

## Formando engenheiros na era digital: tutoria remota em programação sob a perspectiva CTS

### RESUMO

**Riam Ferreira Lopes Duarte**  
Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG), Belo Horizonte, Minas Gerais, Brasil  
[riamlopes98@outlook.com](mailto:riamlopes98@outlook.com)

**Tatielle Menolli Longhini**  
Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG), Belo Horizonte, Minas Gerais, Brasil  
[tatielle.longhini@gmail.com](mailto:tatielle.longhini@gmail.com)

**Bruno de Souza Toledo**  
Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG), Belo Horizonte, Minas Gerais, Brasil  
[bruno.toledo@ifmg.edu.br](mailto:bruno.toledo@ifmg.edu.br)

As disciplinas de Lógica de Programação e Programação de Computadores apresentam, historicamente, elevados índices de reprovação e evasão nos cursos de Engenharia. Este estudo, realizado no IFMG – campus Governador Valadares, identificou que cerca de 70% dos alunos reprovaram ao menos uma vez em disciplinas da área e levaram, em média, de um a três semestres adicionais para obter aprovação. A partir da análise do perfil discente e das principais dificuldades relatadas, foi elaborado o projeto Prog X, um programa de tutoria remota estruturado para oferecer apoio personalizado, com encontros síncronos e assíncronos, resolução de exercícios práticos e acompanhamento contínuo por tutores. O objetivo é consolidar os conhecimentos básicos em programação, potencializando o aprendizado, reduzindo reprovação, evasão e abandono, além de fortalecer a permanência e o desempenho acadêmico nos cursos de Engenharia. Diferencialmente, a proposta é discutida sob a perspectiva CTS (Ciência-Tecnologia-Sociedade), relacionando a aprendizagem em programação ao desenvolvimento de competências críticas e reflexivas para a formação de engenheiros na era digital.

**PALAVRAS-CHAVE:** Aprendizagem em Programação. Evasão Universitária. Apoio Pedagógico. Ensino Remoto. Abordagem CTS.

## INTRODUÇÃO

A Lógica de Programação é uma disciplina complexa por tradição para alunos iniciantes, na qual tem sido apontada como fator para altos índices de reaprovação e evasão, principalmente após conhecidas as barreiras nas disciplinas de programação: “Entre as competências mais difíceis de serem desenvolvidas estão as relacionadas com o desenvolvimento de algoritmos e programas” (RAPKIEWICZ et al., 2006). Uma boa parcela de alunos possui dificuldade em assimilar e aplicar os conceitos de programação, principalmente para criar algoritmos para resolução de problemas concretos (SILVA; TRENTIN, 2016).

O ensino da Lógica de Programação e Algoritmos se tornou obrigatório para quase todos os cursos de Engenharia, buscando conceitos e estratégias para a aplicação da disciplina e definição da linguagem de programação a ser adotada. A inserção dessas disciplinas no ciclo básico ocorre, principalmente, devido à necessidade do desenvolvimento do raciocínio lógico do aluno e também pela utilização dos conceitos ao longo do curso (MARCUSSI et al., 2016).

Aliado à realidade de não estar acostumado a resolver problemas em busca de soluções, está a não familiarização que a maioria dos alunos tem com disciplinas de programação. A necessidade de prática em técnicas de resolução de problemas e de um bom nível de conhecimento são fatores que dificultam a aprendizagem (MENDES, 2002). É necessário que o sentido da aprendizagem esteja evidente, pois dessa maneira o aluno conseguirá compreender os conteúdos estudados (BARBOSA, 2004). Associado ao ensino tradicional, o aluno precisa ser emancipado, guiando-o para “aprender a aprender” e validando seus próprios conhecimentos (WANG, 2015).

Uma parcela dos alunos nunca tivera contato com o meio da Programação de Computadores antes de ingressar na faculdade. Diante disso, o estudante leigo se depara com enunciados complexos, exigindo a formulação de algoritmos para resolução de problemas, o que resulta em uma experiência pouco atraente ou desestimulante (BORGES, 2000). Nesse sentido, Pereira (2013) reforça a importância do contato precoce com o pensamento lógico-computacional no ensino básico, de modo a facilitar a interpretação de sintaxes e a construção de algoritmos. Outra etapa problemática no processo de ensino-aprendizagem é a verificação da funcionalidade e da utilidade dos algoritmos, normalmente realizada manualmente pelos próprios alunos (RAPKIEWICZ et al., 2006).

Identificada essa dificuldade, torna-se necessário o desenvolvimento de ferramentas e estratégias pedagógicas capazes de apoiar o professor e incentivar os alunos ao estudo da programação. Ações pedagógicas voltadas à resolução de problemas são fundamentais para desenvolver habilidades como atenção, concentração e pensamento criativo (RAPKIEWICZ et al., 2006). Dentre essas ações, encontra-se a tutoria, processo que concede suporte humanizado e personalizado ao aluno, em parceria com o professor, com o objetivo de motivar e melhorar a experiência de aprendizagem.

Nas últimas décadas, os estudos em Ciência, Tecnologia e Sociedade (CTS) têm se consolidado como abordagem formativa que supera a visão tecnicista e promove uma compreensão crítica da tecnologia (AULER; BAZZO, 2001). No ensino de Engenharia, esse enfoque propõe integrar competências técnicas ao desenvolvimento de uma consciência social, favorecendo profissionais reflexivos e

capazes de avaliar os impactos da ciência e da tecnologia na sociedade (BAZZO, 1998; 2000; FERREIRA et al., 2017; BAZZO et al., 2003). Nesse cenário, a programação pode ser compreendida não apenas como instrumento operacional, mas como meio para estimular o pensamento crítico e a cidadania tecnológica. Estudos recentes apontam avanços na incorporação do enfoque CTS na graduação tecnológica, ainda que de forma incipiente, fortalecendo a reflexão sobre o papel social do engenheiro (FERRI, 2018).

Dessa maneira, este estudo propõe desenvolver um programa de tutoria para alunos matriculados em disciplinas de programação de computadores, consolidando os conhecimentos básicos necessários para o desempenho acadêmico nos cursos de Engenharia do Instituto Federal de Minas Gerais – campus Governador Valadares (IFMG-GV).

O trabalho possui cinco tópicos: o primeiro aborda a introdução, o problema de pesquisa, e os objetivos; o segundo discute o referencial teórico, enfocando a importância do ensino de programação no Brasil e suas consequências; o terceiro descreve a abordagem metodológica e o desenvolvimento da pesquisa; o quarto apresenta os resultados da análise de dados sobre reaprovação e evasão em disciplinas específicas; e o último trata das considerações finais, dificuldades, aceitação dos envolvidos, manutenção e continuidade do projeto para o próximo semestre.

## REFERENCIAL TEÓRICO

Este referencial aborda o ensino das disciplinas de programação de computadores, as taxas de reaprovação na graduação e destaca a relevância do ensino dessas disciplinas para profissionais contemporâneos. Além disso, são discutidas as dificuldades no processo de ensino-aprendizagem, as evasões observadas nas disciplinas e as ações comumente implementadas para enfrentar esses desafios.

### Programação e formação crítica na Engenharia: desafios do ensino de programação sob a perspectiva CTS

Os avanços tecnológico-científico transformaram o campo da computação, programação de computadores e aparelhos eletrônicos. A tecnologia impôs uma exigência aos profissionais contemporâneos: compreender a programação de computadores para cultivar o pensamento lógico e encontrar novas soluções para problemas cotidianos. As disciplinas relacionadas à programação buscam transformar a mentalidade dos alunos, desenvolvendo habilidades e competências que os destacarão no mercado de trabalho (LIMA, 2018).

Dados demonstram que o Brasil sofre de uma grande defasagem de profissionais que consigam entender linguagens e códigos de programação. O número de vagas em aberto a 3 anos atrás eram de 100 mil e com projeções para chegar em 200 mil nesse ano de 2021 (D'ANGELO, 2021). Há uma crescente tendência globalizada por falta de mão de obra especializada nas áreas de tecnologia, sendo que o Brasil ainda é influenciado pelo ensino de baixa qualidade (em comparação com a demanda internacional), e pela baixa compreensão da língua inglesa (SENA, GRANATO, 2021).

Disciplinas relacionadas a programação são consideradas desafiadoras e por isso precisam de uma atenção especial das universidades que buscam com frequência melhores estratégias para diferenciar o ensino e obter um resultado satisfatório de aprovação e aprendizagem (BRITO, 2019). Garlet et al. (2018) intensificam a importância dessas disciplinas para que o profissional saiba produzir tecnologias e não apenas ser um consumidor.

O objetivo central da programação deve ser o foco em ensinar a lógica por detrás dos códigos, já que a lógica possui grande semelhança em diferentes linguagens e não simplesmente focar em uma linguagem específica. Nem todos os formandos a profissionais da atualidade seguirão o caminho para se tornarem um programador de computadores, mas é importante ressaltar que possuirão capacidades de pensar e agir com maior criatividade (GARLET et al., 2018). Diante das capacidades adquiridas o discente pode sanar as dificuldades para aprender as matérias de programação.

A perspectiva Ciência-Tecnologia-Sociedade (CTS) constitui um referencial fundamental para compreender a inserção da programação nos cursos de Engenharia. Autores clássicos, como Auler e Bazzo (2001) e Santos e Mortimer (2000), destacam que a abordagem CTS favorece uma formação crítica, na qual a tecnologia é compreendida não apenas como instrumento técnico, mas como uma construção social permeada por valores, interesses e impactos diretos na vida cotidiana. Nesse sentido, o ensino de programação, ao estimular o pensamento lógico e a resolução de problemas, pode também contribuir para a emancipação discente, promovendo inserção cidadã em uma sociedade cada vez mais digitalizada.

Pesquisas recentes reforçam esse potencial. Correa e Bazzo (2017) argumentam que o enfoque CTS possibilita tanto a humanização do trabalho docente quanto o engajamento discente em práticas mais reflexivas. Rodrigues (2017) evidencia que a abordagem favorece a articulação entre teoria e prática, aproximando a aprendizagem de contextos significativos para os estudantes. Pelissari (2023), por sua vez, demonstra que a CTS potencializa a construção da autonomia e da criticidade em diferentes ambientes formativos, ampliando a relevância do ensino em áreas tecnológicas.

Além disso, Von Linsingen (2015) sistematiza fundamentos teóricos do movimento CTS, sublinhando sua capacidade de transformar o ensino em práticas emancipadoras. Tais contribuições são convergentes com evidências empíricas que mostram que metodologias ativas mediadas pela CTS não apenas ampliam a motivação, mas também contribuem para reduzir a evasão universitária. Dessa forma, práticas pedagógicas inovadoras fundamentadas no enfoque CTS se apresentam como estratégias promissoras para ressignificar o ensino de programação na Engenharia, tornando-o mais crítico, inclusivo e socialmente relevante.

Os avanços científico-tecnológicos transformaram profundamente a computação, a programação e os dispositivos eletrônicos. Nesse cenário, compreender linguagens de programação tornou-se requisito essencial para o desenvolvimento do pensamento lógico e para a criação de soluções inovadoras aplicáveis a problemas cotidianos (LIMA, 2018).

Entretanto, o Brasil enfrenta um déficit significativo de profissionais qualificados. Há três anos, o número de vagas não preenchidas no setor de tecnologia era de 100 mil, com projeção de alcançar 200 mil em 2021 (D'ANGELO, 2021). Essa carência está relacionada não apenas à crescente demanda global por

mão de obra especializada, mas também a limitações estruturais do ensino nacional, como a baixa qualidade da formação técnica e a insuficiente proficiência em língua inglesa (SENA; GRANATO, 2021).

As disciplinas de programação, por sua complexidade, são frequentemente desafiadoras para os estudantes e exigem das universidades estratégias pedagógicas diferenciadas, que favoreçam a aprendizagem e reduzam os índices de evasão (BRITO, 2019). Nesse sentido, Garlet et al. (2018) ressaltam a importância de tais disciplinas para que o futuro engenheiro atue como produtor de tecnologia e não apenas como consumidor. Mais do que dominar uma linguagem específica, o ensino deve priorizar a lógica por trás dos códigos, pois essa habilidade é transferível entre diferentes linguagens e potencializa a criatividade e a capacidade de resolução de problemas, mesmo para aqueles que não seguirão carreira como programadores.

A perspectiva Ciência-Tecnologia-Sociedade (CTS) surge como referencial estratégico para ressignificar o ensino de programação na Engenharia. De acordo com Auler e Bazzo (2001) e Santos e Mortimer (2000), o enfoque CTS amplia a compreensão da tecnologia como construção social permeada por valores, interesses e impactos cotidianos, promovendo uma formação crítica e cidadã. Assim, o ensino de programação, ao desenvolver raciocínio lógico e autonomia intelectual, pode também favorecer a emancipação discente em uma sociedade digitalizada.

Pesquisas recentes reforçam esse potencial. Correa e Bazzo (2017) destacam que a abordagem CTS contribui tanto para a humanização do trabalho docente quanto para o engajamento discente. Rodrigues (2017) aponta que ela aproxima teoria e prática, conectando a aprendizagem a contextos significativos. Já Pelissari (2023) demonstra que a CTS estimula a autonomia e a criticidade em diversos ambientes formativos. Von Linsingen (2015), ao sistematizar os fundamentos do movimento CTS, evidencia seu caráter emancipador e transformador. Em convergência, metodologias ativas orientadas por esse referencial têm mostrado potencial para aumentar a motivação dos estudantes e reduzir a evasão universitária. Portanto, práticas pedagógicas inovadoras fundamentadas no enfoque CTS configuram-se como caminhos promissores para o ensino de programação em cursos de Engenharia, tornando-o mais crítico, inclusivo e socialmente relevante.

### Ensino de programação e a reaprovação de alunos na graduação

O elevado índice de reaprovação e evasão em disciplinas de programação de computadores tem sido recorrente nos últimos anos (REINO et al., 2015), o que impulsiona pesquisas voltadas ao aprimoramento do ensino e da aprendizagem de algoritmos e programação (DAMASCENO; CARNEIRO, 2018; DEJAVITE et al., 2017). As dificuldades mais frequentes entre os estudantes estão relacionadas à limitação na expressão escrita, interpretação de enunciados, hábitos de estudo insuficientes e lacunas na formação em matemática básica (GIRAFÁ; MORA, 2013).

No âmbito técnico, os entraves mais comuns envolvem problemas de sintaxe, uso de variáveis, compreensão de estruturas condicionais, interpretação das tarefas propostas e manejo de erros no código (AURELIANO et al., 2016). Tais obstáculos revelam que o desafio de aprender programação é igualmente desafiador para ensiná-la, evidenciando o papel central da metodologia e da didática docente na mediação do processo de aprendizagem (SILVA et al., 2019;

HOLANDA et al., 2018; BOSSE; GEROSA, 2015; ODAKURA; SALAZAR; BARVINSKI, 2015; RODRIGUES et al., 2013).

As reprovações decorrem de múltiplos fatores: primeiro contato com o conteúdo, ausência de conhecimentos básicos em computação, exigência precoce de pré-requisitos, excesso de faltas, turmas numerosas e até desinteresse mesmo em presença física nas aulas (KRZYZANOWSKI, 2019; GIRAFÁ; MORA, 2013). Quando se acumulam reprovações, muitos estudantes perdem o estímulo e a motivação para persistir na disciplina (PASCOAL et al., 2015).

Além desses aspectos específicos, outros fatores estruturais impactam a evasão em cursos de engenharia e tecnologia no Brasil (Figura 1), reforçando a necessidade de estratégias institucionais de apoio pedagógico que reduzam reprovações, evitem o abandono e fortaleçam a permanência discente (REINO et al., 2015).

Figura 1 - Motivos de evasão.



Fonte: Reino et al. (2015).

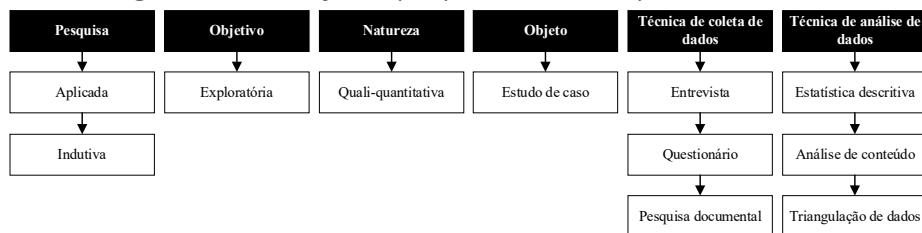
Uma vez identificados os principais fatores que contribuem para a reprovação em disciplinas de programação, torna-se essencial adotar estratégias eficazes de enfrentamento. Entre elas, destacam-se os programas de tutoria e a ampliação do ensino de programação no nível básico, medidas que visam reduzir a evasão e melhorar o desempenho discente.

Os sistemas de tutoria têm se consolidado como ferramentas relevantes de apoio à aprendizagem, permitindo equilibrar o ritmo de estudo dos alunos e oferecer suporte contínuo. Para alcançar bons resultados, é fundamental que o tutor compreenda a realidade do estudante, ajustando o conteúdo de nivelamento e o ritmo de ensino às necessidades individuais, de forma a promover uma atenção mais personalizada (FREIRE et al., 2019; SCAICO et al., 2012).

No que se refere ao ensino de programação no ensino básico no Brasil, observa-se um campo ainda pouco explorado, mas com grande potencial de expansão (AURELIANO; TADESCO, 2012). Nessa etapa da formação, crianças e adolescentes encontram-se em pleno desenvolvimento de habilidades cognitivas, como a capacidade de resolver problemas e o raciocínio lógico. Assim, iniciativas que introduzam conceitos de programação de maneira lúdica e acessível — por meio de atividades como lógica de jogos, desenvolvimento de softwares simples e educação em robótica — podem estimular significativamente a aprendizagem, despertando o interesse e fortalecendo competências fundamentais para etapas posteriores (SILVA, 2017).

## METODOLOGIA

Figura 2 - Classificação da pesquisa científica do presente estudo.



Fonte: autoria própria.

A metodologia desta pesquisa adota uma abordagem quali-quantitativa, combinando dados estatísticos objetivos com informações subjetivas obtidas por questionários. O estudo examina os índices de reprovação, evasão e abandono nas disciplinas de programação dos cursos de Engenharia de Produção, Engenharia Civil e Engenharia Ambiental e Sanitária do IFMG campus Governador Valadares (IFMG-GV), no período de 2019.1 a 2021.1, quando os três cursos passaram a funcionar de forma integrada.

O método científico utilizado é o indutivo, partindo da observação de fenômenos específicos - como os altos índices de reprovação, para formular hipóteses explicativas e possíveis intervenções. Segundo Lakatos e Marconi (2010), o método indutivo permite construir generalizações a partir da análise de dados empíricos.

A pesquisa é aplicada, pois visa implementar, de forma prática, um programa de tutoria no IFMG-GV para reduzir a evasão e melhorar o desempenho acadêmico em disciplinas de programação (Silva & Menezes, 2005). Também apresenta caráter exploratório, ao envolver revisão bibliográfica, aplicação de questionários semiestruturados a sujeitos diretamente relacionados ao problema e análise de casos representativos (Tumelero, 2019; Gil, 2007).

Trata-se de um estudo de caso, com foco na realidade específica do IFMG-GV, incluindo coleta e análise de dados sobre estudantes e seu desempenho em programação. O objetivo é compreender em profundidade os fatores que impactam a aprendizagem e avaliar a eficácia de uma intervenção baseada em tutoria acadêmica (Yin, 2001).

A coleta de dados foi realizada por questionários semiestruturados aplicados a ex-monitores e tutores das disciplinas de programação. Esses tutores, selecionados anualmente por meio de provas, entrevistas e análise de desempenho acadêmico, passaram por capacitação conduzida pelos docentes da área. Sua atuação foi decisiva para identificar dificuldades recorrentes dos alunos, avaliar a frequência às monitorias e compreender percepções sobre o processo de ensino-aprendizagem.

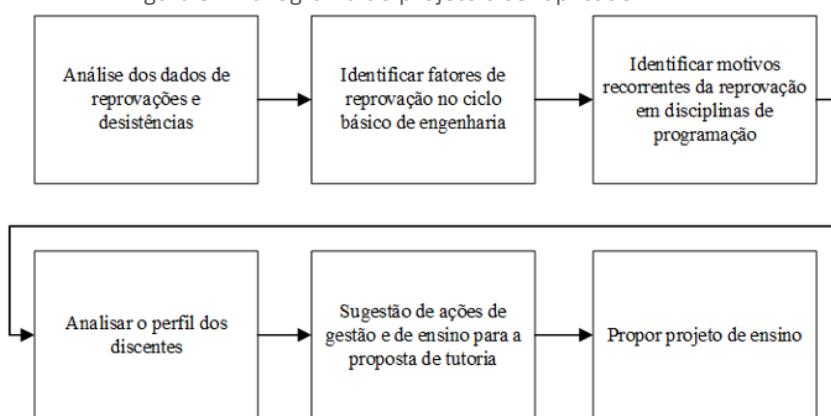
Os questionamentos buscaram identificar principais barreiras enfrentadas pelos discentes, o padrão de adesão às monitorias e a diversidade de participação. Também investigaram se a concessão de pontuação extra no projeto “PROG X” poderia estimular maior engajamento.

Adicionalmente, 30 estudantes responderam a questionários abertos e fechados (Anexo II), abordando temas como familiaridade com tecnologias, experiências prévias, histórico de reprovação, tempo de estudo e uso de atividades de apoio. O objetivo foi traçar o perfil acadêmico e cognitivo dos alunos, considerando conhecimentos prévios, matrícula, reprovação, evasão e envolvimento em atividades extracurriculares.

A pesquisa incluiu ainda análise documental, com dados fornecidos pela Secretaria de Registro Acadêmico do IFMG-GV, permitindo levantar estatísticas detalhadas sobre aprovação, reprovação e evasão em Programação de Computadores (Lakatos & Marconi, 2001).

As técnicas analíticas empregadas foram: análise de conteúdo, para interpretação de dados qualitativos (Campos, 2004); estatística descritiva, para organização e interpretação dos dados quantitativos (Santos, 2007); e triangulação de dados, que integrou múltiplas fontes e técnicas para maior consistência dos resultados (Zappellini & Feuerschütte, 2015). O processamento dos dados foi realizado no Microsoft Excel®, que possibilitou sistematização e visualização das informações. Para a aplicação prática deste estudo foi utilizada a sequência metodológica apresentada pela Figura 3.

Figura 3 - Fluxograma do projeto a ser aplicado.



Fonte: autoria própria.

Inicialmente, foram analisados os dados fornecidos pela Secretaria Acadêmica do IFMG-GV sobre índices de aprovação, reprovação, evasão e abandono nas disciplinas de Lógica e Programação de Computadores I e II, dos cursos de Engenharia de Produção, Engenharia Ambiental e Sanitária e Engenharia Civil. Em seguida, foram identificados os principais fatores associados às reprovações nas disciplinas do ciclo básico de Engenharia.

Essas informações permitem compreender melhor o perfil e as necessidades do público-alvo, oferecendo subsídios para a elaboração de estratégias que favoreçam o desempenho acadêmico em programação. Conforme a Resolução nº 510, de 07 de abril de 2016, pesquisas que utilizam bancos de dados com informações agregadas, sem possibilidade de identificação individual, estão dispensadas de aprovação prévia por Comitê de Ética em Pesquisa com Seres Humanos.

Diante desse cenário, propõe-se a implementação de um projeto de tutoria, voltado à consolidação dos conhecimentos fundamentais exigidos em Lógica e Programação de Computadores I e II. O objetivo é reduzir os índices de reprovação, evasão e abandono, oferecendo acompanhamento adaptado à realidade e às rotinas dos estudantes.

## RESULTADOS E DISCUSSÕES

Neste capítulo, apresentamos os resultados obtidos das análises de dados referentes as disciplinas de Lógica de Programação, Programação em Computadores I e II, além da proposta do projeto “PROG X”.

### Os cursos de engenharia do IFMG-GV e a realidade dos alunos matriculados nas disciplinas de programação

O curso de Bacharelado em Engenharia de Produção, iniciado em abril de 2010, é o mais antigo de graduação do Instituto Federal de Minas Gerais – campus Governador Valadares (IFMG-GV). Seu objetivo é formar profissionais capacitados para atuar em diferentes setores, com foco em produtividade, qualidade e competitividade no mercado de trabalho. São ofertadas 40 vagas por semestre, com ingresso via Vestibular, SISU, transferência externa ou obtenção de novo título.

A expansão do campus ocorreu em 2018, com a criação do curso de Bacharelado em Engenharia Ambiental e Sanitária, oferecendo 40 vagas anuais. Em 2019, após pesquisas sobre demandas da comunidade e do mercado de trabalho, foi implantado o curso de Engenharia Civil, também com 40 vagas anuais. A partir desse ano, os dados de aprovação, reprovação, evasão e abandono passaram a ser levantados, permitindo a comparação do desempenho dos estudantes das três engenharias nas disciplinas de programação de computadores. As informações foram disponibilizadas pela Secretaria de Registro Acadêmico, sendo preservada a confidencialidade dos alunos.

### Dados de aprovação, reprovação, abandono e evasão dos alunos

A Tabela 1 apresenta os percentuais de aprovação, reprovação, abandono (trancamento e desligamento) e evasão (reprovação por falta) dos alunos dos cursos de Engenharia de Produção, Engenharia Ambiental e Sanitária e Engenharia Civil. As disciplinas analisadas correspondem ao ciclo básico de programação de computadores, a saber: Lógica e Programação de Computadores, Programação em Computadores I e Programação em Computadores II. Em 2020/1, foram ofertadas turmas especiais de Lógica de Programação e Programação em Computadores II para os estudantes de Engenharia de Produção.

**Tabela 1 - Desempenho dos estudantes do curso de Engenharia de Produção, Ambiental e Sanitária e Civil nas disciplinas de programação de computadores.**

Disciplinas	Ano	Engenharia de Produção				Engenharia Ambiental e Sanitária				Engenharia Civil			
		Aprovados	Reprovados	Evasão	Abandono	Aprovados	Reprovados	Evasão	Abandono	Aprovados	Reprovados	Evasão	Abandono
Lógica de Programação	2019	24,39%	52,44%	18,29%	4,88%	17,50%	45,00%	35,00%	2,50%	-	-	-	-
	2020	42,86%	24,06%	17,29%	15,79%	43,20%	4,50%	29,50%	22,70%	75,00%	0,00%	16,70%	8,30%
	2021	50,00%	0%	26,92%	23,08%	-	-	-	-	-	-	-	-
Programação em Computadores I	2019	32,00%	38,00%	26,00%	4,00%	50,00%	12,50%	37,50%	0,00%	-	-	-	-
	2020	64,18%	1,49%	19,40%	14,92%	60,00%	0,00%	20,00%	20,00%	-	-	-	-
	2021	45,24%	11,90%	33,33%	10,94%	20,80%	45,80%	20,80%	12,50%	-	-	-	-
Programação em Computadores II	2019	30,00%	26,67%	35,00%	8,33%	-	-	-	-	-	-	-	-
	2020	77,27%	7,58%	7,58%	7,58%	-	-	-	-	-	-	-	-
	2021	46,34%	26,83%	14,63%	12,20%	-	-	-	-	-	-	-	-

Fonte: autoria própria.

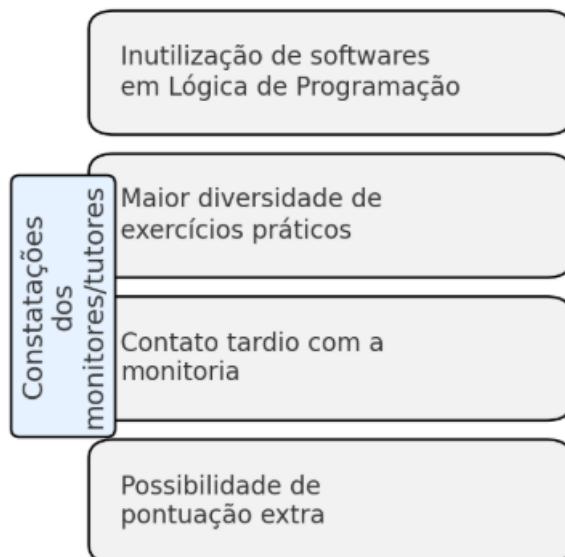
Após a disponibilização dos dados referentes às disciplinas de Lógica e Programação de Computadores I e II pela Secretaria Acadêmica, foi realizado o tratamento das informações e sua organização em formato de tabela. A Tabela 1 foi elaborada considerando os dois semestres de cada ano, de modo que os

percentuais apresentados representassem os eventos registrados ao longo do respectivo ano letivo. Para garantir objetividade, os percentuais identificados em cada semestre foram consolidados e apresentados de forma agregada em um único resultado anual.

### Percepções de monitores e tutores de disciplinas de programação

Com o objetivo de aprofundar a compreensão da realidade estudada, foram coletadas, de forma não estruturada, as percepções de antigos monitores das disciplinas de programação. Esse levantamento permitiu identificar as principais dificuldades enfrentadas pelos alunos, o nível de adesão e frequência às monitorias, bem como a percepção sobre a atribuição de pontuação aos estudantes ingressantes no programa de tutoria como estratégia para estimular maior participação (Figura 4).

Figura 4 - Constatações dos monitores.



Fonte: autoria própria.

Constatou-se a recorrente dificuldade dos alunos em compreender a sintaxe de programação. Esse problema pode ser explicado pelo fato de a disciplina de Lógica de Programação, primeira ofertada no ciclo formativo, não contemplar o uso de softwares que permitam o teste lógico das estruturas construídas. Tal recurso seria fundamental para que os estudantes compreendessem de forma aplicada a linguagem de programação, por meio da demonstração prática do funcionamento dos algoritmos, favorecendo a interpretação e a resolução das questões propostas. Sob a ótica CTS, a ausência de ferramentas computacionais no ensino inicial da programação distancia os alunos de experiências concretas de interação com a tecnologia, comprometendo a construção crítica de saberes necessários à inserção cidadã e profissional em uma sociedade digitalizada.

Os monitores também destacaram a necessidade de maior diversidade de exercícios práticos em sala, capazes de engajar os alunos e evidenciar a utilidade social do conhecimento trabalhado. As três disciplinas ofertadas na área de programação apresentam conteúdos altamente complementares; por isso,

recomenda-se a realização de aulas de revisão no início de cada uma delas, de forma a assegurar a continuidade da aprendizagem e o encadeamento lógico do conhecimento. Essa proposta, na perspectiva CTS, reforça a importância da integração curricular como meio de desenvolver competências alinhadas às demandas tecnológicas e sociais contemporâneas.

Verificou-se, ainda, que a participação em monitoria ocorria, em sua maioria, apenas às vésperas de avaliações. Nesses momentos, muitos estudantes não conseguiam verbalizar suas dúvidas e solicitavam revisões extensas de conteúdos básicos, o que comprometia a progressão do ensino durante os encontros. Essa prática evidencia não apenas a gestão ineficiente do tempo de estudo, mas também a necessidade de estratégias pedagógicas que promovam uma relação mais autônoma e crítica dos alunos com o aprendizado — princípio central da perspectiva CTS, que busca formar sujeitos capazes de analisar e utilizar o conhecimento científico-tecnológico em contextos diversos.

A possibilidade de pontuação extra foi bem avaliada como estratégia de engajamento, mas deve estar articulada a outras ações que estimulem a construção efetiva do conhecimento. É importante que o programa de tutoria forme grupos de alunos comprometidos com a aprendizagem, evitando a participação apenas em função de benefícios avaliativos. Sob o enfoque CTS, incentivos acadêmicos devem contribuir não apenas para ganhos individuais, mas para a construção de comunidades de aprendizagem colaborativas, socialmente relevantes e comprometidas com a apropriação crítica da tecnologia.

Por fim, ressaltou-se que muitos estudantes de engenharia conciliam trabalho e estudo em período noturno, o que limita sua participação presencial nas atividades de monitoria. Nesse contexto, a implementação de um projeto de tutoria remota constitui uma alternativa estratégica, pois amplia o acesso ao conteúdo e ao apoio dos tutores conforme a disponibilidade dos alunos. Essa medida dialoga com a perspectiva CTS ao promover a democratização do acesso ao conhecimento, integrando tecnologia educacional, necessidades sociais e práticas de ensino-aprendizagem, de forma a reduzir desigualdades e favorecer trajetórias acadêmicas mais consistentes.

### Percepções de alunos matriculados nas disciplinas de programação

O perfil dos estudantes de disciplinas de programação foi analisado por meio de questionário anônimo, elaborado no Google Forms® e aplicado de forma voluntária. A Figura X apresenta a distribuição percentual das respostas em relação ao uso de computadores e à experiência prévia em programação.

Os resultados demonstram que 96,7% possuem computador pessoal, 93,3% relatam facilidade de uso e 95,0% dominam ferramentas do pacote Office e navegação na internet. No entanto, 66,7% nunca tiveram contato prévio com programação e 70,0% já reprovaram em disciplinas da área. Entre os reprovados, 40% obtiveram aprovação após um semestre, 13,3% após dois, e aproximadamente 20% somente após três ou quatro semestres.

Adicionalmente, 83,3% afirmaram não frequentar atividades de monitoria, alegando principalmente restrição de tempo (66,7%), busca por meios alternativos de aprendizagem, como livros, apostilas ou vídeos online (50%), e incompatibilidade de horários (36,7%). Os maiores entraves relatados nas disciplinas foram: dificuldade de compreensão do conteúdo em aula, ausência de

---

memorização dos comandos, problemas na interpretação de enunciados e limitações na elaboração da lógica de resolução.

Quanto aos fatores de reprovação, os estudantes destacaram: a ausência de softwares que permitam visualizar a execução do código desde o início do curso, a baixa integração entre teoria e prática, a sobrecarga de disciplinas no semestre, a extensão do conteúdo, a metodologia de ensino e aspectos subjetivos, como desmotivação e falta de interesse.

Sob a perspectiva Ciência-Tecnologia-Sociedade (CTS), os dados confirmam que o acesso a tecnologias digitais não garante, por si só, o desenvolvimento de competências em programação. Há uma lacuna entre o uso cotidiano de tecnologias de consumo (aplicativos, redes e ferramentas de escritório) e a apropriação crítica e criativa desses recursos no âmbito acadêmico e científico. Tal constatação evidencia a necessidade de estratégias pedagógicas que articulem ciência, tecnologia e sociedade, promovendo aprendizagens significativas, contextualizadas e orientadas ao desenvolvimento de habilidades de resolução de problemas.

### Proposta do projeto de tutoria

Com base nos dados de aprovação, reprovação, abandono e evasão, bem como nas percepções de tutores, monitores e alunos matriculados, foi elaborada a proposta de um programa de tutoria. O programa será ofertado em formato remoto, contemplando aulas síncronas e assíncronas, a fim de atender principalmente alunos que conciliam estudos e trabalho. Os tutores produzirão vídeos sob demanda, de acordo com as necessidades de aprendizagem identificadas entre os estudantes.

O projeto, intitulado “Prog X”, contará com três tutores, cada um responsável por uma disciplina: Lógica de Programação de Computadores, Programação de Computadores I e Programação de Computadores II. O estudante poderá participar do programa apenas na disciplina em que estiver matriculado.

Cada disciplina terá um encontro semanal de 1h20, abordando conteúdos vinculados às aulas mais recentes. Os tutores serão acompanhados por um professor coordenador, um coorientador e um colaborador, que fornecerão suporte pedagógico por meio de reuniões periódicas. A dinâmica do programa seguirá as seguintes etapas:

(1) Aplicação de questionário no início do projeto, para entender o perfil pessoal dos alunos ingressantes, e outro ao final, para coletar o feedback sobre a tutoria;

(2) Tutorias com aulas resumo e resolução de exercícios acontecendo em momento exatamente posterior a aula dada na disciplina;

(3) Resolução de exercícios práticos, associados a problemas reais da profissão;

(4) Fóruns, no Moodle, com resolução de questões com dúvidas mais frequentes entre os alunos das listas de exercícios preparatórias das provas das disciplinas (tutor disponibilizará vídeo e gabarito de resolução);

(5) Tutores manterão grupo de WhatsApp com os alunos matriculados na tutoria, para ter contato e comunicação direta;

(6) Atribuir pontuação extra de 5 pontos aos alunos associados à tutoria, mediante a resolução de duas provas (no início do semestre e no fim do semestre);

---

(7) Emissão de certificados aos alunos ingressantes em programa de tutoria e aos tutores do programa (podendo abater em horas complementares).

A organização semanal das atividades da Prog X contempla cinco momentos distintos. No Dia 1, é realizada a tutoria em Lógica de Programação, com duração de 1h20min. No Dia 2, ocorre a tutoria em Programação de Computadores I, também com 1h20min de duração. O Dia 3 é destinado à tutoria em Programação de Computadores II, igualmente com 1h20min semanais. No Dia 4, dedica-se um período de 3 horas ao atendimento de dúvidas dos alunos. Por fim, no Dia 5, são reservadas 3 horas para o planejamento das aulas, correção de atividades, elaboração de gabaritos e produção de vídeos explicativos.

A tutoria de Lógica de Programação e Programação de Computadores I também contemplará os estudantes do curso de Engenharia Civil, uma vez que esse curso possui apenas a disciplina denominada *Lógica e Programação de Computadores*. Os alunos-tutores, na condição de bolsistas, terão dedicação semanal de 10 horas ao projeto.

A tutoria de Lógica de Programação aborda a introdução à Ciência da Computação, contemplando conceitos fundamentais, arquitetura básica do computador e sistema binário. São trabalhados os princípios da lógica de programação, com destaque para o conceito e as diferentes representações de algoritmos, além do estudo de tipos de dados, variáveis, constantes, atribuição e operadores aritméticos, lógicos e relacionais. O conteúdo inclui também a construção de expressões, o uso de comandos básicos e a aplicação de estruturas de controle, como sequenciais, de seleção e de repetição. Complementarmente, são exploradas estruturas de dados homogêneas e heterogêneas, bem como técnicas de desenvolvimento de algoritmos.

A tutoria de Programação de Computadores I aprofunda o estudo dos tipos de dados, variáveis e constantes, enfatizando a estrutura sequencial, operadores, expressões e funções. São desenvolvidos conteúdos relacionados a comandos básicos, estruturas condicionais, estruturas de repetição e estruturas de dados homogêneas e heterogêneas. Essa etapa contempla a aplicação prática de técnicas de desenvolvimento de programas no paradigma da programação estruturada, incluindo o tratamento de arquivos.

Na tutoria de Programação de Computadores II, são apresentados os aspectos conceituais da programação orientada a objetos, destacando-se as diferenças em relação ao paradigma estruturado. São abordados fundamentos como classes, objetos, métodos, mensagens, encapsulamento, herança e polimorfismo, além da modelagem e implementação de hierarquias de classes e o tratamento de exceções. A disciplina também contempla a introdução a banco de dados e sistemas gerenciadores de banco de dados (SGBD), explorando funções, estruturas e fundamentos do modelo relacional. São discutidos o uso da linguagem SQL, voltada tanto à definição de dados (DDL) quanto à sua manipulação (DML), e os principais componentes de interfaces gráficas para integração com bancos de dados. Por fim, são aplicadas técnicas de desenvolvimento de programas orientados a objetos integrados a bancos de dados.

Para apoiar a elaboração das aulas e o estudo autônomo dos alunos, serão utilizados livros de referência disponíveis na biblioteca virtual do IFMG, os quais já constam nas ementas das disciplinas, conforme estabelecido no Projeto Pedagógico de Curso (PPC) dos cursos de graduação do IFMG-GV. Além disso, no decorrer do programa, serão aplicadas três atividades avaliativas, estrategicamente alinhadas aos momentos de realização das provas regulares das

---

disciplinas, de modo a aferir a aprendizagem dos estudantes e reforçar a integração entre tutoria e ensino formal.

O público-alvo do projeto é composto por discentes matriculados nas disciplinas de Lógica de Programação de Computadores, Programação de Computadores I ou Programação de Computadores II, pertencentes aos cursos de Engenharia de Produção, Engenharia Civil e Engenharia Ambiental e Sanitária, do IFMG-GV.

### Análise dos resultados à luz da perspectiva CTS

Os resultados do levantamento evidenciam um paradoxo recorrente no ensino de programação em cursos de engenharia: embora a maioria dos alunos apresente acesso frequente às tecnologias digitais e declare familiaridade com o uso de computadores (96,7%), uma parte significativa enfrenta dificuldades no aprendizado de lógica de programação, refletidas nos elevados índices de reaprovação (70%). Tal constatação reforça a necessidade de repensar estratégias pedagógicas que transcendam o ensino meramente técnico e instrumental.

Nessa direção, a perspectiva CTS (Ciência–Tecnologia–Sociedade) oferece um eixo integrador para compreender as dificuldades relatadas e propor intervenções. A ausência de contato prévio com conteúdos de programação (66,7%) e a baixa procura por monitorias (83,3%), somadas às justificativas de desmotivação, sugerem um distanciamento entre o uso cotidiano da tecnologia (aplicativos, internet, jogos) e a compreensão crítica de sua estrutura e funcionamento. Segundo Auler e Delizoicov (2001), a abordagem CTS busca superar essa lacuna ao promover uma aprendizagem que articule dimensões técnicas, sociais e culturais do conhecimento, aproximando a tecnologia da realidade dos estudantes.

Além disso, a dificuldade relatada pelos discentes em assimilar teoria e prática, bem como em construir lógica de programação, evidencia a pertinência de metodologias que explorem o aprendizado ativo e contextualizado. Santos e Mortimer (2000) destacam que o ensino orientado por CTS deve valorizar problemas reais e estimular a reflexão crítica, tornando o conteúdo mais significativo. Isso dialoga com a sugestão dos alunos de utilizar softwares que favoreçam a visualização do funcionamento dos códigos como forma de primeiro contato com a programação.

O dado de que 50% dos estudantes buscam materiais externos (livros, apostilas, vídeos do YouTube) também reforça a importância de integrar a tutoria e as disciplinas curriculares a práticas interdisciplinares e multimodais, ampliando o repertório didático e promovendo autonomia. Como argumenta Bazzo (2011), iniciativas CTS devem ir além da simples transmissão de conteúdos, estimulando a formação crítica do engenheiro diante da crescente inserção das tecnologias digitais na sociedade contemporânea.

Assim, os achados do trabalho indicam que iniciativas de tutoria alinhadas à perspectiva CTS podem contribuir não apenas para reduzir índices de reaprovação, mas também para formar profissionais capazes de compreender a programação como ferramenta cultural, social e tecnológica, ampliando sua relevância acadêmica e social.

O presente estudo teve como objetivo levantar informações sobre aprovação, reprovação, evasão e abandono de alunos dos cursos de Engenharia em disciplinas de Programação de Computadores ofertadas pelo campus, identificando também os principais fatores relacionados. Para isso, foram analisados dados disponibilizados pela Secretaria de Registro Acadêmico, além das percepções de monitores, tutores e estudantes matriculados. A partir desse diagnóstico, foi proposta a criação e implementação do projeto de ensino “Prog X”, voltado à consolidação dos conhecimentos básicos necessários para que o aluno comprehenda o conteúdo e obtenha aprovação na disciplina em curso.

As informações levantadas foram decisivas para orientar a proposta e traçar a melhor estratégia de desenvolvimento do projeto. Considerando a alta incompatibilidade de horários e a baixa assiduidade identificada, optou-se por um formato remoto e flexível, no qual os tutores acompanham e se comunicam continuamente com os alunos, conforme a demanda. O projeto, portanto, busca superar os obstáculos de aprendizagem e favorecer a permanência acadêmica.

Os resultados obtidos mostraram que 70% dos alunos já haviam reprovado pelo menos uma vez em disciplinas de programação, e que a maioria necessitou de um a três semestres para concluir tais componentes curriculares. Esses achados revelam um quadro crítico, que ultrapassa as dificuldades individuais de aprendizagem e aponta para uma problemática mais ampla no processo formativo.

Sob a ótica CTS (Ciência-Tecnologia-Sociedade), esses resultados evidenciam a necessidade de repensar a formação em Engenharia em diálogo com as demandas da sociedade digital e do setor produtivo. O elevado índice de reprovação em disciplinas basilares de programação indica uma lacuna institucional na preparação de profissionais capazes de atuar em um mundo cada vez mais dependente de competências digitais. A abordagem CTS busca justamente conectar o processo educativo às transformações científicas e tecnológicas, ampliando seu impacto social. Assim, o “Prog X” não deve ser visto apenas como uma solução pedagógica pontual, mas como uma estratégia de caráter social, capaz de contribuir para a permanência estudantil, para a equidade no acesso ao conhecimento científico e para a formação de profissionais em áreas estratégicas para o desenvolvimento tecnológico do país.

Para trabalhos futuros, recomenda-se avaliar o desempenho dos estudantes em Lógica e Programação de Computadores I e II após a implementação do projeto de tutoria, analisando taxas de aprovação, reprovação, evasão e abandono. Além disso, sugere-se aplicar questionários qualitativos e quantitativos ao final dos semestres para identificar percepções dos alunos e aspectos a serem aprimorados. Ressalta-se, como limitação do presente estudo, a ausência de informações detalhadas sobre a base de ensino anterior à graduação, o que poderia esclarecer parte das dificuldades encontradas.

Por fim, destaca-se a importância de institucionalizar o projeto “Prog X”, assegurando sua oferta contínua e semestral, com envolvimento de diferentes professores e constante aprimoramento metodológico. Espera-se que, com isso, o processo de aprendizagem em Programação se torne progressivo e consistente, refletindo-se em maior compreensão dos conteúdos, melhor desempenho acadêmico e, sobretudo, em uma formação mais alinhada às demandas científicas, tecnológicas e sociais contemporâneas.

# Educating Engineers in the Digital Era: Remote Tutoring in Programming from an STS Perspective

## ABSTRACT

The subjects of Logic Programming and Computer Programming have historically shown high failure and dropout rates in Engineering courses. This study, conducted at IFMG – Governador Valadares campus, identified that around 70% of students failed at least once in these courses and, on average, required one to three additional semesters to achieve approval. Based on the analysis of students' profiles and the main reported difficulties, the Prog X project was developed: a remote tutoring program designed to provide personalized support through synchronous and asynchronous sessions, practical exercise resolution, and continuous follow-up by tutors. The objective is to consolidate basic programming knowledge, enhance learning, reduce failure, dropout, and withdrawal rates, as well as strengthen student retention and academic performance in Engineering programs. Distinctively, the proposal is framed under the CTS (Science-Technology-Society) perspective, linking programming learning to the development of critical and reflective skills essential for training engineers in the digital era.

**KEYWORDS:** Programming Learning. University Dropout. Pedagogical Support. Remote Teaching. CTS Approach.

## REFERÊNCIAS

Auler, D., & Bazzo, W. A. (2001). Reflections on the implementation of the STS movement in the Brazilian educational context. *Ciência & Educação* (Bauru), 7(1), 1–13. <https://doi.org/10.1590/S1516-73132001000100001>

Bazzo, W. A. (1998). Sc

Aureliano, O. C. V., Tadesco, R. A. C. P., & Giraffa, M. M. L. (2016). Desafios e oportunidades aos processos de ensino e de aprendizagem de programação para iniciantes. XXXVI Congresso da Sociedade Brasileira de Computação. Porto Alegre. <https://doi.org/10.5753/wei.2016.9650>

Barbosa, J. C. (2004). A contextualização e a modelagem na educação matemática do ensino médio. In Anais do VIII Encontro Nacional de Educação Matemática. Recife: SBEM. [https://www.academia.edu/4561571/A\\_contextualizacao\\_e\\_a\\_modelagem\\_na\\_educacao\\_matematica\\_do\\_EM](https://www.academia.edu/4561571/A_contextualizacao_e_a_modelagem_na_educacao_matematica_do_EM)

Bazzo, W. A., Pereira, L. T. V., & von Linsingen, I. (2000). Technological education: approaches to engineering teaching. UFSC Press.

Bazzo, W. A., von Linsingen, I., & Pereira, L. T. V. (Eds.). (2003). Introduction to STS studies (Science, Technology and Society). OEI.

Borges, M. A. (2000). Avaliação de uma metodologia alternativa para a aprendizagem de programação. <http://www.niee.ufrgs.br/eventos/SBC/2000/pdf/wei/relatos/selecionados/wei06.pdf>

Bosse, Y., & Gerosa, M. A. (2015). Reprovações e trancamentos nas disciplinas de introdução à programação da Universidade de São Paulo: Um estudo preliminar. In Anais do XXIII Workshop sobre Educação em Computação (WEI) (pp. 426–435). Sociedade Brasileira de Computação. <https://doi.org/10.5753/wei.2015.10259>

Campos, C. (2004). Método de análise de conteúdo: ferramenta para a análise de dados qualitativos no campo da saúde. *Revista Brasileira de Enfermagem*. <https://www.scielo.br/j/reben/a/wBbjs9fZBDrM3c3x4bDd3rc/abstract/?lang=pt>

Corrêa, L. F., & Bazzo, W. A. (2017a). Contributions of the Science, Technology and Society approach to the humanization of teaching work. *Revista Contexto e Educação*, 32(102), 57–80. <https://doi.org/10.21527/2179-1309.2017.102.57-80>

Corrêa, L. F., & Bazzo, W. A. (2017b). Contributions of the Science, Technology and Society approach to the humanization of teaching work. *Revista Eletrônica Engenharia Viva*, 1(2), 41–55. <https://doi.org/10.25112/reev.v1i2.1334>

Damasceno, I., & Carneiro, M. (2018). Panorama da evasão no curso de sistemas de informação da Universidade Federal de Uberlândia: Um estudo preliminar. *Simpósio Brasileiro de Informática na Educação (SBIE)*, 29(1), 1766–2018. <https://br-ie.org/pub/index.php/sbie/article/view/8147>

D'Angelo, F. (2021). Haverá um colapso nas áreas de TI e inovação no pós-pandemia. Canaltech. <https://canaltech.com.br/mercado/havera-um-colapso-nas-areas-de-ti-e-inovacao-no-pos-pandemia-entenda/>

Dejavite, L. M., Guedes, K., Beleti Jr., C. R., Molin Filho, R. G. D., & Santiago Jr., R. M. (2017). O desempenho dos estudantes nas disciplinas de algoritmos e programação de computadores: Um retrato dos cursos de engenharia da UFPR. XLV Congresso Brasileiro de Educação em Engenharia. <https://www.br-ie.org/pub/index.php/wcbie/article/download/8943/6492>

Ferreira, M. L. A., de Souza, C. G., & Spritzer, I. M. P. A. (2017). Contributions of the STS approach to engineering education in Brazil. *Revista Espacios*, 38(20), 33–46.

Freire, L., Coutinho, J., Lima, V., & Lima, N. (2019). Uma proposta de encontros de tutoria baseada em metodologias ativas para disciplinas de programação introdutória. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 8(1), 298. <https://www.br-ie.org/pub/index.php/wcbie/article/view/8971>

Gil, A. C. (2007). Como elaborar projetos de pesquisa. São Paulo: Atlas.

Giraffa, M. M., & da Costa, M. M. (2013). Evasão na disciplina de algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno. *III Conferencia Latinoamericana sobre el abandono en la Educación Superior*. Recuperado de <https://repositorio.pucrs.br/dspace/handle/10923/8684>

Garlet, D., Bigolin, N. M., & Silveira, S. R. (2018). Ensino de programação de computadores na educação básica: um estudo de caso. *Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica*, 8(1), 1–15. Recuperado de <https://revista.unifacef.com.br/index.php/resiget/article/view/1542>

Holanda, W., Coutinho, J., & Fontes, L. (2018). Uma intervenção metodológica para auxiliar a aprendizagem de programação introdutória: um estudo experimental. *Anais do Workshop sobre Educação em Computação (WCBIE)*, 699–708. <https://doi.org/10.5753/cbie.wcbie.2018.699>

Krzyzanowski, L., et al. (2019). Ensino de programação: um estudo preliminar nos cursos de licenciatura em computação no Brasil. *VIII Congresso Brasileiro de Informática na Educação (CBIE)*. Recuperado de <https://www.br-ie.org/pub/index.php/wcbie/article/view/8943/6492>

Lakatos, E. M., & Marconi, M. A. (2001). Fundamentos de metodologia científica (4<sup>a</sup> ed.). São Paulo: Atlas.

Lima, M. (2018). Concepções docentes acerca do ensino de programação de computadores no ensino superior. *Revista Teoria e Prática da Educação*, 21(3), 43–57. Recuperado de <https://periodicos.uem.br/ojs/index.php/TeorPratEduc/article/view/45599>

Linsingen, I. von. (2015). The STS approach in science education in Brazil: Challenges and potentialities. *Linhas Críticas*, 21(45), 13–31. <https://periodicos.unb.br/index.php/linhascriticas/article/view/4536>

Marcussi, L., et al. (2016). Pesquisa no ensino de algoritmos e programação nas engenharias: estudo e resultados preliminares. SIMEPROD – Simpósio de Engenharia de Produção. Universidade Federal do Paraná. Recuperado de <http://www.dep.uem.br/gdct/index.php/simeprod/article/view/988>

Mendes, A. J. N. (2002). Software educativo para apoio à aprendizagem de lógica de programação. VIII Taller Internacional de Software Educativo (TISE), Santiago, Chile. Recuperado de <http://www.seer.ufrgs.br/index.php/renote/article/view/14284/8203>

Miranda, S. de. (2002). No fascínio do jogo, a alegria de aprender. *Linhas Críticas*, 8(14), 21–34. Disponível em: <http://www.gestaoesaude.unb.br/index.php/linhascriticas/article/view/6493/5248>

Pelissari, L. (2023). STS education and critical formation: Reflections and practices. *Revista Brasileira de Educação*, 28, e280090.

Oliveira, M. F. (2011). Metodologia científica: Um manual para realização de pesquisas em administração. Catalão: Universidade Federal de Goiás. Disponível em: [https://files.cercomp.ufg.br/weby/up/150/o/Manual\\_de\\_metodologia\\_cientifica.pdf](https://files.cercomp.ufg.br/weby/up/150/o/Manual_de_metodologia_cientifica.pdf)

Pascoal, T. A., Brito, D. M., & Rêgo, T. G. (2015). Uma abordagem para a previsão de desempenho de alunos de Computação em disciplinas de programação. Anais do Workshop de Informática na Escola, 454–458. Disponível em: <http://www.tise.cl/volumen11/TISE2015/454-458.pdf>

Paulo, L. (s.d.). Metodologia da pesquisa aplicada à tecnologia. São Paulo: SENAI-SP Editora. Disponível em: [https://books.google.com/books/about/Metodologia\\_da\\_pesquisa\\_aplicada\\_%C3%A0\\_t.html?id=](https://books.google.com/books/about/Metodologia_da_pesquisa_aplicada_%C3%A0_t.html?id=)

Rapkiewicz, C. E., Wives, L. K., Seara, I. C., Lima, J. V., & Rigo, S. J. (2006). Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. RENOTE – Revista Novas Tecnologias na Educação, 4(2), 1–11. Disponível em: [https://seer.ufrgs.br/renote/article/view/14284/8203](http://seer.ufrgs.br/renote/article/view/14284/8203)

Reino, L. R. A. C., Medeiros, E. L., Paiva, L. R., & Fernandes, C. C. (2015). Análise das causas da evasão na educação a distância em uma instituição federal de ensino superior. In Anais do Simpósio Brasileiro de Informática na Educação (SBIE) (pp. 91–100). Porto Alegre: SBC. <https://doi.org/10.5753/cbie.sbie.2015.91>

Rodrigues, R., Medeiros, F. P. A., & Gomes, A. S. (2013). Modelo de regressão linear aplicado à previsão de desempenho de estudantes em ambiente de aprendizagem.

In Anais do Simpósio Brasileiro de Informática na Educação (SBIE) (pp. 607–616). Campinas: SBC. <https://doi.org/10.5753/cbie.sbie.2013.607>

Rodrigues, F. M. (2017). Appropriations of the STS approach in teacher education in Science [Master's thesis, Federal University of Minas Gerais]. UFMG Institutional Repository. <https://repositorio.ufmg.br/handle/1843/BUOS-AU9KZL>

Salazar, R., Odakura, V., & Barvinski, C. (2015). Scratch no ensino superior: Motivação. In Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE) (pp. 1–10). Maceió: SBC. Disponível em: <http://www.brie.org/pub/index.php/sbie/article/view/5470/3829>

Santos, C. (2008). Manual de autoaprendizagem: Estatística descritiva (3<sup>a</sup> ed.). Lisboa: Sílabo.

Santos, W. L. P., & Mortimer, E. F. (2000). An analysis of theoretical assumptions of the STS (Science-Technology-Society) approach in the context of Brazilian education. *Ensaio: Pesquisa em Educação em Ciências*, 2(2), 1–16. <https://doi.org/10.1590/1983-21172000020201>

Scaico, P., et al. (2012, julho). Relato de um modelo de tutoria para programação baseado em experiências com ingressantes de um curso de Licenciatura em Computação. Anais dos Workshops do XX Congresso da Sociedade Brasileira de Computação (SBC 2012), Curitiba. Sociedade Brasileira de Computação (SBC). Disponível em: <http://www.ufpr.br/> (Acesso em: 9 maio 2023).

Sena, V., & Granato, L. (2021). 260.000 vagas sem dono: um raio-x das vagas mais quentes agora e no futuro. Exame. Disponível em: <https://exame.com/carreira/260-000-vagas-de-trabalho-sem-dono-conheca-o-setor-que-ganhou-forca-com-a-pandemia/> (Acesso em: 10 set. 2023).

Silva, J. (2017). Ensino de Programação para alunos do Ensino Básico: Um levantamento das pesquisas realizadas no Brasil [Dissertação de mestrado, Universidade Federal da Paraíba]. Repositório UFPB. Disponível em: <https://repositorio.ufpb.br/jspui/handle/123456789/3328> (Acesso em: 31 ago. 2023).

Silva, N. D., Brito, R. J., & Vaz, P. A. N. (2019). Lógica de programação: Dificuldades de ensino-aprendizagem, métodos e ferramentas computacionais. Anais do STI-SIC. Universidade Estadual de Goiás. Disponível em: [https://anais.ueg.br/index.php/sti\\_sic/article/download/13982/10917](https://anais.ueg.br/index.php/sti_sic/article/download/13982/10917) (Acesso em: 31 ago. 2023).

Tumelero, N. (2019). Pesquisa exploratória: conceitos, características e aplicação em 4 passos. Blog Mettzer. Disponível em: <https://blog.mettzer.com/pesquisa-exploratoria/> (Acesso em: 11 set. 2023).

Yin, R. K. (2001). Estudo de caso: planejamento e métodos (5<sup>a</sup> ed.). Porto Alegre: Bookman.

Zappellini, M., & Feuerschütte, S. (2015). O uso da triangulação na pesquisa científica brasileira em administração. *Revista de Administração, Ensino e Pesquisa (RAEP)*, 16(2), 241–273. <https://doi.org/10.13058/raep.2015.v16n2.238>

**ANEXO I - Questionamentos aos antigos tutores e monitores**

1. Quais as dificuldades que você identificou ser mais comum entre os alunos?
2. Qual era a adesão de participação?
3. Os alunos que participavam das monitorias eram sempre os mesmos? Ou havia diversificação?
4. Você acha que, se houver pontuação extra a quem participará do "PROG X", tendo que fazer provas para isso, os alunos participariam do projeto?

**ANEXO II - Questionário aplicado junto aos alunos**

1. Qual o seu período de ingresso no IFMG-GV?

2015/1  
 2015/2  
 2016/1  
 2016/2  
 2017/1  
 2017/2  
 2018/1  
 2018/2  
 2019/1  
 2019/2  
 2020/1  
 2020/2

2. Você tem familiaridade com o uso de computadores?

Sim  
 Não

3. Possui computador pessoal?

Sim  
 Não

4. Pra você, é fácil usar o computador?

Sim  
 Não  
 Talvez

5. Quais programas você mais utiliza no computador? (Você pode selecionar mais de uma alternativa)

Pacote Office  
 Internet  
 Aplicativos de Música  
 Edição de vídeo e imagens  
 Jogos  
 Outros

6. Antes de cursar disciplinas de programação no IFMG-GV, você já teve algum tipo de contato com conteúdos sobre programação de computadores?

Sim  
 Não

7. Já teve reprovação em disciplinas da área de programação (Lógica de Programação, Programação de Computadores I e/ou Programação de Computadores II)?

Sim  
 Não

7.1. Quais os principais motivos de reprovação e/ou abandono nas disciplinas de programação?

8. Quanto tempo demorou entre a conclusão da disciplina de Lógica de Programação e Programação de Computadores I?

1 semestre  
 2 semestres  
 3 semestres  
 4 semestres  
 5 semestres  
 6 semestres ou mais  
 Não se aplica

---

9. Quanto tempo demorou entre a conclusão da disciplina de Programação de Computadores I e o ingresso na disciplina de Programação de Computadores II?

- 1 semestre
- 2 semestres
- 3 semestres
- 4 semestres
- 5 semestres
- 6 semestres ou mais
- Não se aplica

10. Caso nas questões 8 e/ou 9 você tenha marcado alguma opção de 3 semestres ou mais, justifique o que aconteceu.

11. Você costumava ir em monitorias de disciplinas de programação?

12. Qual frequência nas monitorias?

- Em véspera de prova
- Eu era frequente
- Eu não frequentava

13. Se você não tinha o costume de ir às monitorias, aponte os motivos:

- Restrição de tempo
- Horário de monitoria não compatível
- Não tinha certeza das matérias que tinha dificuldade
- Outros colegas me ajudavam
- Tentava aprender a matéria de outra forma (por livros, apostilas ou vídeos do Youtube)
- Outros

14. Quais as suas principais dificuldades em disciplinas de programação? (Você pode selecionar mais de uma alternativa)

- Entender o que a questão pedia
- Elaborar a lógica para resolução da questão
- Perda de aulas acabava impactando no entendimento do restante do conteúdo
- Conseguir recordar todos os comandos necessários para programação
- Dificuldade de entendimento nas aulas

- Não conseguia acompanhar a resolução de questões práticas
- Não conseguia se recordar de assuntos abordados em disciplinas anteriores
- Falta de interesse na disciplina
- Outros

15. Quais as suas sugestões para melhoria de ensino e aprendizagem nas disciplinas de programação?

**Recebido:** 27/02/2024  
**Aprovado:** 24/08/2025  
**DOI:** 10.3895/rts.v21n67.18209

**Como citar:**

DUARTE, Riam Ferreira Lopes; LONGHINI, Tatielle Menolli; TOLEDO, Bruno de Souza. Formando engenheiros na era digital: tutoria remota em programação sob a perspectiva CTS. **Rev. Tecnol. Soc.**, Curitiba, v. 21, n. 67, p.50-74, out./dez, 2025. Disponível em:

<https://periodicos.utfpr.edu.br/rts/article/view/18209>

Acesso em: XXX.

**Correspondência:**

**Direito autoral:** Este artigo está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.

