticas nos laboratórios reduziu as dúvidas operacionais, refletindo na elaboração de códigos mais organizados e simulações mais coerentes. A adoção de ferramentas em nuvem, como o Google Colaboratory e o Overleaf, também contribuíram para esse cenário ao viabilizar a edição simultânea e fortalecer o trabalho colaborativo.

3.1 Aplicações Desenvolvidas

Nas últimas três edições da disciplina, o trabalho teve como tema autômatos celulares, em que cada grupo poderia escolher quais implementações seriam feitas ao código original proposto. Com isso, ao longo das edições da disciplina, foram desenvolvidos os mais diversos tipos e implementações em autômatos celulares pelos alunos. A partir desses trabalhos, é possível notar o quanto os alunos desenvolveram habilidades e exercitaram a capacidade de modelar problemas com técnicas computacionais e trabalho em equipe.

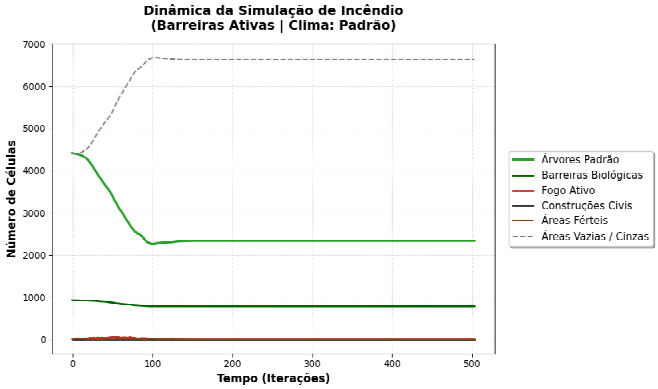
Um trabalho de destaque abordou uma simulação estocástica de incêndio florestal com modificadores ambientais, introduzindo uma malha com oito estados distintos. No relatório, os alunos descreveram ter realizado uma pesquisa prévia sobre os principais meios de contenção e prevenção ambientais e, a partir dela, decidiram modelar o problema implementando: variações climáticas com três tipos de clima que alteram a inflamabilidade do ambiente, vegetações com diferentes níveis de resistência, construções civis, corpos hídricos, áreas férteis e aceiros. O resultado dessa modelagem espacial complexa é ilustrado na Figura 3, enquanto a Figura 4 apresenta graficamente a dinâmica populacional e a evolução quantitativa de cada um desses estados ao longo das iterações.

Figura 3 – Evolução espaço-temporal e dinâmica populacional da simulação estocástica de incêndio florestal



Fonte: Os autores (2026)

Figura 4 – Gráfico de evolução espaço-temporal e dinâmica populacional da simulação estocástica de incêndio florestal

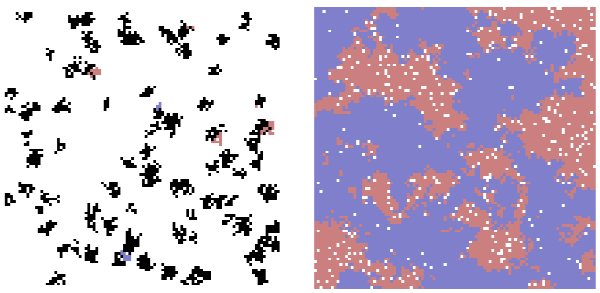


Fonte: Os autores (2026)

Ao estruturar essa aplicação, os discentes aprimoraram significativamente a capacidade de compreender fenômenos complexos, consolidando a habilidade de transpor situações do mundo real para abstrações matemáticas e lógicas. Mais do que a construção do código, as equipes vivenciaram uma das etapas mais críticas da modelagem computacional: a validação e a análise de sensibilidade do modelo. Ao variar os parâmetros estocásticos e ambientais, os estudantes realizaram uma avaliação pertinente dos dados gerados, aprendendo a interpretar criticamente não apenas os resultados gráficos finais, mas também as causas e as consequências diretas que cada variável exercia na dinâmica do sistema.

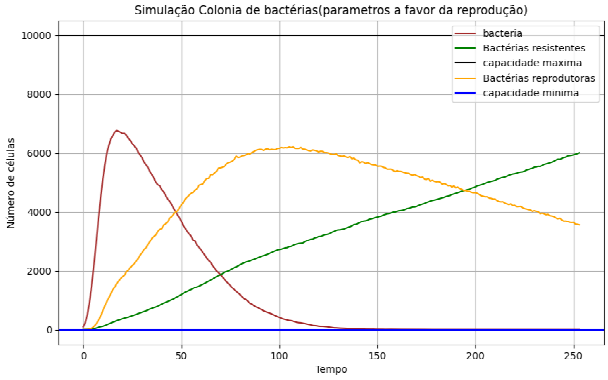
Já no contexto da colônia de bactérias, é possível destacar um trabalho que teve como foco a parametrização, o teste de hipóteses e a dinâmica de sistemas ao longo do tempo. Aqui, os alunos focaram em representar a expansão territorial, a mutação e a competição biológica de microrganismos. A malha foi estruturada com quatro estados fundamentais: espaço vazio, bactéria comum, bactéria resistente (com menor taxa de mortalidade) e bactéria super-reprodutora (com maior probabilidade de proliferação). A Figura 5 demonstra visualmente a competição espacial dessas colônias bacterianas parametrizadas a favor da reprodução. Complementarmente, a Figura 6 exibe o gráfico detalhando o crescimento e a sobreposição populacional das diferentes cepas no decorrer do tempo.

Figura 5 – Análise visual e quantitativa da competição espacial em colônias bacterianas parametrizadas a favor da reprodução.



Fonte: Os autores (2026)

Figura 6 – Gráfico de análise visual e quantitativa da competição espacial em colônias bacterianas parametrizadas a favor da reprodução.



Fonte: Os autores (2026)

O grande destaque metodológico da equipe foi a implementação de um sistema estocástico de mutação capaz de modificar as características das bactérias, impactando diretamente o comportamento global do modelo. A estruturação desse algoritmo evidenciou um notável amadurecimento no pensamento científico e computacional dos discentes voltado à modelagem de problemas complexos. Além disso, a atividade proporcionou uma valiosa experiência prática com sistemas dinâmicos sensíveis às condições iniciais e paramétricas, permitindo que os estudantes realizassem simulações comparativas para analisar o domínio populacional em diferentes cenários virtuais.

3.2 Avaliação da Percepção Discente

Para quantificar a percepção dos estudantes sobre a proposta didática, aplicou-se o formulário de avaliação mencionado na metodologia, o qual obteve resposta de oito discentes. Os resultados numéricos foram padronizados em índices percentuais de satisfação para facilitar a análise. Em relação à contribuição do GET para a disciplina, o índice de aprovação registrado foi de 87,6%. Quando questionados se os conteúdos abordados contribuíram para o aprendizado, a percepção positiva também alcançou 87,6%, mesmo valor quantificado para o nível de compreensão geral dos temas trabalhados. A clareza na forma de apresentação dos conteúdos obteve a menor taxa relativa, com 85%. Por outro lado, quanto à percepção de capacidade para aplicar os conhecimentos adquiridos na prática, a turma indicou o maior índice do levantamento, correspondente a 90%.

No espaço destinado a sugestões, os relatos apontaram a relevância da disciplina como uma introdução efetiva à graduação. Em contrapartida, as críticas construtivas concentraram-se no ritmo e na densidade das aulas de programação. Discentes indicaram que o ensino de Python e de suas bibliotecas ocorreu de forma acelerada, gerando certa insegurança frente à complexidade das aplicações exigidas no desenvolvimento do projeto final. Esse aspecto foi bem ilustrado pelo relato de um dos estudantes, que pontuou: *“senti que os conteúdos [...] foram ensinados com certa pressa. Como esses conteúdos são aplicados nos trabalhos a seguir, sinto que poderia ter havido uma melhor preparação prévia”.* Como sugestão de melhoria, foi recomendada a reestruturação do cronograma das aulas teóricas. A proposta apresentada pelos estudantes sugere a divisão do conteúdo de programação em três encontros: dois dedicados exclusivamente à assimilação da lógica e da sintaxe básica da linguagem, e um focado apenas em bibliotecas essenciais para a manipulação de matrizes e dados, como NumPy e Pandas. Essa readequação visa deixar

ferramentas computacionais específicas para aprendizado sob demanda, reduzindo assim a sobrecarga cognitiva inicial.

4 LIÇÕES APRENDIDAS

Uma atividade de ensino no geral promove grande aprendizado não somente para os que se encontram na posição de alunos, mas também é essencialmente enriquecedora para seus ministrantes. O GET, sabe e pauta nisso seus projetos, alinhando-se ao pensamento de Freire (1996), que defende que "quem ensina aprende ao ensinar". A partir disso, foram percebidos aprendizados gerais dos membros envolvidos na atividade.

4.1 Aprendizados na Gestão da Heterogeneidade Discente

É inegável que um dos maiores desafios da educação seja lidar com as diferentes necessidades dos discentes. Em uma universidade como a UFJF, o corpo estudantil é formado por indivíduos dos mais diversos contextos sociais e acadêmicos, apresentando interesses e dificuldades bastante heterogêneos. Nesse cenário, ministrar aulas em uma disciplina introdutória na qual os alunos ainda demandam um suporte muito mais personalizado do que estudantes em final de graduação mostra-se uma tarefa tão desafiadora quanto enriquecedora. Afinal, durante as exposições, é necessário gerir e balancear a forma como o conteúdo é abordado para contemplar essas demandas particulares, uma competência fundamental no ensino superior contemporâneo (MASETTO, 2012).

Para lidar com essa questão, conforme abordado na metodologia, o GET adotou a estratégia de manter sempre ao menos três membros em cada aula, garantindo um suporte mais individualizado. Dessa forma, quando um aluno apresenta maior dificuldade de compreensão ou, inversamente, maior interesse em um tópico específico, é possível designar um dos membros exclusivamente para suprir essas necessidades. Consequentemente, esse contato mais próximo com os ingressantes permite aos integrantes do grupo interagir com diferentes ritmos de aprendizagem e visões de mundo, o que promove um precioso crescimento pessoal, profissional e acadêmico para toda a equipe.

4.2 A Importância da Interdisciplinaridade

Após a participação, elaboração e construção do projeto, os membros relataram perceber ainda mais a importância da interdisciplinaridade. No contexto de um curso inerentemente interdisciplinar como a Engenharia Computacional, essa abordagem é natural, mostrando-se tanto enriquecedora quanto desafiadora. Isso ocorre porque, durante as aulas, é preciso gerir e balancear as diferentes áreas de conhecimento, esclarecendo aos participantes como essa integração se dá na prática.

Percebe-se, também, que a interdisciplinaridade oportuniza um grande crescimento acadêmico e profissional. Com a proposta de relacionar conceitos, o estudante não pode ficar preso apenas à aplicação de fórmulas ou à reprodução de padrões; é fundamental compreender a natureza do problema para destrinchá-lo e encontrar suas soluções. No contexto do uso de autômatos celulares para a simulação de incêndios florestais, por exemplo, é necessário que o aluno compreenda muito bem as ferramentas computacionais. Só assim ele será capaz de traduzir os eventos do mundo real, que são fenômenos distantes do ambiente teórico das salas de aula de matemática e programação, em um modelo lógico estruturado.

4.3 Evolução do Projeto: Uma Abordagem Pesquisa-Ação

A atuação do GET na disciplina caracteriza-se como uma prática pedagógica contínua de mais de cinco anos, fundamentada em um modelo de pesquisa-ação (Thiollent, 2011), no qual a metodologia é iterativamente refinada com base na análise de resultados anteriores. Observando o recorte longitudinal das três últimas edições, implementaram-se ajustes curriculares estratégicos. Na primeira delas, o cronograma possuía apenas uma aula introdutória ao Python e nenhuma voltada para boas práticas de escrita ou apresentação, priorizando a variedade de aplicações. No entanto, era perceptível a defasagem dos discentes tanto na lógica de programação quanto nas soft skills de apresentação, organização de projetos em equipe e elaboração de relatórios. Essa lacuna foi notada tanto pelos membros do GET quanto pelos professores da disciplina e de matérias mais avançadas.

Para solucionar essa questão, na segunda edição desse recorte temporal, introduziu-se uma aula extra de práticas em Python focada em bibliotecas, além de um encontro sobre boas práticas de apresentação. Ainda assim, observou-se pouco engajamento por parte dos alunos. Isso levou o grupo a adicionar atividades avaliativas pontuais para esses respectivos temas. Essa estratégia surtiu um efeito muito positivo na última edição, com relatos de professores e discentes do GET destacando a formação de uma base técnica mais robusta e uma organização consideravelmente superior nos trabalhos finais.

5 CONSIDERAÇÕES FINAIS

A atuação do GET na disciplina de Introdução à Engenharia Computacional apresenta-se como uma iniciativa prática para a formação técnica e a integração dos discentes ao ambiente universitário. A análise do período letivo entre 2024 e 2026 indica que a aplicação da Aprendizagem Baseada em Projetos estimulou o engajamento da turma. A adoção da modelagem de autômatos celulares, em conjunto com a inserção de aulas voltadas ao desenvolvimento de competências de comunicação, como o uso de LaTeX e técnicas de apresentação, atenuou deficiências observadas em semestres anteriores. As avaliações e as entregas dos estudantes confirmaram que esta reestruturação resultou em trabalhos com maior rigor técnico e melhor articulação coletiva.

O processo de aprimoramento, embasado nos princípios da pesquisa-ação, reiterou a eficácia da mentoria de pares para fornecer um suporte individualizado frente à heterogeneidade da turma. O levantamento de dados junto aos alunos, no entanto, expôs a necessidade de ajustes no ritmo das aulas de programação, consideradas densas e aceleradas pelos ingressantes. Como perspectivas futuras, propõe-se a readequação do cronograma para permitir uma assimilação mais gradual da lógica computacional e de suas ferramentas auxiliares. Sugere-se também o acompanhamento longitudinal desses estudantes para mensurar o impacto das intervenções iniciais em seu desempenho acadêmico ao longo do curso.

AGRADECIMENTOS

Os autores agradecem à instituição de fomento pelas bolsas concedidas e ao corpo docente do curso pelo espaço cedido. Expressando também a gratidão aos professores da UFJF pela confiança, parceria e oportunidade de colaboração ativa nas atividades da disciplina.

REFERÊNCIAS

BACICH, L.; MORAN, J. (Org.). **Metodologias ativas para uma educação inovadora: uma abordagem teórico-prática**. Porto Alegre: Penso, 2018.

BARBOSA, E. F.; MOURA, D. G. Metodologias ativas de aprendizagem no ensino de Engenharia. In: INTERNATIONAL CONFERENCE ON ENGINEERING AND TECHNOLOGY EDUCATION, 13., 2014, Guimarães. **Anais...** Guimarães: COPEC, 2014. p. 111-117.

BARTSCH, Robert A.; COBERN, Kristi M. Effectiveness of PowerPoint presentations in lectures. **Computers & Education**, v. 41, n. 1, p. 77–86, 2003.

BENDER, W. N. **Aprendizagem baseada em projetos: educação diferenciada para o século XXI**. Porto Alegre: Penso, 2014.

FIGUERÊDO, J. S. L. et al. A experiência da monitoria de Algoritmos e Programação em cursos de engenharia na perspectiva dos monitores. In: SIMPÓSIO BRASILEIRO DE EDUCAÇÃO EM COMPUTAÇÃO (EDUCOMP), 1., 2021, Evento Online. **Anais...** Porto Alegre: SBC, 2021. p. 183-192.

FREIRE, P. **Pedagogia da autonomia: saberes necessários à prática educativa**. São Paulo: Paz e Terra, 1996.

IEEE WIE UFJF. A relevância das atividades do IEEE WIE UFJF para desenvolvimento de soft skills em graduandas em engenharia. In: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA (COBENGE), 49., 2021, Evento Online. **Anais...** Brasília: ABENGE, 2021.

LAMPORT, L. **LaTeX: a document preparation system**. 2. ed. Reading: Addison-Wesley, 1994.

LOBOSCO, Marcelo et al. Design and Analysis of an Undergraduate Computational Engineering Degree at Federal University of Juiz de Fora. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE, 2018, Cham. **Anais...** Cham: Springer International Publishing, 2018. p. 293-303.

MARINHO, C. O.; SILVA, I. M. L.; COÊLHO, D. G. A importância de disciplinas introdutórias no ensino da engenharia química. In: CONGRESSO BRASILEIRO DE ENGENHARIA QUÍMICA, 2018. **Anais...** Blucher Chemical Engineering Proceedings, v. 1, p. 3246-3249, 2018. Disponível em: http://www.proceedings.blucher.com.br/article-details/a-importncia-de-disciplinas-introdutrias-no-ensino- da-engenharia-qumica-29247. Acesso em: 23/04/2026.

OLIVEIRA, C. N. de; SOUSA, L. M. O.; SAMPAIO, T. S. Sistemas de Representação do Conhecimento e Educação: Contribuições Epistemológicas e Pedagógicas da Máquina de Turing, Fractais e dos Autômatos Celulares. **Artefactum - Revista de Estudos Interdisciplinares**, v. 20, n. 1, 2021. Disponível em: https://www.artefactum

journal.com/index.php/artefactum/article/view/2024. Acesso em: 23/04/2026.

SEVERINO, A. J. **Metodologia do trabalho científico**. 24. ed. São Paulo: Cortez, 2016.

SHIFLET, Angela B.; SHIFLET, George W. **Introduction to Computational Science: modeling and simulation for the sciences**. 2. ed. Princeton, NJ: Princeton University Press, 2014.

THIOLLENT, M. **Metodologia da pesquisa-ação**. 18. ed. São Paulo: Cortez, 2011.

TOCK, Kalée. Google CoLaboratory as a platform for Python coding with students. **RTSRE Proceedings**, v. 2, n. 1, 2019.

VAN ROSSUM, Guido et al. **Python 3 Reference Manual**. Scotts Valley, CA: Createspace, 2009. 242 p.

PROGRAMMING AND MODELING PRACTICES IN THE INITIAL TRAINING OF COMPUTER ENGINEERING STUDENTS: AN APPLICATION OF ACTIVE METHODOLOGIES

Abstract: *In undergraduate programs, especially in the early stages, incoming students often struggle to fully grasp the practical scope of their future professional field, making introductory courses essential for academic integration and leveling. In Computational Engineering, this initial contact occurs through the study of computational modeling, an abstract area that typically generates initial resistance. To mitigate this difficulty, this paper reports the intervention of a tutorial education group that, in partnership with the faculty, applied the principles of Project-Based Learning and peer mentoring in the course. The methodology consisted of structuring practical classes in Python, LaTeX, and communication skills, culminating in the development of a visual simulation final project using cellular automata. Data analysis from the 2024 to 2026 editions demonstrated the effectiveness of the approach, recording an 87.6% student approval rate regarding the initiative's contribution to practical learning. The continuous evaluation process also highlighted the need to readjust the pace of programming classes, generating an action plan focused on reducing cognitive overload and improving future editions of the course.*

Keywords: *Computational Engineering; Tutorial Education Program; Computational Modeling; Python; Engineering Education; Cellular Automata.*