

## Coletânea de uma Década de Ensino de Programação para Estudantes da Rede Pública no Projeto Introcomp

*Title: A Decade's Collection of Teaching Programming to the Public School System in the Introcomp Project*

Marco A. M. Oliari  
UFES  
marcooliari@gmail.com

José J. M. Uliana  
UFES  
ulianamjorge@gmail.com

Beatriz M. S. Maia  
UFES  
beatrizmatiasmaia@gmail.com

Mirelly M. da Silva  
UFES  
mirellymicaella@gmail.com

Sophie D. Gama  
UFES  
sophiedilhon@yahoo.com.br

Thiago T. Paiva  
UFES  
thiago.t\_27@hotmail.com

Roberta L. Gomes  
UFES  
rgomes@inf.ufes.br

Patrícia D. Costa  
UFES  
pdcosta@inf.ufes.br

Rodrigo L. Guimarães  
UFES  
rlaiola@inf.ufes.br

### Resumo

*Este artigo apresenta os 10 anos de trajetória do projeto de extensão universitária Introcomp. O projeto visa introduzir a programação de computadores para estudantes de ensino médio da rede pública. O artigo apresenta uma reflexão sobre a experiência da equipe do Introcomp na aplicação de diferentes abordagens para facilitar a aprendizagem de programação para seu público-alvo no decorrer desses anos – passando da linguagem C para Python, de aulas tradicionais a aulas totalmente interativas, chegando em 2020 ao modelo totalmente remoto devido à pandemia de Covid-19. Com isso, o artigo mostra como as lições aprendidas têm ajudado a equipe a repensar o uso de metodologias, tecnologias e plataformas digitais no ensino de programação para jovens do ensino médio em tempos de pandemia. A contribuição deste trabalho traz um avanço para o estado da arte e tem o potencial de ampliar o entendimento da comunidade sobre intervenções de educação em computação direcionadas principalmente ao público jovem.*

**Palavras-chave:** *Ensino de programação; Programação no ensino médio; Ensino remoto; Computação interativa; Aprendizado baseado em problema - ABP; Correção automática; Live streaming.*

### Abstract

*This article presents the 10-year trajectory of the university extension project Introcomp. The project aims to introduce computer programming to students from state high school. The article reflects on the experience of the Introcomp team in applying different approaches to promote computer programming learning to its target audience over these years – moving from the C language to Python, from traditional lessons to fully interactive lessons, arriving in 2020 to the fully remote education model due to the Covid-19 pandemic. Hence, the article shows how the lessons learned have helped the team to rethink the use of methodologies, technologies and digital platforms in teaching programming to high school youth in pandemic times. The contribution of this work brings a breakthrough to the state of the art and has the potential to broaden the community's understanding of computer education interventions aimed primarily at young audiences.*

**Keywords:** *Computer Programming education; Computer programming in High School; Distance education; Interactive computing; Problem Based Learning - PBL; Online judge; Live streaming.*

Cite as: Oliari, M. A. M., Uliana, J. J. M., Maia, B. M. S., Silva, M. M., Gama, S. D., Paiva, T. T., Gomes, R. L., Costa, P. D., & Guimarães, R. L. (2021). Coletânea de uma Década de Ensino de Programação para Estudantes da Rede Pública no Projeto Introcomp. *Revista Brasileira de Informática na Educação*, 29, 1202-1231. DOI: 10.5753/rbie.2021.2125.

## 1 Introdução

O avanço e a popularização da tecnologia em praticamente todos os setores da sociedade vêm exigindo o desenvolvimento de novas habilidades nos profissionais e cidadãos de uma forma geral. Conseqüentemente, cursos de programação básica de computadores têm se tornado cada vez mais importantes para o desenvolvimento de habilidades como resolução de problemas e raciocínio lógico em nossos jovens (Gerhardt et al., 2018). Com isso em mente, inúmeros trabalhos investigaram aspectos inerentes ao impacto de linguagens de programação (Aureliano & Tedesco, 2012), (Nikula, Sajaniemi, Tedre, & Wray, 2007), de estratégias/metodologias de ensino e aprendizagem (Looi & Seyal, 2014), (Müller, Silveira, & de Souza, 2018), (Odekirk-Hash & Zachary, 2001) e de ferramentas de apoio à programação (Alencar, Moura, & Bitencourt, 2013), (Kelleher & Pausch, 2005), (Gerhardt et al., 2018), para citar alguns, visando tornar o ato de programar mais acessível para programadores iniciantes (usuários tentando aprender uma linguagem de programação).

Dentro desse contexto, este artigo relata a trajetória de dez anos de experiência do projeto Introcomp<sup>1</sup> na tentativa de diminuir as barreiras para programadores iniciantes por meio de um curso de Introdução à Computação oferecido para estudantes do ensino médio da rede pública. Idealizada por estudantes da Ufes em 2010, a edição piloto do Introcomp ocorreu em 2011, em parceria com apenas uma escola estadual do Espírito Santo. A partir de 2013, o curso passou a abranger a maioria das escolas da rede pública da Grande Vitória, incluindo os institutos federais. Por meio de uma parceria firmada em 2015 com a Secretaria de Educação do Espírito Santo (SEDU), o projeto obteve ainda mais visibilidade, chegando a receber naquele ano 1249 inscrições (Figura 2), totalizando até 2019 mais de 4 mil inscrições.

No decorrer desse período, o Introcomp passou por diversas reformulações, conforme resumido na Figura 1, sendo o estudante sempre o foco. A principal contribuição deste trabalho é um relato detalhado da experiência no ensino de programação para nosso público-alvo, estudantes do ensino médio da rede pública, compartilhando as lições aprendidas nessa trajetória, e discutindo o andamento do uso de plataformas digitais e do ensino remoto visando contornar os desafios que surgiram com a pandemia.

O restante do artigo está estruturado da seguinte forma: Seção 2 contextualiza e posiciona a nossa contribuição à luz de outros trabalhos na área; Seção 3 relata experiências e compartilha lições aprendidas durante o período pré-pandêmico; Seção 4 discorre sobre as estratégias de ensino adotadas pelo projeto; Seção 5 discute os desafios impostos pela pandemia e as soluções encontradas pela equipe para superá-los; a Seção 6 relata a experiência do processo de seleção dos candidatos desde a divulgação do curso até o desenvolvimento de um sistema remoto de provas; a Seção 7 descreve a experiência da equipe ao pôr o planejamento do ensino remoto em prática durante a pandemia; a Seção 8 pondera sobre os desafios enfrentados, os impactos do projeto nos estudantes e do ensino remoto sobre o Introcomp; e, finalmente, a Seção 9 faz uma reflexão acerca dos relatos feitos neste artigo, pensando no futuro do projeto.

---

<sup>1</sup><https://introcomp.ufes.br/>

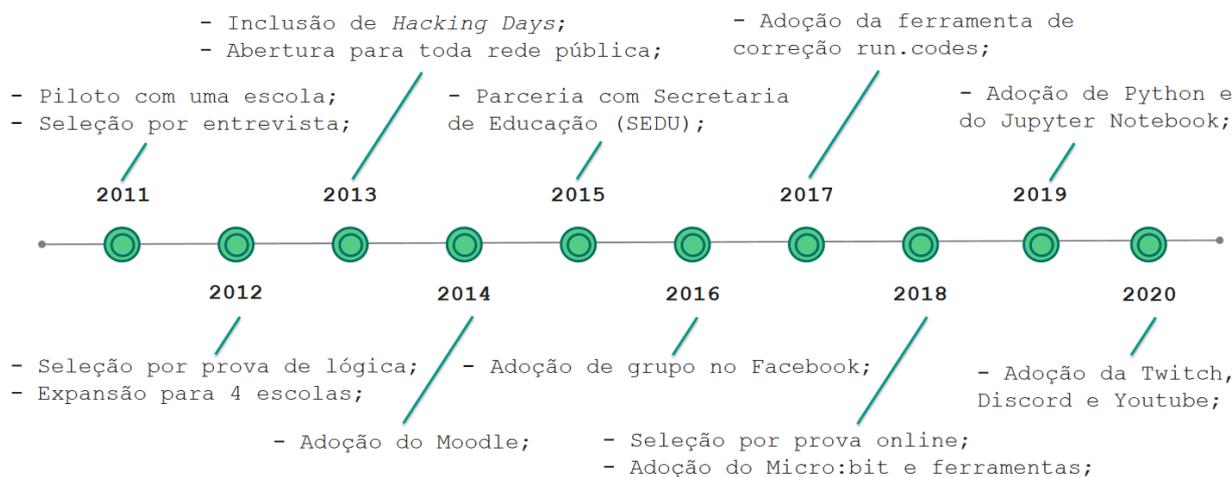


Figura 1: Trajetória do Introcomp ao longo dos anos.

## 2 Trabalhos Relacionados

Na literatura encontramos estudos que abordam os mais variados aspectos relacionados ao ensino de programação a estudantes do ensino médio. Aqui no Brasil, Pantaleão, Amaral, & Braga e Silva (2017) descrevem a experiência de uso de uma ferramenta de simulação de batalha entre robôs, em que se joga programando em Java, para o ensino de algoritmos e programação. Os autores observaram que um ambiente lúdico de ensino aliado à realização competições contribuiu significativamente para a motivação dos estudantes do ensino médio em aprender e superar desafios. Ferreira et al. (2016) descrevem suas experiências em um curso de iniciação à programação de computadores voltado para jovens e adultos da região metropolitana de Salvador sem contato prévio com programação. Segundo os autores, o uso de uma metodologia de ensino com atividades lúdicas ao longo do curso e a inserção de um projeto final de desenvolvimento de um jogo contribuíram para despertar o raciocínio lógico e computacional, a criatividade e mesmo atrair potenciais interessados para os cursos de graduação em computação. Farias, Oliveira, & Silva (2018) relatam a experiência de um projeto de extensão que promoveu um curso de Introdução à Lógica de Programação para os estudantes ingressantes no Curso Técnico de Informática utilizando a ferramenta Scratch. Apesar de não terem sido observadas melhorias significativas na taxa de aprovação, a abordagem se mostrou positiva para desenvolvimento do raciocínio lógico dos estudantes do ensino médio.

Kemppainen, Fraley, Hamlin, & Hein (2016) investigaram a importância do acesso e familiaridade com a computação para o sucesso de estudantes do ensino médio que ingressam nos cursos de engenharia. Para isso, os autores buscaram entender como as diferenças entre os estudantes afetam a dinâmica de sala de aula e podem ser utilizadas como base para um planejamento mais assertivo pelos docentes. Na mesma linha, Hagan & Markham (2000) apresentam um estudo sugerindo que os estudantes experientes em pelo menos uma linguagem de programação no início de um curso de graduação têm desempenho significativamente melhor na avaliação do que aqueles

sem nenhum conhecimento prévio, e que quanto mais linguagens um estudante tem experiência, melhor tende a ser o desempenho. Por sua vez, Guzdial (2009) compartilha as experiências de professores do Instituto de Tecnologia da Geórgia (Georgia Tech) ao longo de uma década de ensino de computação para os estudantes que acabaram de sair do ensino médio para entrar na graduação, mesmo os com ênfase em outras áreas que não a de computação. Apesar de desafiador e caro (por ex., por demandar o envolvimento de professores de outros departamentos que entendem as necessidades de seus estudantes em disciplinas posteriores e em suas futuras profissões), o ensino de computação mais contextualizado, por meio de exemplos de domínios específicos, foi um passo significativo para tornar os cursos introdutórios de computação um sucesso.

Já com foco no ensino superior, Silva, Rivero, & Santos (2021) apresentam um jogo do gênero *Escape the Room* cuja utilização visa apoiar o processo de ensino-aprendizagem de conceitos de lógica de programação para ingressantes. O *feedback* do público-alvo foi positivo, destacando que aspectos como acessibilidade e facilidade de aprender a jogar podem auxiliar no processo inicial de ensino-aprendizagem de programação. Santos, Sousa, Raiol, Cerqueira, & Bezerra (2019) relatam resultados positivos na avaliação de um método pedagógico que englobou aulas, exercícios, atividades lúdicas, avaliações e desafios no ensino de conceitos de programação e robótica para ingressantes do ensino superior, enquanto Coutinho, Bonates, & Moreira (2018) descrevem a experiência de utilização de uma abordagem baseada no desenvolvimento de jogos digitais para o ensino de lógica de programação a estudantes iniciantes de um curso de graduação. Os resultados desse último sugerem que a estratégia utilizada conseguiu motivar os estudantes através da adequação das atividades práticas propostas em aulas a aspectos que envolviam multimídia. Contudo, os autores pontuam o desafio de conciliar as expectativas dos estudantes, essencialmente nos jogos, com os fundamentos de programação necessários ao longo do curso.

O cenário inesperado e incerto da pandemia provocada pelo novo coronavírus tem impactado significativamente não só as estratégias de ensino-aprendizagem (Deus, Fioravanti, Oliveira, & Barbosa, 2020; A. Santana, Costa, & Castro, 2020), incluindo de programação, mas também tido efeitos sociais e afetivos em professores e alunos (Araujo, Amato, Martins, Eliseo, & Silveira, 2020; Menezes & Francisco, 2020). Nessa perspectiva, Neri & Osorio (2021) analisam dados do PNAD COVID/IBGE referentes ao segundo semestre de 2020 com o intuito de obter um índice que reflete se alunos do grupo de 6 a 15 anos estão matriculados, recebendo atividades escolares para estudar de casa e quantos dias e horas da semana se dedicam ao aprendizado à distância. O estudo revela que, além de diferenças regionais e socioeconômicas, houve uma queda significativa no tempo médio de estudo por aluno e um crescimento da evasão, em particular, do grupo de 5 a 9 anos ao longo do ano de 2020. Já Nunes (2021) buscou compreender as razões pelas quais estudantes de cursos superiores do IFFluminense - Campus de Cabo Frio decidiram abandonar as disciplinas após a implantação do ensino remoto durante a pandemia do COVID-19. A autora ressalta que, além dos problemas psicológicos e emocionais, a sobrecarga de trabalho, as tarefas domésticas e a falta de uma rotina de estudos também contribuem para a evasão dos estudantes.

Reconhecemos a importância das contribuições discutidas nesta seção (e de inúmeras outras existentes na literatura) e o trabalho descrito neste artigo compartilha vários aspectos presentes nos trabalhos aqui citados. Mas um grande diferencial trata-se da consideração de diferentes estratégias e ferramentas ao longo dos anos, inclusive durante a pandemia, bem como de uma reflexão sobre o impacto dessas na motivação e engajamento de um público-alvo específico, uma vez que o projeto foca exclusivamente no ensino de programação no contexto da rede pública de ensino mé-

dio, visando com isso promover mais oportunidades a estudantes que potencialmente apresentem maior vulnerabilidade social (Costa et al., 2020). Nas próximas seções descrevemos as intervenções e instrumentos pedagógicos utilizados, e refletimos sobre as nossas experiências, lições aprendidas, resultados alcançados e perspectivas após uma década de realização dessa iniciativa.

### 3 Trajetória e Lições Aprendidas Pré-Pandemia

O Introcomp (Introdução à Computação) é uma iniciativa que oferece um curso introdutório de programação de computadores a estudantes do ensino médio da rede pública. O projeto surgiu com o objetivo de democratizar o acesso à tecnologia, aumentar a procura pelos cursos de computação e, conseqüentemente, atenuar o déficit de profissionais na área (Gerhardt et al., 2018).

O curso é oferecido anualmente durante o segundo semestre, sendo que no primeiro semestre ocorre o planejamento, divulgação, inscrições e processo seletivo. Até 2019, o curso era realizado presencialmente aos sábados na Ufes, tendo uma duração total de aproximadamente 5 meses. O número de estudantes atendidos anualmente também foi aumentando no decorrer dos anos, chegando a 80 estudantes (divididos em duas turmas). Esse número foi definido baseado nos bolsistas, voluntários e nas limitações de infraestrutura (laboratórios).

De 2011 a 2018, o Introcomp focou no ensino de programação utilizando a linguagem C. O conteúdo era fundamentado no livro “Introdução à Programação: Uma Nova Abordagem Usando C” (Varejão, 2016), cuja edição contou com a participação de integrantes do projeto. Nas primeiras edições, o curso era dividido em aulas teóricas, em sala de aula, e práticas, em laboratório. Além disso, eram propostas atividades extraclasse (chamadas de *workings*). A partir de 2013 o curso também passou a incluir *Hacking Days* – aulas apoiadas na metodologia ativa de Aprendizagem Baseada em Problemas (PBL) (Looi & Seyal, 2014). Em cada *Hacking Day*, os estudantes são apresentados a alguma tecnologia ou área nova, como robótica, desenvolvimento *web* e programação competitiva, e é proposto um desafio a ser resolvido em equipe. Esse desafio normalmente é apresentado como um problema contextualizado em alguma situação real. Os estudantes devem então desenvolver uma solução aplicando os conhecimentos de programação já aprendidos e a tecnologia apresentada no *Hacking Day*. Ocasionalmente, são oferecidas premiações aos estudantes que obtenham o melhor desempenho.

Em 2013, ocorreu uma ampliação importante do projeto, que passou a ser aberto a toda rede de ensino médio público da região metropolitana e as vagas para o curso passaram de 31 para 80, como mostrado na Figura 3. Devido a essa ampliação, a equipe instituiu um processo seletivo constituído de duas etapas. A primeira etapa tem variado ao longo dos anos (atualmente ela tem o formato de uma aula expositiva), enquanto a segunda trata-se de uma prova de lógica. Além disso, a equipe também sentiu a necessidade de um ambiente de apoio para a gestão das atividades e a comunicação com os estudantes. Com isso, em 2014, o projeto passou a adotar o Moodle<sup>2</sup> para que os estudantes submetessem seus *workings*, permitindo que a equipe acompanhasse de forma mais individualizada as atividades das turmas. O Moodle foi escolhido por ser uma ferramenta *open source* amplamente utilizada no meio acadêmico.

Após a edição de 2014, um dado que nos chamava a atenção eram os baixos índices de

---

<sup>2</sup><https://moodle.org/>

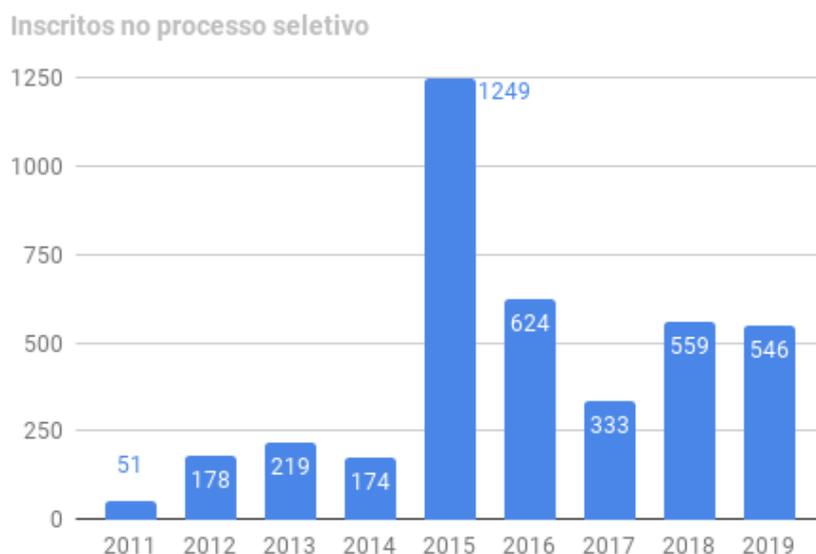


Figura 2: Evolução dos números de inscritos no processo seletivo.

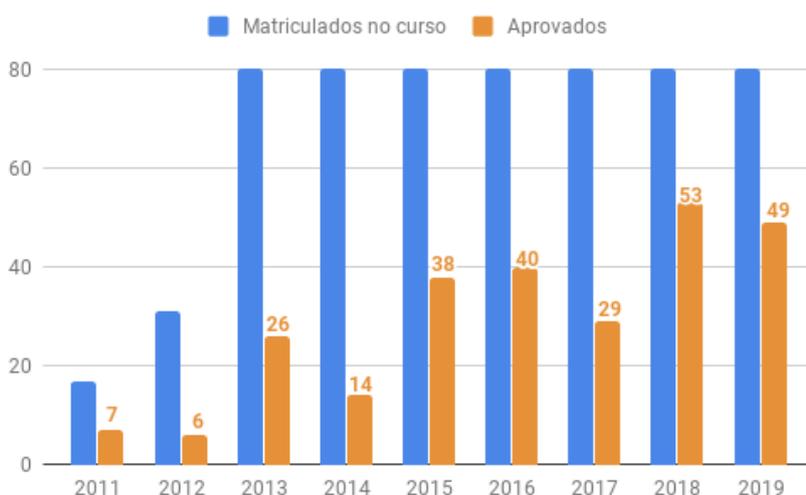


Figura 3: Evolução dos números de matriculados e aprovados no curso durante os anos pré-pandemia.

aprovação no curso (o estudante é aprovado no curso se obtém ao menos 70% de aproveitamento). Como mostrado na Figura 3, nesse ano a taxa de aprovação havia sido a menor até então.<sup>3</sup> De fato, a programação de computadores, mesmo em nível introdutório, pode não ser uma tarefa simples (Farias et al., 2018), (Looi & Seyal, 2014), (Müller et al., 2018), (Odekirk-Hash & Zachary, 2001). Ela exige habilidades técnicas e cognitivas como bom raciocínio lógico e capacidade de interpretação (Priesnitz Filho, Abegg, & de Oliveira Simonetto, 2012). Para que se desenvolva o pensamento algorítmico e se compreenda bem as técnicas de programação é fundamental que sejam realizadas muitas atividades práticas (Ng, Choy, Kwan, & Chan, 2005). Tendo isso em vista, em 2015, houve uma mudança importante na organização das aulas – elas deixaram de ser

<sup>3</sup>Alunos não aprovados incluem aproveitamento inferior a 70% ou desistência.

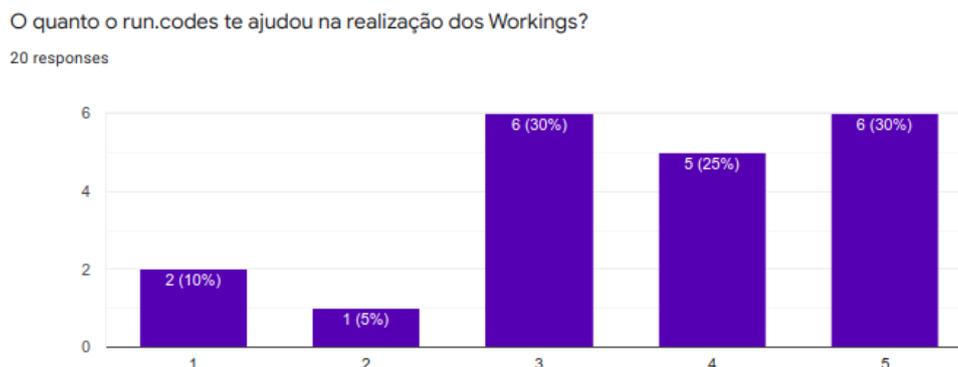


Figura 4: Pesquisa realizada com egressos da turma de 2017 quanto ao uso do run.codes. O índice na escala Likert de 5 pontos reflete a intensidade do apoio.

divididas em aulas teóricas e práticas. Todas as aulas passaram a ter pelo menos metade da carga horária dada em laboratório, com a realização de atividades práticas. Acreditamos que, com isso, em 2015 e 2016 tivemos um aumento importante no número de aprovados.

Para ampliar as oportunidades de interação com os estudantes fora da sala de aula (por ex., para tirar dúvidas e compartilhar materiais de apoio), também em 2016 a equipe criou um grupo no Facebook <sup>4</sup>. Na época, essa rede social apresentava grande aceitação entre estudantes e membros da equipe, havendo igualmente relatos de suas experiências bem sucedidas no suporte ao ensino e aprendizagem (Alencar et al., 2013).

Até 2016, a prova de lógica, aplicada como a segunda etapa do processo seletivo, era uma prova objetiva com questões de lógica de diferentes níveis realizada de forma manual, em papel. A equipe tinha que se esforçar muito para aplicar e corrigir todas as provas (geralmente em menos de uma semana). Visando otimizar esse processo, foi dada a ideia de que a prova fosse realizada de forma digital, usando os computadores do laboratório de graduação. Para tanto, em 2017, um membro da equipe desenvolveu um sistema simples de prova integrado a um banco de questões. Com isso, conseguíamos aplicar um processo justo, economizar milhares de folhas de papel e tinta de impressora, aumentar a confiabilidade das notas registradas (menos sujeitas a erro humano) e acelerar a divulgação dos resultados.

Fazendo uma reflexão ao longo desses primeiros anos, constatamos que, para melhorar o processo de ensino e aprendizagem, precisávamos nos apoiar em métodos e ferramentas de programação centrados no usuário (afinal, programadores também são usuários) (Kelleher & Pausch, 2005). Nessa direção, identificamos que a correção dos *workings*, feita até então de forma manual, demandava muito tempo da equipe e inviabilizava a verificação de mais de uma submissão para cada estudante/exercício. Apostando que interatividade e *feedback* poderiam contribuir de forma significativa para a motivação e desempenho dos estudantes (Odekirk-Hash & Zachary, 2001), em 2017 passamos a adotar a ferramenta de correção automática run.codes <sup>5</sup>. A partir desse ponto, o tempo de correção caiu para alguns segundos e permitiu não só uma melhor utilização da equipe como também um maior engajamento dos estudantes (Gerhardt et al., 2018). A Figura 4 mostra o

<sup>4</sup><https://www.facebook.com/>

<sup>5</sup><https://run.codes/>

resultado de uma pesquisa realizada com egressos da turma de 2017 quanto ao uso do run.codes.

No entanto, mesmo com a adoção do run.codes, ao final da edição de 2017, a taxa de aprovação no curso voltou a ser uma preocupação da equipe. De forma recorrente, as pesquisas realizadas com os egressos do curso ao final de cada edição mostravam que, além da limitação no acesso a computador e internet em casa, uma das grandes dificuldades dos estudantes era conciliar a participação no curso com suas demais atividades, curriculares ou mesmo extracurriculares. A título de ilustração, seguem algumas respostas dadas à pergunta “Houve pontos que te desmotivaram ao longo do curso? Quais?” em enquetes realizadas com egressos do curso:

“sim como estávamos atolados de dever na escola e mais nesse curso, sei que iria sobrecarregar mas os professores pegaram pesado”

“O que mais me desmotivou a continuar no curso, foi o tempo do qual não tinha para desenvolver as atividades propostas pelo curso. Mas obviamente não foi falha dos instrutores”

“Estava em ano de vestibular e para mim foi bem difícil, no final do ano perto dos vestibulares com provas na escola, listas de exercícios do introcomp e o trabalho do introcomp para entregar um dia depois das provas da ufes Não estou reclamando que são muitos exercícios, muito pelo contrário, sei que programação é necessário bastante prática.”

Nos questionamos então se não seria possível oferecer um curso de introdução à programação que mostrasse a importância e o potencial dessa ferramenta, mas que ao mesmo tempo pudesse ser melhor conciliado com as demais atividades dos estudantes de ensino médio da rede pública. A solução colocada pela equipe foi dividir o curso em dois módulos, básico e avançado, com duração aproximada de 3 meses (9 aulas) e 2 meses (8 aulas), respectivamente, totalizando assim aproximadamente 51 horas de curso.

No módulo básico, passaram a ser abordados os conceitos fundamentais da programação, começando com a noção de algoritmo e terminando com modularização. Já no módulo avançado, passaram a ser trabalhadas estruturas de dados mais elaboradas (vetores e matrizes) chegando até ponteiros. A ideia com essa divisão do curso foi permitir que mais estudantes conseguissem seguir o conteúdo do primeiro módulo, o que já permitiria a eles entender a base do que é programar. A partir dali, o estudante já teria mais condições de identificar alguma afinidade com a área de computação (na perspectiva de um possível futuro profissional) além de ter alguma autonomia para explorar novos materiais ou conteúdos para o aprendizado de programação. Já o módulo avançado seria uma oportunidade do estudante se aprofundar um pouco mais, aplicando conceitos mais elaborados da programação. Assim, com a nova estrutura oferecida pelo curso, em 2018 observamos um impacto positivo no número de estudantes que conseguiram finalizar o módulo básico (Figura 3)<sup>6</sup>. Vale ressaltar aqui que para cada edição a equipe procurou manter os mesmos instrumentos de avaliação e seus níveis de dificuldade.

<sup>6</sup>Para os anos de 2018 e 2019, o gráfico da Figura 3 mostra o número de aprovados nos módulos básicos de cada ano.

Em 2019 reconsideramos o uso da linguagem C, uma vez que parte do nosso público a julgava demasiadamente complexa. Após fazer um levantamento, decidimos adotar a linguagem de programação Python por ter uma abordagem mais voltada à lógica do que à sintaxe (Nikula et al., 2007) e também por ter passado a ser utilizada tanto na Maratona de Programação<sup>7</sup> da Sociedade Brasileira de Computação (SBC) quanto na Olimpíada Brasileira de Informática<sup>8</sup> (OBI). Apesar de tal decisão poder, até certo ponto, suavizar a curva de aprendizagem, outra ponderação recorrente dos estudantes era que o curso não proporcionava uma experiência prática durante as aulas. Lembrando que no início de algumas aulas, parte do conteúdo ainda era abordado de forma teórica, fora do laboratório. Para mitigar isso, reformulamos todas as aulas, que passaram então a ser integralmente no laboratório. Simultaneamente, adotamos a ferramenta Jupyter Notebook<sup>9</sup>, que permite à equipe do projeto confeccionar material didático interativo, e aos estudantes interagir e modificar os exemplos de código-fonte fornecidos em seus próprios computadores. Essa estratégia interativa e baseada em exemplos (Müller et al., 2018) trouxe mais agilidade à resolução dos exercícios propostos, ao esclarecimento de dúvidas e, conseqüentemente, à aprendizagem prática dos estudantes.

Também, com o sucesso do Micro:bit<sup>10</sup> – placa de desenvolvimento com LED's, botões, acelerômetro, magnetômetro, etc. – em alguns Hacking Days de 2018, o microcomputador foi adicionado às aulas semanais do módulo básico. Tal mudança resultou em um aprendizado mais dinâmico e facilitou o estudo de conceitos ensinados nas aulas de forma mais aplicada. Concomitantemente, passamos a utilizar as ferramentas Microsoft MakeCode<sup>11</sup>, um editor on-line de programação em bloco (Aureliano & Tedesco, 2012) que permite a criação e simulação de projetos do Micro:bit e Mu<sup>12</sup>, um editor *offline* com integração com o Micro:bit e foco em programadores iniciantes. Ambas as mudanças, a inclusão de Micro:bits para a aula e a utilização da ferramenta Jupyter Notebook, serão abordadas posteriormente na Seção 4.

Por fim, ainda em 2019, propomos como projeto final do módulo avançado o desenvolvimento de parte do simulador de batalha do jogo Pokémon<sup>13</sup>. O desafio foi recebido com entusiasmo e motivou os estudantes a aplicarem os conceitos vistos durante o curso.

## 4 Estratégias de Ensino

Uma das principais características da linguagem de programação Python é priorizar a legibilidade do código sobre o desempenho computacional, combinando uma sintaxe concisa e clara com os recursos de sua biblioteca padrão e de módulos desenvolvidos por terceiros (Nikula et al., 2007). Com isso, códigos escritos em Python são considerados de fácil leitura e relativamente curtos se comparados a códigos similares em outras linguagens. Atualmente é uma linguagem muito popular para processamento de textos e dados científicos e, de acordo com o TIOBE Index (TIOBE.com, 2020), Python é a terceira linguagem mais popular do mundo, perdendo apenas para

<sup>7</sup><http://maratona.sbc.org.br>

<sup>8</sup><https://olimpiada.ic.unicamp.br>

<sup>9</sup><https://jupyter.org/>

<sup>10</sup><https://microbit.org/about/>

<sup>11</sup><https://www.microsoft.com/en-us/makecode>

<sup>12</sup><https://codewith.mu/>

<sup>13</sup><https://www.pokemon.com>

Java e C, e foi considerada a “linguagem do ano” em 2007, 2010 e 2018.

Por priorizar legibilidade, Python vem sendo amplamente utilizada em cursos introdutórios de programação. No ensino médio, a utilização bem-sucedida de Python como a primeira linguagem de programação é praticamente uma unanimidade, como mostrado por (B. Santana, Figuerêdo, & Bittencourt, 2017), (Rodrigues et al., 2013) e (Mannila, Peltomäki, Back, & Salakoski, 2006). Os benefícios são claros principalmente com os estudantes que ainda não estão totalmente certos da carreira que querem seguir. Além da linguagem de programação, estudos indicam que o emprego de recursos educacionais diferenciados, como plataformas interativas, *e-learning*, laboratórios autoguiados e exercícios práticos, trazem muitos benefícios ao ensino de programação (Radenski, 2006).

Jupyter é um exemplo de plataforma *online, open-source*, que permite desenvolvimento interativo de *software* com suporte a várias linguagens de programação, dentre elas, Python. Com Jupyter é possível criar e gerir documentos (chamados “*notebooks*”) que podem conter texto formatado, figuras, “*live code*” (código executável) e equações. Desta forma, é possível que conteúdo teórico seja intercalado com exercícios de forma mais fluida. A plataforma está disponível para instalação e desenvolvimento local, sendo também possível a utilização de um Jupyter Hub, que permite disponibilizar “*notebooks* virtuais” por curso (ou instituição) para que várias pessoas possam acessar *online* e simultaneamente, em contas individuais, sob a disponibilidade de uma infraestrutura separada.

A utilização de exercícios que façam sentido na prática também é um desafio no ensino de computação: geralmente, *softwares* reais são muito complexos e praticamente impossíveis de serem analisados e projetados em sala de aula. Isso acaba fazendo com que os exemplos, exercícios e trabalhos sejam simplificados demais, desestimulando assim, o aprendizado. Uma alternativa relativamente simples e viável em alguns cursos é a utilização de plataformas de sistemas embarcados, como Arduino<sup>14</sup> ou Micro:bit. Com essas plataformas é possível interagir fisicamente (de forma lúdica) com os dispositivos e materializar os comportamentos codificados de forma abstrata nas linguagens de programação.

#### 4.1 Nossos *Notebooks*

Nesses quase 10 anos de experiência lecionando programação para estudantes do ensino médio, percebemos algumas características do material didático que estimulam o aprendizado: o material precisa usar um linguajar simples, compatível com a faixa etária; precisa oferecer interatividade; e, sempre que possível, minimizar teoria e maximizar exercícios práticos.

Portanto, temos buscado priorizar essas características na elaboração do material didático utilizado no curso. Como pode ser visto na Figura 5, os *notebooks* elaboram o conteúdo seguindo todas essas diretrizes: o conteúdo teórico é minimizado e intercalado com exercícios práticos por meio de “*live-codes*” e os textos que transmitem conteúdos teóricos são escritos como uma “conversa entre amigos”. Além disso, todos os *notebooks* contem uma série de exercícios práticos e uma seção “O que pode dar errado?”, discutindo os erros mais comuns nos códigos que abordam aquele conteúdo.

Os temas das aulas do módulo básico abordam conteúdos fundamentais de programação,

---

<sup>14</sup><https://www.arduino.cc/>

desde o entendimento do que são algoritmos, estruturas condicionais, estruturas de repetição até modularização de código. Já o módulo avançado, além de introduzir alguns conceitos mais elaborados de programação, como orientação a objeto, aborda temas com grande apelo junto aos jovens atuais, como programação de jogos e inteligência artificial.

**Exemplo:**

Façamos um programa para calcular as raízes de uma equação de segundo grau da forma  $ax^2 + bx + c = 0$ . As entradas e saídas do nosso programa devem ser as mesmas da tabela abaixo, ou seja, para cada valor de  $a$ ,  $b$  e  $c$  na entrada a saída do seu programa deve ser a mesma da tabela. Tarefa simples, não é mesmo?

a	b	c	Saída Esperada
1	2	3	Não há raiz real :(
3	2	1	Não há raiz real :(
-1	-2	-3	Não há raiz real :(
1	0	0	x1 = 0.0 & x2 = 0.0
1	5	3	x1 = 4.0 & x2 = -9.0
4	9	3	x1 = 3.0 & x2 = -5.25

**NOTA:** A explicação abaixo contém conceitos ainda não explicados no curso, eles estão presentes apenas para fazer com que o exemplo funcione, sem eles teríamos erros.

```

j1: # PRIMEIRA TENTATIVA:

# Recebendo os valores de a, b e c:
c = int(input("Qual o valor de 'a'? "))
b = int(input("Qual o valor de 'b'? "))
a = int(input("Qual o valor de 'c'? "))

# Calculando o delta:
delta = b**2 - 4*a*c

```

Figura 5: Trecho de um notebook com um exemplo na teoria e na prática (código interativo). É possível visualizar uma tabela e diferentes formatações de texto.

Ressaltamos que o desenvolvimento dos notebooks é contínuo. Desde que começamos a utilizar o Jupyter, para cada aula, ouvimos os *feedbacks* dos estudantes e também da equipe. Em reuniões semanais, a equipe se reúne e analisa a aula lecionada no sábado anterior sob duas óticas: (i) a do instrutor, que fornece *feedback* sobre, por exemplo, relação entre conteúdo e tempo de aula, comportamento dos estudantes frente ao conteúdo (nível de dificuldade), etc.; (ii) a do estudante, que preenche um formulário depois de cada aula com questões gerais sobre a qualidade da aula. Esses *feedbacks* são anotados e as soluções e melhorias são discutidas pela equipe e incorporadas ao material didático para a próxima edição do curso.

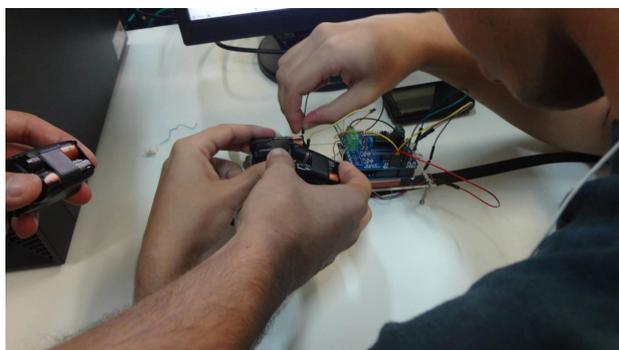


Figura 6: Alunos configurando o dispositivo Micro:bit.



Figura 7: *Display* do Micro:bit indicando, por meio de uma seta, o botão que deve ser pressionado pelo usuário.

## 4.2 Micro:bits

Nas edições presenciais do curso, usamos com frequência a plataforma Micro:bit. Esse equipamento permite criar códigos embarcados rapidamente por meio de interfaces simples, tanto de *hardware* quanto de *software* e, assim, simplificar o trabalho com eletrônica.

O Micro:bit é parte de um grande projeto desenvolvido em parceria com a emissora estatal inglesa BBC visando promover o ensino da programação para crianças e jovens. Ele contém, além de outras funcionalidades, sensores de movimento, luminosidade, temperatura e campos magnéticos. Com isso, é possível programar uma variedade de sistemas, desde instrumentos musicais, a sistemas de monitoramento e Internet das Coisas (IoT). Há documentação de qualidade disponível na Internet, bem como exemplos de projetos interessantes, dos mais simples aos mais complexos.

Durante a aula, cada estudante recebe um kit Micro:bit (Figura 6) contendo a placa com o microprocessador, cabos para fazer o *upload* do código e, quando necessário, materiais para execução do projeto (papelão, papel alumínio, etc.). A Figura 7 ilustra um dos exercícios planejados pela equipe para a quinta aula do curso, que consiste no desenvolvimento de um sistema capaz de fazer um teste de atenção e reação nos funcionários de uma indústria para verificar se estão aptos a trabalhar. Visto que a falta de atenção é uma das maiores causas de acidentes de trabalho, um sistema desse tipo pode ajudar a evitar possíveis acidentes. Nesse exercício, cabe ao estudante: (i) criar uma forma aleatória de indicar ao funcionário que deve apertar um determinado botão do Micro:bit; (ii) implementar uma forma do sistema contabilizar acertos e erros; e, (iii) ao final, criar uma forma de mostrar no *display* se o funcionário está apto a trabalhar ou não (baseado em algum limite de acertos/erros pré-definido).

## 5 Planejamento no Avançar da Pandemia

Começamos 2020 com a intenção de expandir o projeto e elaborar novas versões do curso – para estudantes do ensino fundamental (“IntrocompJr”) e para professores do ensino médio da rede pública do estado, visando formar multiplicadores do Introcomp. Com a ajuda de integrantes externos, denominados internamente de *trainees*, seriam gerenciados três cursos simultaneamente.

Todavia, com o contexto da pandemia, tivemos que repensar nossos planos. Inicialmente, o calendário de 2020 foi suspenso e a equipe aguardou uma posição formal da UFES e da SEDU. Mas com as incertezas ao longo do tempo a equipe decidiu primeiramente pelo cancelamento do curso de formação de professores, devido ao provável cenário conturbado da educação na volta às aulas. Além disso, o módulo básico para estudantes do ensino médio seria adaptado para ser realizado de forma *online*. Já o IntrocompJr e o módulo avançado para o ensino médio seriam, a princípio, mantidos presenciais. Nesse cenário, o principal desafio seria reformatar o módulo básico, e a nova experiência em uma edição especial para estudantes do fundamental com o IntrocompJr.

Em meados de junho de 2020 ainda não existia nenhuma previsão concreta para a volta às aulas presenciais e o cenário se tornara cada vez menos favorável. Por conseguinte, a equipe decidiu cancelar o IntrocompJr, diminuir a carga horária do módulo avançado para 6 aulas (18 horas) e concentrar seus esforços em realizar os módulos básico e avançado no formato totalmente *online* e remoto, atendendo o público tradicional do projeto – estudantes do ensino médio da rede pública. Uma vez definida a meta, eis a questão: *qual seria o ambiente virtual de aprendizagem e as ferramentas de interação e comunicação adequados?*

## 5.1 Escolha do Ambiente Virtual de Aprendizagem

No processo de escolha do ambiente virtual de aprendizagem, começamos por um levantamento das plataformas de videoconferência que vêm sendo adotadas recentemente para o ensino *online* e remoto de escolas públicas e privadas. As seguintes plataformas foram consideradas: Jitsi<sup>15</sup>; BigBlueButton<sup>16</sup>; Webex<sup>17</sup>; e Riot<sup>18</sup>. Em seguida, definimos alguns critérios de interesse para nosso caso de uso, sendo eles: gratuidade; *open source* (para o caso de ser preciso modificar a plataforma conforme as nossas necessidades); acesso por meio de um navegador *web*; dispensabilidade de cadastro do usuário; gravação de aula; interação entre usuários por áudio e vídeo; comunicação por meio de troca de mensagens; e criação de turmas. Conforme ilustrado na Tabela 1, Jitsi e BigBlueButton se destacaram, com uma pequena vantagem para a primeira.

Por outro lado, também ponderamos a utilização de plataformas de *live streaming*, devido à evidência que essas têm ganhado durante a pandemia. Se num primeiro momento tal modalidade é associada ao entretenimento<sup>19</sup>, vale ressaltar casos de sucesso no ensino à distância, como o exemplo do professor de biologia Paulo Jubilut em preparatórios para o Enem, com centenas de milhares de visualizações<sup>20</sup>. Portanto, tendo em mente a característica de um único *stream* das plataformas de *live streaming*, em contraposição aos múltiplos *streams* das plataformas de videoconferência (que podem fugir do controle), reconsideramos nossos planos iniciais de expansão do público-alvo e começamos a trabalhar com a possibilidade de adotar uma plataforma de *live streaming* para realização de um curso aberto, sem limite máximo de participantes.

---

<sup>15</sup><https://jitsi.org/>

<sup>16</sup><https://bigbluebutton.org/>

<sup>17</sup><https://www.webex.com/>

<sup>18</sup><https://riot.im/>

<sup>19</sup><https://www.reuters.com/article/us-health-coronavirus-brazil-livestreami/how-housebound-brazil-popstars-and-ceos-caught-streaming-mania-and-dominated-youtube-idUSKBN22B1BL>

<sup>20</sup><https://www.youtube.com/c/biologiatotalbrjubilut>

Tabela 1: Comparativo entre as plataformas de vídeo conferência consideradas.

Nome	Gratuito	Open source	Acesso via Navegador	Dispensa Cadastro	Gravação de Aula	Interação por Áudio e Vídeo	Troca de Mensagens	Criação de Turmas
Jitsi	<input checked="" type="checkbox"/>							
BigBlueButton	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Webex	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Riot	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Nesse contexto, consideramos a plataforma Twitch<sup>21</sup>, principalmente por seu foco em *live streamings* de vídeo games e competições de *e-sports*, que atraem o público jovem e interessado em tecnologia. Além de atender vários dos critérios apontados anteriormente, trabalhos demonstram o potencial dessa plataforma no suporte ao processo de ensino e aprendizagem (Faas, Dombrowski, Young, & Miller, 2018). Na programação, por exemplo, o usuário brasileiro *danielhe4art* acumulou mais de 1200 espectadores simultâneos durante uma *stream* em março de 2020.

Todavia, observamos que, apesar de a Twitch permitir a troca de mensagens via *chat*, essa não seria a melhor opção para interação e comunicação entre os membros da equipe e os estudantes. Isso porque os canais de *chat* são abertos e geralmente recebem uma quantidade grande de mensagens simultâneas, o que seria muito difícil de gerenciar. De forma a mitigar essa característica da Twitch, avaliamos o Discord<sup>22</sup>, uma plataforma que possibilita a criação de servidores públicos ou privados com a opção de se criar múltiplos canais de texto e voz, usuários com papéis (e permissões) distintos, dentre outras. Ambas as plataformas são gratuitas e podem ser acessadas por meio de um navegador *web* ou aplicativo para celular. Assim, optamos pela combinação das duas, o que nos permite manter o material produzido no Jupyter Notebook enquanto as aulas podem ser acompanhadas, a princípio, por um número ilimitado de estudantes. Já o acesso ao *run.codes* continua sendo limitado (via processo seletivo remoto), devido a limitações em sua infraestrutura.

## 5.2 Infraestrutura e Planejamento do Curso

Para tornar as transmissões na Twitch mais atrativas, foi utilizado o Streamlabs OBS<sup>23</sup>, um programa que permite personalizar o *stream* ao vivo com a seleção de cenas, recursos audiovisuais de maneira fácil e intuitiva. Além disso, essa ferramenta permite inserir recursos para interação entre o *streamer* (no nosso caso, o instrutor) e o público. Exemplo disso seria a exibição de GIFs personalizados quando algum visualizador começa a seguir o perfil do curso. Esses recursos de interação podem ser configurados programaticamente com o auxílio de *bots* capazes de “escutar” o canal de *chat* da Twitch, interceptar mensagens inapropriadas ou mesmo disparar um comando com informações relevantes, como a contagem de estudantes com dúvidas em um dado instante. Com isso, é possível montar uma transmissão mais elaborada que a tradicional “câmera com compartilhamento de tela”.

Com tais recursos, foi planejado que durante a aula o instrutor permanece em um canal de voz no Discord limitado apenas à equipe organizadora do curso. Além disso, é aberto um

<sup>21</sup><https://www.twitch.tv/>

<sup>22</sup><https://discord.com/>

<sup>23</sup><https://streamlabs.com/>

número limitado de canais de voz, cada qual com no máximo dois estudantes e um membro da equipe para tirar dúvidas. Um *bot* monitora essa alocação e, em caso de aumento da demanda, cria automaticamente uma fila de espera. Se observada uma alta movimentação nesses canais de comunicação, a equipe pode eventualmente solicitar ao instrutor uma pausa na aula, seja para os estudantes processarem melhor o conteúdo, ou para explicar novamente algo que não tenha ficado claro.

A equipe também pode estabelecer horários adicionais para atender os estudantes e esclarecer dúvidas (fora do horário convencional das aulas). O servidor do Discord ainda deve prever canais destinados à interação entre os próprios estudantes do curso e, com auxílio de um *bot*, um mecanismo de bonificação para os mais ativos. Essa bonificação é uma pontuação, computando *Intropoints*, que determina o cargo no servidor do Discord (semelhante à reputação), de maneira que um usuário melhor pontuado tem seu nome exibido mais próximo do início da lista de usuários e o nome com cor personalizada. Mesmo que simples, em um primeiro momento, esse mecanismo de incentivo pode ajudar na formação de uma comunidade mais participativa.

Como foi dito anteriormente, há uma limitação quanto ao número de usuários que podem ser registrados no run.codes. Apesar disso, os *workings* permanecem compartilhados com todos os estudantes. Assim, sempre que os *workings* forem passados para a turma, todos podem tentar resolvê-los, e nas aulas subsequentes eles são corrigidos pelo instrutor, de maneira a tornar o curso mais inclusivo e atraente para o público-alvo.

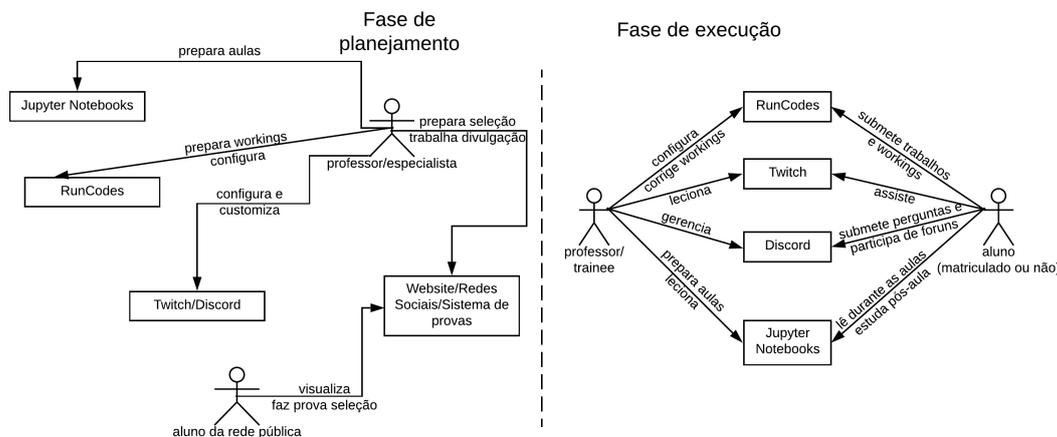


Figura 8: Visão geral das tecnologias envolvidas e os papéis dos usuários.

A Figura 8 fornece uma visão geral das tecnologias envolvidas no curso e os papéis dos usuários em relação a essas tecnologias. Nela são ilustradas as duas principais fases do desenvolvimento do curso, contemplando o planejamento até a execução de uma edição.

Na fase de planejamento o curso é preparado, realizando a configuração e customização das principais ferramentas que serão aproveitadas ao longo do período letivo. Também dedica-se parte considerável dos esforços na área de divulgação, alimentando as redes sociais e contando com a parceria estabelecida com a SEDU para a aproximação com as escolas públicas do estado. Por último, elabora-se o processo seletivo, iniciando pela abertura das inscrições e encerrando com alunos já selecionados.

Já na fase de execução, tem-se como principal base as aulas lecionadas aos sábados na Twitch, de forma que os *notebooks* desenvolvidos pela equipe são utilizados tanto pelo professor quanto pelos alunos. A plataforma *run.codes* é programada para liberar novos exercícios aos alunos assim que uma aula é encerrada, contendo questões relacionadas ao conteúdo explorado. O suporte aos alunos ocorre durante toda a semana, com a interação concentrada principalmente no servidor do Discord. Em volta de todo o processo, tem-se reuniões semanais com a equipe do projeto, levantando os principais pontos de uma semana específica, expondo potenciais dores em algum setor do projeto e estabelecendo perspectivas para os próximos passos.

## 6 Experiência do Processo Seletivo

Em anos convencionais, a divulgação do processo seletivo tinha como recurso mais forte a visita a escolas estratégicas da Grande Vitória, distribuindo cartazes e divulgando diretamente nas salas de aula. Para a atual edição contamos com a colaboração direta da SEDU, que encaminhou emails para escolas, além de mensagens para diferentes grupos de professores, os quais ficaram responsáveis por repassar as informações aos estudantes. Prevendo um alcance menor, intensificamos nossa participação nas redes sociais, enviando mensagens diretas para escolas estaduais e institutos federais de todo o país. A efetividade dessa ação ficou clara, visto que o número de seguidores do perfil do Instagram<sup>24</sup> aumentou mais de 50%.

Ao final do processo seletivo alcançamos o número de 867 inscrições, um aumento de 44,7% em relação ao ano anterior. Com a abertura do nosso escopo para além da região metropolitana concentrada do Espírito Santo, o número de escolas atingidas saltou de 89 para 221, sendo 154 de cidades que anteriormente não eram contempladas pelo processo seletivo.

Nosso primeiro grande teste das plataformas ocorreu na aula expositiva (primeira etapa do processo seletivo), em que todos os estudantes que tivessem interesse na matrícula deveriam assisti-la e submeter um formulário com 11 questões sobre o projeto. Na transmissão ao vivo da Twitch tivemos 412 acessos de usuários únicos, com média de 254 espectadores ao longo de toda a aula, e a gravação possui mais de 1350 visualizações.

Em termos de interatividade dos estudantes com a aula, devemos ressaltar que entre os usuários únicos, 193 (46,6%) enviaram mensagens pelo *chat*, totalizando 1873 mensagens. Tivemos também uma média geral de acertos nas questões do formulário de 87,32%, com 321 participantes únicos. Todos esses números demonstram que o estudante não só se sente confortável em interagir durante a aula, como também valoriza o conteúdo e se concentra.

Desde 2019, a ampliação do projeto com uma possível expansão para outros municípios localizados fora da Grande Vitória já estava sendo pensada. Com isso, seria necessário um sistema de provas melhor, mais escalável, robusto e portátil para que a segunda etapa do processo seletivo pudesse ocorrer remotamente. O sistema original só podia ser executado em rede local, sendo projetado para que os estudantes realizassem a prova de lógica localmente na Ufes.

Para a implementação do novo sistema de provas foi decidido utilizar a tecnologia Docker<sup>25</sup> para garantir maior escalabilidade e portabilidade, em que foram criadas duas imagens separadas:

<sup>24</sup><https://www.instagram.com/petengcomp/>

<sup>25</sup><https://www.docker.com/>

uma para o servidor *web* e outra para o banco de dados. O banco de dados utilizado foi o MySQL 5.7, por ser um dos mais utilizados no mercado e com maior compatibilidade com o *framework* utilizado. Para o servidor *web* foi escolhido o AdonisJS<sup>26</sup>, um *framework* em JavaScript baseado na arquitetura Model-View-Controller (MVC), que já fornece todo o sistema de rotas de endereços e tratamento de requisições HTTP. A Figura 9 ilustra a arquitetura do novo sistema de provas.

Após implementação e alguns testes locais, o sistema foi migrado para uma máquina virtual<sup>27</sup> alocada para o projeto pela Superintendência de Tecnologia da Informação da Universidade. O mesmo foi então populado com 95 questões, divididas em 4 níveis de dificuldade. Para cada usuário, o sistema cria dinamicamente uma prova escolhendo de forma aleatória um número predefinido de questões para cada um dos níveis. As questões são apresentadas de forma embaralhada (bem como suas alternativas de respostas). Ao assinalar uma questão corretamente, tem-se a contabilização de um ponto para o usuário, entretanto, o erro o penaliza com um desconto de meio ponto, de forma a desincentivar respostas aleatórias por parte dos estudantes.

Após a realização de vários testes por parte da equipe para verificar se a criação das provas e a correção das notas funcionavam corretamente, também foram realizados alguns testes de escalabilidade do sistema. Finalmente, o sistema foi utilizado por 266 inscritos que haviam sido selecionados na primeira etapa do processo, não havendo nenhuma intercorrência durante a execução da prova.

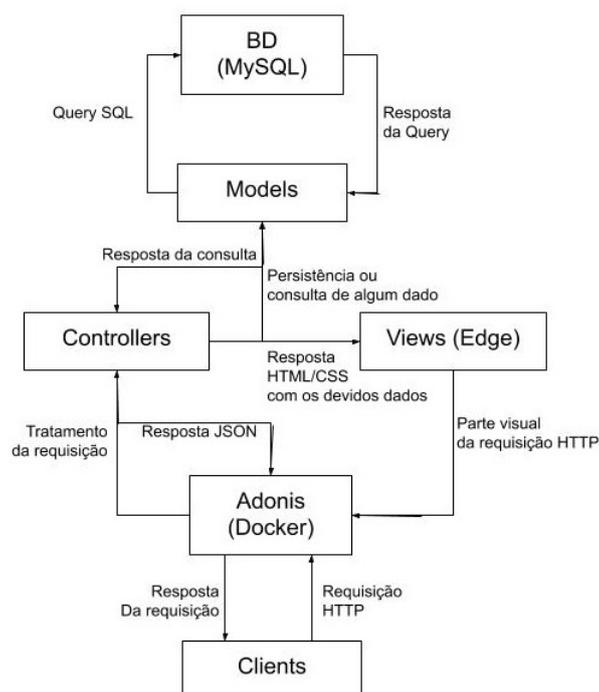


Figura 9: Arquitetura geral do sistema de provas.

<sup>26</sup><https://adonisjs.com/>

<sup>27</sup>Xeon(R) CPU 2.53GHz, 4 cores, 4GB RAM

## 7 Experiência do Ensino Remoto na Pandemia

Após o processo seletivo, iniciamos o curso com o auxílio das plataformas virtuais. Como era esperado, algumas adaptações fizeram-se necessárias. Parte dessas adaptações visaram a um maior aproveitamento das plataformas online enquanto outras adaptações foram elaboradas com o intuito de sanar algumas carências geradas pelo ensino remoto em si. O calendário de aulas e os respectivos conteúdos programáticos ficam disponíveis para os estudantes na página do Introcomp.

Dentre os recursos adaptados para o novo modelo de ensino, podemos citar a implementação de alguns mecanismos para promover uma maior interação entre os próprios estudantes, uma vez que para alguns participantes o ambiente virtual pode não ser tão convidativo à interação quanto o ambiente presencial. Por outro lado, esse mesmo ambiente virtual pode promover mais oportunidades de interação entre a equipe e os estudantes, sem a necessidade de deslocamento para encontros presenciais. Isso possibilitou a implantação de monitorias e aulas extra, como será abordado mais em detalhes a seguir.

Além disso, a equipe enfrentou problemas técnicos com relação à infraestrutura. A transmissão das aulas em *live streaming* são altamente custosas em termos de rede e processamento, estando sempre sujeitas a erros. Tendo isso em vista, a equipe teve que utilizar algumas estratégias durante as transmissões de forma a atenuar esses problemas, que serão discutidas nas próximas seções.

### 7.1 Execução do Planejamento

Durante o período de execução do curso, a equipe definiu internamente que, para cada aula, haveria sempre dois membros dedicados à transmissão, o instrutor e um comentarista. Esse último ficaria de apoio para trazer informações pertinentes do *chat* ou complementares à fala do instrutor. Também foi elaborado um sistema de redundância para a escala de aulas, havendo sempre um instrutor reserva que poderia assumir em caso de adversidade, principalmente relacionados a problemas de conexão.

As aulas síncronas aos sábados (em *live streaming*) inicialmente contavam com a presença de cerca de quatro membros da equipe no chat da Twitch, já pré escalados semanalmente, além de no mínimo três membros adicionais para o atendimento direto por voz pelo Discord. Entretanto, com o andamento das aulas, observou-se que a interação pelos canais de voz não estava ocorrendo com muita frequência, provavelmente devido à simplicidade e o baixo tempo de resposta na solução das dúvidas pelo próprio chat da Twitch. Desse modo, parte da equipe de suporte do Discord foi realocada para interagir exclusivamente pelo chat da Twitch, mantendo-se ainda os dois canais de forma a estimular a interatividade com os estudantes.

Logo após a primeira aula, a equipe procurou disponibilizar horários adicionais para melhorar o atendimento e a esclarecimento de dúvidas. Foi realizada uma pesquisa junto aos estudantes para identificar quais seriam os melhores horários para eles. A partir do resultado obtido, a equipe alocou horários semanais em que passaram a ser realizadas monitorias diretamente no servidor do Discord.

Muito além desses encontros, o servidor do Discord apresentou papel fundamental para a interação entre todos os envolvidos no projeto, sendo esta interação impulsionada pela utilização

do *bot* atrelado ao sistema de pontuação anteriormente exposto (seção 5.2). Após terem sido estabelecidos prêmios para os estudantes baseados nos *Intropoints* conquistados, criou-se uma rede de compartilhamento de conhecimento entre os próprios estudantes. Esses prêmios consistiam em, ao início de cada transmissão, os cinco estudantes com maior pontuação na semana (chamados *alunos-destaque*) terem seus nomes apresentados pelo instrutor, além de ao final do curso, todas essas conquistas serem registradas diretamente no certificado de conclusão. A funcionalidade que estabelecia uma fila de atendimento também foi muito bem aceita pelos estudantes, que se adaptaram facilmente à utilização dos comandos e aos recursos da plataforma.

Assim como no modelo presencial, foram realizados *Hacking Days*, sendo dois deles de programação competitiva e um de desenvolvimento web, havendo necessidade de adaptá-los ao modelo remoto. Nos dois primeiros, os estudantes se reuniam em equipes de até três integrantes e participavam de um torneio online, realizado na plataforma URI<sup>28</sup>. No último, os estudantes foram introduzidos aos conceitos básicos de desenvolvimento web, conhecendo HTML, CSS e JavaScript, com o código final produzido compartilhado pelo servidor do Discord.

Os métodos avaliativos também tiveram de ser repensados. A prova final do módulo básico anteriormente era realizada durante o horário de uma aula comum, entretanto, por questões de conectividade decidimos realizá-la de forma assíncrona, com duração de cinco dias e exercícios mais longos. Já o trabalho final do módulo avançado teve alteração no número de integrantes dos grupos, passando de dois para três, visto que a quantidade de estudantes matriculados neste módulo era muito maior do que nas edições presenciais e os trabalhos são corrigidos manualmente pela equipe.

## 7.2 Novos Recursos e Adaptações

Mesmo com o planejamento bem definido desde as fases iniciais, durante a execução do projeto diversos incrementos e novos recursos foram adotados para melhorar a experiência dos estudantes no curso, sendo direcionados principalmente à solução de dúvidas e conteúdos extras.

Tentando aumentar a interação entre os instrutores e os estudantes, foi introduzida a ferramenta CodeCollab<sup>29</sup>, que gera um arquivo e permite que mais de uma pessoa possa programar em um mesmo ambiente. Por meio dela, o instrutor poderia escolher um exercício do *notebook* para ser resolvido por um estudante sorteado durante a aula. Uma questão era anunciada e o estudante que quisesse resolver poderia enviar o comando *!sorteio* no *chat* da Twitch. Após determinado período de tempo do início das inscrições (de dois a cinco minutos), o *bot* selecionava o vencedor e o indicava para o instrutor, que fazia em seguida o anúncio na aula. Um dos membros da equipe, escolhido previamente, mandava o link do CodeCollab para o estudante resolver o exercício. Apesar de bem definido, o processo envolvia ações tanto da equipe quanto dos estudantes, o que o tornava mais complexo de ser executado.

Com o surgimento de dúvidas recorrentes, observadas pela equipe através dos canais de texto do Discord e nas monitorias, foram introduzidas aulas não obrigatórias no cronograma inicial às quartas-feiras, realizadas na Twitch. Nessas aulas, a equipe utilizou uma abordagem mais prática, com resolução de exercícios do URI e demonstração de códigos mais elaborados. Além

<sup>28</sup><https://www.urionlinejudge.com.br/>

<sup>29</sup><https://codecollab.io/>

disso, essas transmissões também eram utilizadas para introduzir integrantes mais novos da equipe na preparação e execução de aulas.

Foram acrescentadas também duas aulas de revisão ao cronograma do curso, buscando esclarecer o conteúdo posto e direcioná-los para futuras atividades. A primeira delas ocorreu no sábado antes da prova de encerramento do módulo básico ser liberada (os estudantes tiveram 5 dias para realizar a prova). Foi montada uma breve apresentação resumindo todos os conteúdos até então lecionados, com exercícios e pausas para que os estudantes pudessem manifestar suas dúvidas. Já a segunda foi ministrada antes do início do módulo avançado, e se fez necessária devido a um intervalo superior a dois meses sem aulas entre módulos, ocasionado pelo deslocamento de calendário decorrente da pandemia. Com isso, foram retomados os principais conceitos que seriam pertinentes para o prosseguimento do curso.

Outra facilidade não prevista inicialmente foi a publicação de todas as gravações das aulas no canal do projeto, no Youtube<sup>30</sup>. O objetivo foi ampliar e facilitar o acesso assíncrono às aulas gravadas uma vez que os vídeos na Twitch ficam salvos por apenas 15 dias. Essa iniciativa passou a permitir que em qualquer momento os estudantes pudessem ter acesso aos vídeos, seja porque tiveram problemas pessoais ou técnicos durante as aulas, seja para fazer a revisão de algum conteúdo.

Por fim, utilizada como uma ferramenta de apoio para a formação de trios, necessários tanto nos *Hacking days* de programação competitiva quanto no trabalho final do curso, foi adicionada uma funcionalidade ao *bot* do servidor do Discord que permitia que os estudantes se unissem a outros a partir do comando *!grupo*. O estudante interessado deveria acionar o comando e, quando outros dois estudantes também o fizessem, o trio seria formado. O *bot* então os comunicaria em um chat privado pela própria plataforma, citando o nome dos outros dois integrantes do trio.

## 8 Resultados e Aprendizados

As adaptações realizadas nessa nova fase do curso trouxeram resultados interessantes que conferiram à equipe uma nova visão sobre esta modalidade de ensino, tanto sobre a eficiência de diversos métodos quanto sobre as diferentes possibilidades e oportunidades desse novo modelo.

De fato, foram observados alguns impactos negativos, como algumas limitações de hardware da própria equipe, as quais impediram a execução de certas atividades por parte de seus membros. Essas limitações também afetaram os estudantes, muitos deles chegaram a tentar seguir o curso utilizando *smartphones*, que acabavam não apresentando um ambiente ideal para o aprendizado de programação. Estudantes com esse tipo de problema teriam muita dificuldade em entregar o trabalho computacional, por exemplo.

Por outro lado, diversos resultados positivos foram promovidos graças ao formato *online*. Um exemplo foi a interação estudante-projeto, que poderia ter sido comprometida pela transição para o novo ambiente, mas com a ajuda das plataformas e dos recursos automatizados, acabou sendo ainda mais fomentada. Formou-se uma comunidade virtual entre os estudantes possibilitando uma interação cordial e descontraída, além de ter contado também com estudantes de fora

<sup>30</sup><https://www.youtube.com/introcomp>

do Espírito Santo, com várias inscrições de outros estados. Esse ambiente também possibilitou uma aproximação entre os estudantes e a equipe, permitindo uma comunicação ainda mais ágil.

## 8.1 Feedback dos Estudantes

Além de ensinar a programação, o projeto também tem como objetivo despertar o interesse das pessoas pela computação em geral, instigando os estudantes a buscar conhecimento sem se prender a uma linguagem específica. Sendo assim, ao final de cada módulo (assim como em todas as aulas), disponibilizamos um formulário de feedback para que os estudantes pudessem avaliar o curso de uma forma geral em seus diversos aspectos como qualidade das aulas, didática dos professores, adaptação às plataformas e nível de interesse pela computação no início e no fim do curso.

Ao fim do módulo básico e do módulo avançado, buscamos medir o nível de interesse dos estudantes pela área da computação em relação ao início do curso com valores em uma escala de 0 a 10, na qual 0 significa nenhum interesse e 10 total interesse. Como apresentado nos gráficos da Figura 10, os resultados obtidos com essa pesquisa foram satisfatórios, apontando um aumento significativo do interesse ao final de cada módulo. Também, ao fim do curso, foi perguntado se os estudantes gostariam de seguir na área da computação e as respostas foram positivas: aproximadamente 54% tiveram certeza que sim e cerca de 39% consideraram a possibilidade. Assim, com estes resultados, acreditamos que o projeto conseguiu alcançar um dos seus objetivos, despertando interesse nos estudantes pela área da computação. Vale ressaltar que além de investir na divulgação em meios virtuais (ex: em mídias sociais, inclusive com o importante apoio da Sedu), a divulgação “boca a boca” dos próprios estudantes acaba criando uma reputação e uma tradição em torno o projeto. Com isso, também foi perguntado nos formulários qual seria a possibilidade dos estudantes recomendarem o Introcomp a um amigo. Numa escala de 0 a 10, a grande maioria respondeu com valor 10, e o restante respondeu com valores entre 7 e 9.

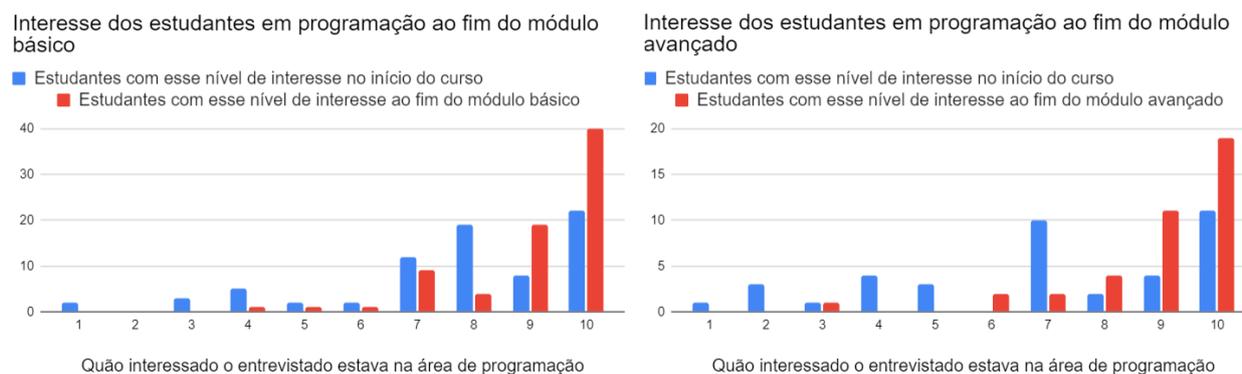


Figura 10: Interesse dos estudantes em programação no início do curso, no fim do módulo básico e no fim do módulo avançado.

Buscando obter uma opinião mais subjetiva dos estudantes, foi disponibilizada uma área para que estes fizessem comentários livres como relatos, críticas e sugestões de maneira anônima. Ter um *feedback* subjetivo melhora a percepção da equipe sobre a experiência dos estudantes, revelando pontos tanto positivos quanto negativos que não seriam percebidos por meio apenas de questões com respostas numéricas. Diversos comentários foram enviados, alguns mais resumidos e outros muito elaborados, até mesmo separados em tópicos. Muitos estudantes usaram o espaço

apara agradecer pela oportunidade de participarem do curso, para elogiar seus instrutores preferidos, para falar de sua experiência com a programação, para relatar suas dificuldades e para falar sobre o que gostariam que continuasse ou que fosse mudado. Vale ressaltar que alguns estudantes demonstraram muito interesse em como poderiam se tornar colaboradores do projeto quando terminassem o curso, seja atuando na divulgação, organização ou até mesmo como professores.

Também foi disponibilizado um espaço para que os estudantes avaliassem a utilidade do IntroBOT e fizessem comentários específicos sobre ele. O gráfico apresentado na Figura 11 mostra as respostas dos estudantes acerca da utilidade do IntroBOT. Tanto as respostas quanto os comentários foram muito positivos indicando que a experiência dos estudantes no Discord realmente foi melhorada por conta dessa automatização elaborada no ambiente.

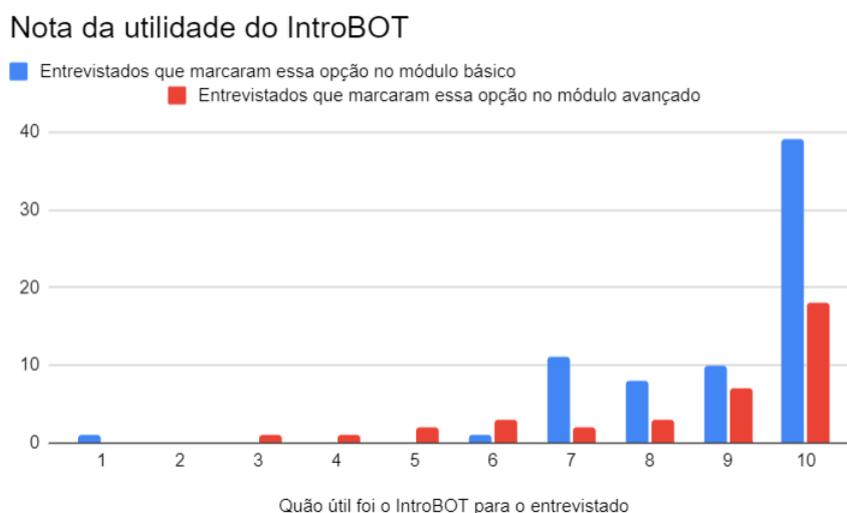


Figura 11: Utilidade do IntroBOT nos módulos básico e avançado.

## 8.2 Introspeções Pós-Curso

Com mais estudantes matriculados do que qualquer outra edição, em 2020 tivemos 150 matrículas. Destes, foram aprovados 74 no módulo básico e, na sequência, 39 estudantes no módulo avançado. Nesta seção, analisaremos as aprovações em ambos os módulos, uma vez que a partir de cada um podemos levantar hipóteses sobre as possíveis causas dos resultados obtidos.

A edição de 2020 foi a primeira a contar com a presença de estudantes de fora da Grande Vitória. Conforme observado na Figura 12, mais de 20 alunos do interior foram aprovados no módulo básico. Já no módulo avançado, o projeto obteve 11 aprovados, número maior do que o total de sua primeira edição (em 2011 o Introcomp aprovou 7 alunos). Pode-se dizer que, apesar do número aparentemente pouco expressivo de aprovados, temos um público promissor. É importante buscar o mesmo crescimento experimentado na região metropolitana do Espírito Santo para o público do interior e de outros estados.

Tendo isso em vista, para comparar os resultados das aprovações obtidas em 2020 com os resultados de 2019, passa a ser interessante observá-los considerando duas categorias diferentes de estudantes: (i) aqueles matriculados em escolas da região metropolitana de Vitória (público-alvo do Introcomp desde a concepção do projeto), e aqueles (ii) matriculados em escolas do interior

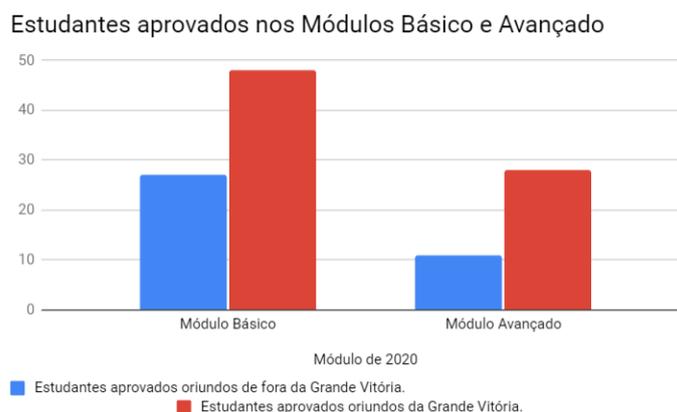


Figura 12: Gráfico mostrando o número de aprovações em cada módulo de 2020.

do Espírito Santo. Os resultados de estudantes oriundos de outros estados não foram comparados pois o número de participantes não foi expressivo.

Com base nos gráficos da Figura 13, é possível observar que o desempenho dos estudantes da Grande Vitória no módulo básico de 2020 foi muito semelhante ao resultado obtido em 2019. Isso representa um resultado muito positivo uma vez que, apesar das limitações do formato online, de todo um contexto de pandemia, e mantendo-se os mesmos níveis de cobrança dos anos anteriores, o projeto conseguiu manter a taxa de aprovações.

A edição de 2019 foi a primeira a ser realizada com base na linguagem Python e, devido à falta de experiência com o ensino da linguagem, a equipe acabou não adotando uma estrutura ideal para as aulas, em particular para o módulo avançado. Em 2020, a equipe focou na melhoria da didática e da estruturação do conteúdo. Exemplos disso foram a aula de Estruturas de Dados e as aulas voltadas à elaboração do trabalho. Outra novidade foram as aulas práticas extras, que ocorriam habitualmente às quartas-feiras. Todas essas reestruturações, juntamente com as salas de discussão no Discord e com as monitorias, permitiram um apoio muito maior aos estudantes. Esses fatores podem ser apontados como grandes potencializadores do aumento nas aprovações observado no módulo avançado, como ilustrado nos dois primeiros gráficos da Figura 14.

Pelos gráficos das Figuras 13 e 14 é possível observar que a taxa de aprovação dos estudantes oriundos do interior do ES foi menor que a dos estudantes da região metropolitana, especialmente no módulo básico. Seria necessário um estudo mais aprofundado sobre as possíveis causas para esses resultados. Possíveis fatores seriam maiores restrições de acesso por parte desses estudantes a recursos como computador e Internet.

Quanto à participação de estudantes de outros estados, podemos dizer que a inserção de estudantes de outras localidades foi extremamente benéfica, uma vez que tal ato tornou o Intro-comp acessível a pessoas que nunca tiveram a oportunidade de participar do projeto. Além disso, continuamos ofertando o curso para o público local. Sabemos, entretanto, que o curso remoto é mais sensível às limitações de hardware/internet, gerando uma certa restrição aos estudantes que não possuem a infraestrutura adequada. Isso pode acarretar em uma possível diminuição do nosso alcance na Grande Vitória.

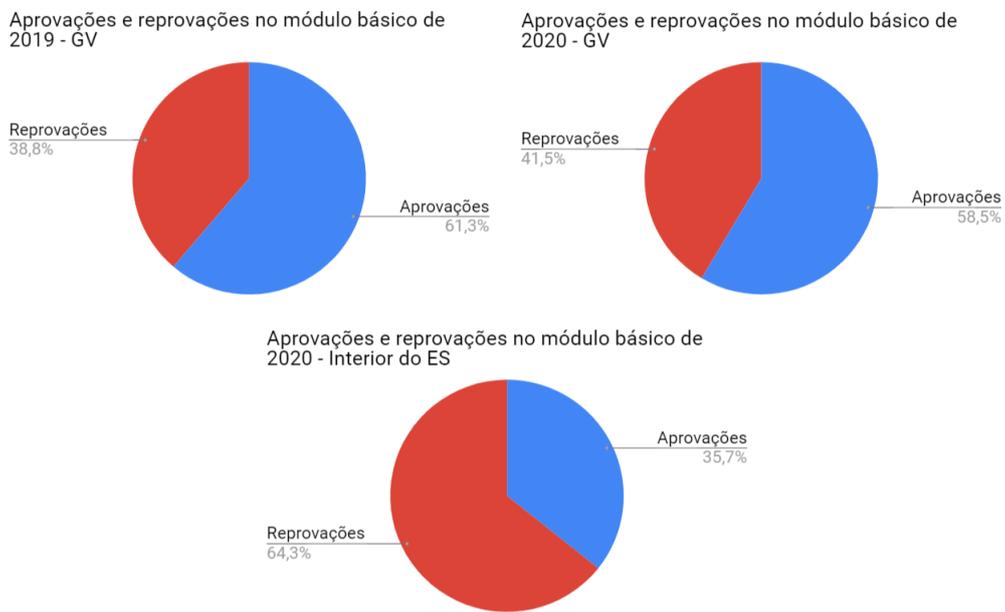


Figura 13: Gráficos mostrando as taxas de aprovados e reprovados no módulo básico nos anos de 2019 (estudantes da Grande Vitória - GV) e 2020 (estudantes da Grande Vitória e do interior do estado).

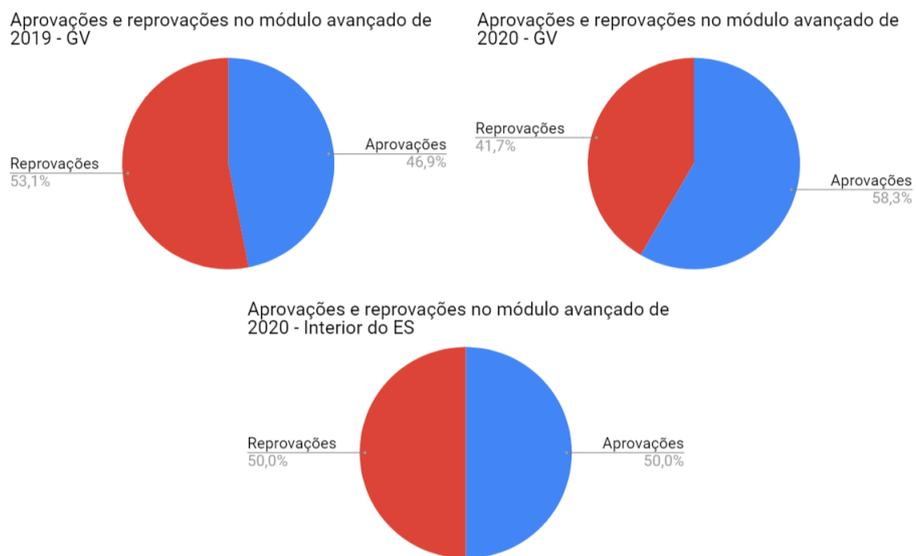


Figura 14: Gráficos mostrando as taxas de aprovados e reprovados no módulo avançado nos anos de 2019 (estudantes da Grande Vitória - GV) e 2020 (estudantes da Grande Vitória e do interior do estado).

Um fato interessante ocorrido devido à mudança do curso para o formato remoto foi a repercussão gerada na mídia local. Em 28 de novembro de 2020 foi exibida uma reportagem sobre o Introcomp<sup>31</sup> na TV Gazeta<sup>32</sup>, em que foi abordada a mudança do ensino presencial para o ensino remoto.

### 8.3 Lidando com os Desafios

Apesar de atingirmos resultados de grande expressão, também foram observados alguns desafios em diversos âmbitos ao longo da execução do curso, seja na exploração de ferramentas ou nas relações interpessoais entre os membros participantes do projeto.

No contexto geral do curso, diversas ferramentas foram utilizadas por professores e por estudantes, o que por vezes criava barreiras tanto para o ensino quanto para o aprendizado. Por parte dos professores, o programa utilizado para a transmissão requer um hardware consideravelmente potente, o que impediu alguns dos colaboradores de lecionarem as aulas. Apesar de interceptarmos os problemas de conexão com professores substitutos nas aulas, chuvas fortes ou manutenções em redes de internet eram capazes de influenciar diretamente na qualidade da aula.

Já no lado dos estudantes, a dificuldade ao acesso de equipamentos como computadores ou a pouca experiência com o uso destes foi um fator de grande dificuldade. Apesar de o projeto não exigir que os alunos possuam um computador para acompanharem o curso, ao longo da aula expositiva foi mencionada a importância de ter um dispositivo acessível para programação, sendo esse principalmente um computador, mas também um dispositivo móvel, como celular e tablet.

Mesmo que tenhamos dedicado esforço considerável em apoio e oferecendo diversas alternativas viáveis de acesso aos materiais e às aulas, como o notebook em formato *.html*, parte dos usuários ainda apresentavam carências no uso e entendimento dos recursos, resultando até mesmo na evasão em alguns casos pontuais.

Como consequência da modalidade de *live streaming*, a transmissão apresentava um atraso de aproximadamente 10 segundos. Isso resultava em uma comunicação dessincronizada, podendo o professor avançar no conteúdo da aula, porém as dúvidas presentes no *chat* serem relacionadas ao conteúdo anterior. Foi preciso adaptar-se a essa defasagem de tempo, incorporando uma abordagem com mais pausas e intervalos durante a aula.

O atraso causado pela transmissão teve impacto direto no uso do CodeCollab, pensado para promover maior interatividade com os estudantes durante a aula. Apesar de ele ter sido utilizado no início do curso, a sua adaptação foi difícil e o seu uso foi eventualmente descontinuado. Os principais problemas com a resolução de exercícios ao vivo foram a persistência de muitas pausas sucessivas em pequenos intervalos, interferindo na fluidez da aula, e a fuga do cronograma inicial estabelecido para cada aula. Como a comunicação pelo *chat* já apresentava boa participação por parte dos estudantes, a decisão do abandono da plataforma não nos trouxe prejuízos.

O uso da Twitch para transmitir as aulas permitiu alcançar um grande público de espectadores não matriculados no curso, o que é vantajoso para a visibilidade do projeto, entretanto pode apresentar certo perigo. Por vezes, esses novos espectadores entravam no servidor do Discord e passavam a se comunicar com os outros participantes do curso pelos canais de texto, tanto do ser-

<sup>31</sup>Reportagem disponível em <https://globoplay.globo.com/v/9059774/>, acesso em 02/06/2021

<sup>32</sup><https://redeglobo.globo.com/tvgazetaes/>, acesso em 02/06/2021

vidor quanto em canais privados. Tendo em vista que os estudantes eram majoritariamente menores de idade, essa comunicação deliberada entre pessoas desconhecidas, que eventualmente eram maiores de idade, e os estudantes apresenta riscos à segurança e à privacidade destes. Mesmo sem ter sido apresentada alguma denúncia de uma situação dessa forma ao longo da edição, a equipe percebeu a necessidade de enfatizar durante as aulas a importância da segurança digital, não compartilhando informações pessoais, e fazendo uso, principalmente, dos canais do servidor para as interações entre estudantes.

Outra adversidade a ser contornada foi a ocorrência de plágio presente no módulo básico. A comunicação relacionada aos *workings*, desde que construtiva e sem compartilhamento de respostas, era encorajada pela equipe por meio do servidor. Optamos por não realizar a verificação de plágio nos exercícios, principalmente pelo número expressivo de submissões, o que acarretou em problemas maiores na aplicação da avaliação final do módulo básico. Após os cinco dias de prova, a equipe foi capaz de observar, por meio do anti-plágio ofertado pelo run.codes, códigos com similaridade superior a 50%. Selecionando esses envios dentro da faixa especificada, realizamos uma análise manual e fomos capazes de confirmar plágio em cerca de 11,5% dos participantes da prova. A equipe buscou lidar com a situação de maneira compreensiva, agendando entrevista com cada envolvido e anulando apenas a questão em que a infração foi observada.

Todos os problemas aqui citados foram realmente desafiadores, porém o recesso de pouco mais de um mês e meio entre os módulos foi o que teve maior impacto negativo na permanência dos estudantes no curso. A fragilidade da comunicação a distância é exposta em situações como na taxa de evasão do curso, pois, usando um sistema de aula sem controle de frequência, não foi possível identificar nem mensurar com exatidão o abandono por parte dos estudantes. Ao final do módulo básico, tivemos 75 estudantes aprovados, porém no retorno do curso observamos que apenas 45 destes enviaram exercícios para a plataforma de correção. Visto isso, supomos que a quebra do fluxo do curso (entre dezembro e janeiro) pode ter sido um fator determinante nessa taxa expressiva, assim como um período considerável longe da programação por parte dos estudantes, o que acentuou a dificuldade de adaptação aos novos conteúdos mais complexos, mesmo que tenha sido realizada uma aula de revisão no início do segundo módulo.

## 9 Considerações Finais

Neste artigo fazemos um relato e uma reflexão sobre a nossa experiência na aplicação de diferentes estratégias e ferramentas para facilitar a aprendizagem de programação para estudantes do ensino médio da rede pública antes e durante a pandemia.

A edição de 2020, apesar do contexto global e de todos os problemas operacionais, pode ser considerada bem sucedida. A implementação do curso totalmente online requereu da equipe a absorção rápida de todas as mudanças elaboradas para que estas pudessem ser efetivadas. Além disso, uma grande capacidade de se resolver imprevistos também foi necessária, uma vez que estes foram muito mais frequentes devido a todas as novidades envolvidas no processo. Dos membros da equipe que se voluntariaram para atuarem como professores, exigiu-se um nível ainda maior de desenvoltura na comunicação e interação com os estudantes para poder compensar as limitações inerentes ao ambiente online.

Na perspectiva da equipe do projeto, as adaptações realizadas no material didático surtiram bons efeitos em comparação a edições anteriores. Entretanto, a partir dos dados obtidos por meio dos formulários de *feedback* fornecidos nas aulas, foi possível realizar reflexões sobre possíveis alterações adicionais, tanto em questões específicas de conteúdo quanto em uma adaptação ainda maior para o modelo remoto do curso, tornando o material o mais autossuficiente possível para ser utilizado de forma assíncrona. Essas iniciativas já foram esquematizadas e estão sendo postas em prática para que o material esteja ainda mais refinado para a próxima edição *online*.

Outro ponto destacável foi a utilização dos notebooks interativos que continua sendo uma opção viável e interessante para as próximas edições. A disponibilidade do conteúdo em diversos formatos, atitude tomada após diversos estudantes reportarem dificuldade no processo de instalação do Jupyter Notebook, tornou todo o conteúdo ainda mais acessível. Porém, em certos formatos há a perda da interatividade com os códigos Python presentes no material, como no caso do formato *.html*. Dessa forma, buscamos para a próxima edição alternativas que dispensassem instalação ou maiores configurações locais. O *Google Colab*<sup>33</sup> foi o escolhido, sendo este um ambiente capaz de executar notebooks Jupyter diretamente na nuvem Google. O usuário necessita apenas de um navegador web para acessar e executar os notebooks. Com isso, o estudante poderá interagir diretamente com o material didático do Introcomp via web, sem a necessidade de configurações locais específicas.

Mesmo que tenhamos apresentado índices importantes de crescimento do projeto, tanto em termos de alcance (atingindo um público que não era anteriormente alcançado) quanto de aprovações, algumas questões devem ser levantadas. Dados obtidos do processo seletivo indicam que mais de um quinto dos inscritos no curso não possuíam computador, e que não houve nenhuma aprovação no módulo avançado dentro deste grupo. A oportunidade de assistir a aulas presenciais com um computador fornecido pela própria universidade abre as portas para um público mais carente, que por vezes esbarra nas limitações de recurso e não consegue dar prosseguimento ao curso. Tendo isso em vista, ainda que a edição presencial não resolva completamente o problema descrito, esta modalidade ainda apresenta um papel social significativo para o público da Grande Vitória.

A experiência adquirida na edição de 2020 foi fundamental para que nossa equipe decidisse expandir ainda mais o projeto na próxima edição e em cenários pós-pandêmicos. Com isso, para a edição de 2021 resolvemos ampliar o número de vagas de 150 para 500, tornando o curso ainda mais amplo e expressivo em níveis estadual e nacional. Em uma futura edição em que os protocolos de distanciamento não se tornem mais necessários, estamos projetando um modelo híbrido, em que as aulas continuem ocorrendo de forma online, porém fornecendo acesso ao laboratório exclusivamente para os estudantes que necessitarem desse apoio.

De todo modo, acreditamos que a experiência foi e ainda será de grande valor, não só para a equipe e para a história do Introcomp, como também para toda comunidade científica interessada nos avanços do ensino de programação para iniciantes.

---

<sup>33</sup><https://colab.research.google.com>

## Agradecimentos

Agradecemos à Pró-Reitoria de Extensão da UFES (José J. M. Uliana é bolsista PIBEx – PRO-EX/UFES), ao PET-SESu/MEC (Programa de Educação Tutorial, Secretaria de Educação Superior, Ministério da Educação) e à FAPES (Fundação de Amparo à Pesquisa e Inovação do Espírito Santo - Termo de Outorga: 054/2020) pelo apoio financeiro. Também agradecemos à Secretaria de Educação do Espírito Santo (SEDU), que vem apoiando as ações do projeto desde 2015. Deixamos também um agradecimento especial à equipe do run.codes por disponibilizar o acesso ao sistema e todo o suporte fornecido. Por fim, agradecemos aos estudantes André Oliveira Cunha e Gabriel Castro de Rezende pelas contribuições para a realização deste trabalho.

## Artigo Premiado Estendido

Esta publicação é uma versão estendida do artigo premiado no Simpósio Brasileiro de Educação em Computação (EduComp 2021), intitulado “*Introcomp: Reflexões de uma Década de Desafios e Conquistas no Ensino de Programação para a Rede Pública de Ensino*”, DOI: [10.5753/educomp.2021.14483](https://doi.org/10.5753/educomp.2021.14483).

## Referências

- Alencar, G. A., Moura, M. R., & Bitencourt, R. B. (2013). Facebook como Plataforma de Ensino/Aprendizagem: o que dizem os Professores e Alunos do IFSertão-PE. *Educação, Formação & Tecnologias-ISSN 1646-933X*, 6(1), 86–93. [GS Search]
- Araujo, R., Amato, C., Martins, V., Eliseo, M. A., & Silveira, I. (2020). COVID-19, Mudanças em Práticas Educacionais e a Percepção de Estresse por Docentes do Ensino Superior no Brasil. *Revista Brasileira de Informática na Educação*, 28, 864–891. doi: [10.5753/rbie.2020.28.0.864](https://doi.org/10.5753/rbie.2020.28.0.864) [GS Search]
- Aureliano, V. C. O., & Tedesco, P. (2012). Avaliando o uso do Scratch como abordagem alternativa para o processo de ensino-aprendizagem de programação. In *Anais do XX Workshop sobre Educação em Computação* (p. 10). [GS Search]
- Costa, L. D., Camera, D. T., Zeferino, A. M., Kaliniski, A., Trevisan, M. G., & Zonta, F. d. N. S. (2020). Análise da vulnerabilidade entre estudantes da rede pública e privada. *Rev. Saúde Pública Paraná (Online)*, 108–119. doi: [10.32811/25954482-2020v3n1p108](https://doi.org/10.32811/25954482-2020v3n1p108) [GS Search]
- Coutinho, E., Bonates, M., & Moreira, L. O. (2018). Relato sobre o Uso de uma Ferramenta de Desenvolvimento de Jogos para o Ensino Introdutório de Lógica de Programação. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* (Vol. 7, p. 689). doi: [10.5753/cbie.wcbie.2018.689](https://doi.org/10.5753/cbie.wcbie.2018.689) [GS Search]
- Deus, W., Fioravanti, M. L., Oliveira, C., & Barbosa, E. (2020). Emergency Remote Computer Science Education in Brazil during the COVID-19 pandemic: Impacts and Strategies. *Revista Brasileira de Informática na Educação*, 28, 1032–1059. doi: [10.5753/rbie.2020.28.0.1032](https://doi.org/10.5753/rbie.2020.28.0.1032) [GS Search]
- Faas, T., Dombrowski, L., Young, A., & Miller, A. D. (2018). Watch me code: Programming

- mentorship communities on Twitch. tv. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW), 1–18. doi: [10.1145/3274319](https://doi.org/10.1145/3274319) [GS Search]
- Farias, C. M., Oliveira, A. S., & Silva, E. D. d. A. (2018). Uso do Scratch na Introdução de Conceitos de Lógica de Programação: relato de experiência. In *Anais do XXVI Workshop sobre Educação em Computação*. Porto Alegre, RS, Brasil: SBC. doi: [10.5753/wei.2018.3491](https://doi.org/10.5753/wei.2018.3491) [GS Search]
- Ferreira, A. C., Santos, J., Silva, R., Oliveira, A. T. R., Zabet, D., Abdalla, D., & Matos, E. (2016). Hello World: relato de experiência de um curso de iniciação à programação. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* (Vol. 5, p. 1306). doi: [10.5753/cbie.wcbie.2016.1306](https://doi.org/10.5753/cbie.wcbie.2016.1306) [GS Search]
- Gerhardt, L. O., Bustamante, I. M., Mai, L. F. F., Nunes, N. R., Sandrini, L., Rezende, G. C., ... Gomes, R. L. (2018). Estudo de Ferramentas de Apoio à Correção de Atividades de Programação no Contexto do Projeto IntroComp. , 10. [GS Search]
- Guzdial, M. (2009, May). Education Teaching Computing to Everyone. *Commun. ACM*, 52(5), 31–33. doi: [10.1145/1506409.1506420](https://doi.org/10.1145/1506409.1506420) [GS Search]
- Hagan, D., & Markham, S. (2000, July). Does It Help to Have Some Programming Experience before Beginning a Computing Degree Program? *SIGCSE Bull.*, 32(3), 25–28. doi: [10.1145/353519.343063](https://doi.org/10.1145/353519.343063) [GS Search]
- Kelleher, C., & Pausch, R. (2005, June). Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers. *ACM Comput. Surv.*, 37(2), 83–137. doi: [10.1145/1089733.1089734](https://doi.org/10.1145/1089733.1089734) [GS Search]
- Kemppainen, A., Fraley, M., Hamlin, A., & Hein, G. (2016). How important is high-school computing experience for first-year engineering student success? In *2016 IEEE Frontiers in Education Conference (FIE)* (p. 1-5). doi: [10.1109/FIE.2016.7757602](https://doi.org/10.1109/FIE.2016.7757602) [GS Search]
- Looi, H. C., & Seyal, A. H. (2014). Problem-based learning: An analysis of its application to the teaching of programming. *International Proceedings of Economics Development and Research*, 70, 68. [GS Search]
- Mannila, L., Peltomäki, M., Back, R.-J., & Salakoski, T. (2006, 01). Why Complicate Things? Introducing Programming in High School Using Python. *Conferences in Research and Practice in Information Technology Series*, 52. [GS Search]
- Menezes, S., & Francisco, D. (2020). Educação em tempos de pandemia: aspectos afetivos e sociais no processo de ensino e aprendizagem. *Revista Brasileira de Informática na Educação*, 28(0), 985–1012. doi: [10.5753/rbie.2020.28.0.985](https://doi.org/10.5753/rbie.2020.28.0.985) [GS Search]
- Müller, L., Silveira, M. S., & de Souza, C. S. (2018). Do I Know What My Code is “Saying”? A Study on Novice Programmers’ Perceptions of What Reused Source Code May Mean. In *Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems*. New York, NY, USA: ACM. doi: [10.1145/3274192.3274209](https://doi.org/10.1145/3274192.3274209) [GS Search]
- Neri, M., & Osorio, M. C. (2021). Evasão escolar e jornada remota na pandemia. *Revista NECAT-Revista do Núcleo de Estudos de Economia Catarinense*, 10(19), 28–55. [GS Search]
- Ng, S. C., Choy, S. O., Kwan, R., & Chan, S. (2005). A web-based environment to improve teaching and learning of computer programming in distance education. In *International Conference on Web-based Learning* (pp. 279–290). doi: [10.1007/1152804328](https://doi.org/10.1007/1152804328) [GS Search]
- Nikula, U., Sajaniemi, J., Tedre, M., & Wray, S. (2007). Python and roles of variables in introductory programming: experiences from three educational institutions. *Journal of Information Technology Education: Research*, 6(1), 199–214. [GS Search]

- Nunes, R. C. (2021, Mar.). An overview of the evasion of university students during remote studies caused by COVID-19 pandemic. *Research, Society and Development*, 10(3), e1410313022. doi: [10.33448/rsd-v10i3.13022](https://doi.org/10.33448/rsd-v10i3.13022) [GS Search]
- Odekirk-Hash, E., & Zachary, J. L. (2001). Automated feedback on programs means students need less help from teachers. In *Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education* (pp. 55–59). doi: [10.1145/364447.364537](https://doi.org/10.1145/364447.364537) [GS Search]
- Pantaleão, E., Amaral, L., & Braga e Silva, G. (2017). Uma abordagem baseada no ambiente Robocode para ensino de programação no Ensino Médio. *Revista Brasileira de Informática na Educação*, 25(03), 95. doi: [10.5753/rbie.2017.25.03.95](https://doi.org/10.5753/rbie.2017.25.03.95) [GS Search]
- Priesnitz Filho, W., Abegg, I., & de Oliveira Simonetto, E. (2012). Uma abordagem diferenciada no ensino de algoritmos através da utilização de uma lousa digital. *Revista GEINTEC-Gestão, Inovação e Tecnologias*, 2(2), 129–137. doi: [10.47059/geintecmagazine.v2i2.29](https://doi.org/10.47059/geintecmagazine.v2i2.29) [GS Search]
- Radenski, A. (2006). "Python first": a lab-based digital introduction to computer science. In R. Davoli, M. Goldweber, & P. Salomoni (Eds.), *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ITiCSE 2006, Bologna, Italy, June 26-28, 2006* (pp. 197–201). ACM. doi: [10.1145/1140124.1140177](https://doi.org/10.1145/1140124.1140177) [GS Search]
- Rodrigues, R. S., Morais, L. A. M., Dantas, S. S., Filho, J. G. L., Abilio, C. R. G. I., & Suárez, P. R. (2013). Ensino de algoritmos e linguagem de programação no nível médio: um relato de experiência. In *XXI Workshop sobre Educação em Computação*. [GS Search]
- Santana, A., Costa, J., & Castro, S. (2020). Considerações relevantes para o ensino online durante a pandemia de Covid-19 nas escolas públicas do Amapá. In *Anais Estendidos do XXVI Simpósio Brasileiro de Sistemas Multimídia e Web* (pp. 157–160). Porto Alegre, RS, Brasil: SBC. doi: [10.5753/webmedia\\_estendido.2020.13080](https://doi.org/10.5753/webmedia_estendido.2020.13080) [GS Search]
- Santana, B., Figuerêdo, J. S., & Bittencourt, R. A. (2017). Motivação de Estudantes Non-Majors em uma Disciplina de Programação. In *Anais do XXV Workshop sobre Educação em Computação*. Porto Alegre, RS, Brasil: SBC. doi: [10.5753/wei.2017.3545](https://doi.org/10.5753/wei.2017.3545) [GS Search]
- Santos, R., Sousa, B., Raiol, A., Cerqueira, P., & Bezerra, F. (2019). Uma Proposta de Método de Ensino e Relatos de Experiências com a Robótica Educacional. In *Anais do XXVII Workshop sobre Educação em Computação* (pp. 111–120). Porto Alegre, RS, Brasil: SBC. doi: [10.5753/wei.2019.6622](https://doi.org/10.5753/wei.2019.6622) [GS Search]
- Silva, R., Rivero, L., & Santos, R. (2021). ProgramSE: Um Jogo para Aprendizagem de Conceitos de Lógica de Programação. *Revista Brasileira de Informática na Educação*, 29, 301–330. doi: [10.5753/rbie.2021.29.0.301](https://doi.org/10.5753/rbie.2021.29.0.301) [GS Search]
- TIOBE.com (2020). *TIOBE Index*. Acessado em 23 de setembro de 2020. Consultado em <https://www.tiobe.com/tiobe-index>
- Varejão, F. (2016). *Introdução à Programação: Uma Nova Abordagem Usando C*. Elsevier Brasil. doi: [10.5753/rbie.2021.29.0.301](https://doi.org/10.5753/rbie.2021.29.0.301) [GS Search]