

Introcomp: Reflexões de uma Década de Desafios e Conquistas no Ensino de Programação para a Rede Pública de Ensino

Marco A. M. Oliari, José J. M. Uliana, Mirelly M. da Silva, Beatriz M. S. Maia, Thiago T. Paiva, Roberta L. Gomes, Patrícia D. Costa, Rodrigo L. Guimarães
Universidade Federal do Espírito Santo
Av. Fernando Ferrari, S/N
29075-910 – Vitória, ES, Brazil
{marcooliari, ulianamjorge, mirellymicaella, beatrizmatiasmaia}@gmail.com
thiago.t_27@hotmail.com, {rgomes, pdcosta, rlaiola}@inf.ufes.br

RESUMO

Este artigo compartilha a trajetória do projeto de extensão universitária Introcomp ao longo dos seus quase 10 anos e apresenta a estratégia que passou a ser adotada na edição do ano de 2020, pela primeira vez, integralmente a distância. Nesse sentido, fazemos uma reflexão sobre a nossa experiência na aplicação de diferentes abordagens para facilitar a aprendizagem de programação para o nosso público-alvo no decorrer desses anos – passando da linguagem C para Python, de aulas tradicionais a aulas totalmente interativas. Com isso, mostramos como as lições aprendidas têm nos ajudado 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, Computação interativa, PBL, Correção automática, Live streaming

1 INTRODUÇÃO

Com a popularização do acesso à tecnologia, 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 [8]. Com isso em mente, inúmeros trabalhos investigaram aspectos inerentes ao impacto de linguagens de programação [2][17], de estratégias/metodologias de ensino e aprendizagem [13][15][18] e de ferramentas de apoio à programação [1][11][8], para citar alguns, visando tornar o ato de programar mais acessível para programadores iniciantes (usuários tentando aprender uma linguagem de programação).

Nessa perspectiva, este artigo apresenta nossa experiência na tentativa de diminuir as barreiras para programadores iniciantes

de um curso de Introdução à Computação (Introcomp¹) oferecido para estudantes do ensino médio da rede pública ao longo de quase uma década (vide Figura 1). 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 2a), 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 principal 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; e, finalmente, a Seção 7 pondera sobre os desafios enfrentados pela equipe durante o ano de 2020 e faz uma reflexão sobre o futuro do projeto.

2 TRABALHOS RELACIONADOS

Na literatura encontramos estudos que abordam os mais variados aspectos relacionados ao ensino de programação a alunos do ensino médio. Aqui no Brasil, Ferreira et al. [7] 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

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

EduComp '21, Abril 27–30, 2021, Jataí, Goiás, Brasil (On-line)

© 2021 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

¹<https://introcomp.ufes.br/>



Figura 1: Trajetória do Introcomp ao longo dos anos.

curso de graduação em computação. de Farias et al. [5] relatam a experiência de um projeto de extensão que promoveu um curso de Introdução à Lógica de Programação para os alunos 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 alunos.

Kempainen et al. [12] investigaram a importância do acesso e familiaridade com a computação para o sucesso de alunos do ensino médio que ingressam nos cursos de engenharia. Para isso, os autores buscaram entender como as diferenças entre os alunos afetam a dinâmica de sala de aula e podem ser utilizadas como base para um planejamento mais assertivo pelos docentes. Na mesma linha, Hagan and Markham [10] apresentam um estudo que sugere que os alunos que têm experiência em pelo menos uma linguagem de programação no início de um curso de graduação têm desempenho significativamente melhor na avaliação do que aqueles sem nenhuma, e que quanto mais linguagens um aluno tem experiência, melhor tende a ser o desempenho.

Por sua vez, Guzdiak [9] compartilha as experiências de professores do Instituto de Tecnologia da Geórgia (Georgia Tech) ao longo de uma década de ensino de computação para os alunos 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 alunos 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.

Santos et al. [23] 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 et al. [4] descreve a experiência de utilização de uma abordagem baseada no desenvolvimento de jogos digitais para o ensino de lógica de programação a alunos iniciantes de um curso de graduação. Os resultados desse último sugerem que a estratégia utilizada conseguiu motivar os alunos 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 alunos, essencialmente nos jogos, com os fundamentos de programação necessários ao longo do curso.

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, 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 médio, visando com isso promover mais oportunidades a estudantes que potencialmente apresentem maior vulnerabilidade social [3]. Nas próximas seções descrevemos as intervenções e instrumentos pedagógicos utilizados, e refletimos sobre as nossas experiências, lições aprendidas e resultados alcançados em aproximadamente 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 alunos do ensino médio da rede pública. O projeto surgiu com o objetivo de democratizar o acesso à tecnologia, aumentar o interesse pelos cursos de computação e, conseqüentemente, atenuar o déficit de profissionais na área [8].

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 alunos atendidos anualmente também foi aumentando no decorrer dos anos, chegando a 80 alunos (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

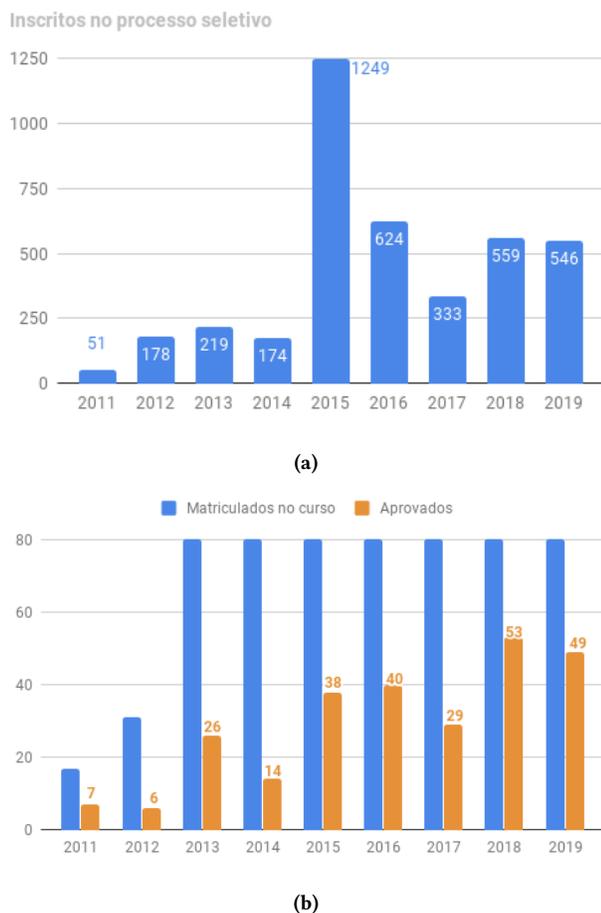


Figura 2: Evolução dos números de (a) inscritos no processo seletivo, (b) matriculados e aprovados no curso durante os anos pré-pandemia.

“Introdução à Programação: Uma Nova Abordagem Usando C” [25], 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) [13]. Em cada *Hacking Day*, os estudantes eram apresentados a alguma tecnologia ou área nova, como robótica, desenvolvimento *web* e programação competitiva, e era proposto um desafio que eles deveriam resolver em equipe. Ocasionalmente, eram oferecidas premiações aos alunos que obtivessem o melhor desempenho.

Em 2013, ocorreu uma ampliação importante do projeto, que passou a ser aberto à toda rede de ensino médio público da região metropolitana e as vagas para o curso passaram de 31 para 80, como mostrado na Figura 2b. 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 Moodle² 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 aprovação no curso. Como mostrado na Figura 2b, nesse ano a taxa de aprovação havia sido a menor até então. De fato, a programação de computadores, mesmo em nível introdutório, pode não ser uma tarefa simples [5][13][15][18]. Ela exige habilidades técnicas e cognitivas como bom raciocínio lógico e capacidade de interpretação [19]. Para que se desenvolva o pensamento algorítmico e se compreenda bem as técnicas de programação é fundamental que sejam realizadas muitas atividades práticas [16]. Tendo isso em vista, em 2015, houve uma mudança importante na organização das aulas – elas deixaram de ser 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 alunos fora da sala de aula (por ex., para tirar dúvidas e compartilhar materiais de apoio), também em 2016 a equipe criou um grupo no Facebook³. Na época, essa rede social apresentava grande aceitação entre alunos e instrutores, havendo igualmente relatos de suas experiências bem sucedidas no suporte ao ensino e aprendizagem [1].

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) [11]. 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 aluno/exercício. Apostando que interatividade e *feedback* poderiam contribuir de forma significativa para a motivação e desempenho dos alunos [18], em 2017 passamos a adotar a ferramenta de correção automática *run.codes*⁴. 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 alunos. [8]. A Figura 3 mostra o resultado de uma pesquisa realizada com egressos da turma de 2017 quanto ao uso do *run.codes*.

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

²<https://moodle.org/>

³<https://www.facebook.com/>

⁴<https://run.codes/>

das notas registradas (menos sujeitas a erro humano) e acelerar a divulgação dos resultados.

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 e 2 meses, respectivamente.

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 alunos que conseguiram finalizar o módulo básico (Figura 2b)⁵. Vale ressaltar aqui que para cada edição a equipe procurou manter os mesmos instrumentos de avaliação e seus níveis de dificuldade.

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 [17] e também por ter passado a ser utilizada tanto na Maratona de Programação⁶ da Sociedade Brasileira de Computação (SBC) quanto na Olimpíada Brasileira de Informática⁷ (OBI). Apesar de tal decisão poder, até certo ponto, suavizar a curva de aprendizagem, outra ponderação recorrente dos alunos 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 Notebook⁸, que permite aos instrutores confeccionar material didático interativo, e aos alunos interagir e modificar os exemplos de código-fonte fornecidos em seus próprios computadores. Essa estratégia interativa e baseada em exemplos [15] trouxe mais agilidade à resolução dos exercícios propostos, ao esclarecimento de dúvidas e, conseqüentemente, à aprendizagem prática dos alunos.

Também, com o sucesso do Micro:bit⁹ – 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 MakeCode¹⁰, um editor on-line de programação em bloco [2] que permite a criação e simulação de projetos do Micro:bit e Mu¹¹, 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 no artigo, 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 Pokémon¹². O desafio foi recebido com entusiasmo e motivou os alunos a aplicarem os conceitos vistos durante o curso.

O quanto o `run.codes` te ajudou na realização dos Workings?

20 respostas

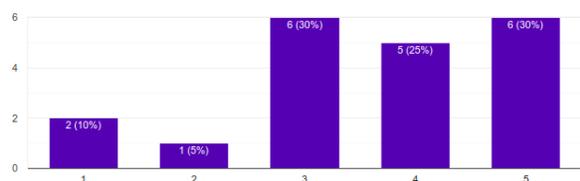


Figura 3: Pesquisa realizada com egressos da turma de 2017 quanto ao uso do `run.codes`. O índice na escala Likert de 5 pontos reflete a intensidade do apoio.

⁵Para os anos de 2018 e 2019, o gráfico da Figura 2b mostra o número de aprovados nos módulos básicos de cada ano.

⁶<http://maratona.sbc.org.br>

⁷<https://olimpiada.ic.unicamp.br>

⁸<https://jupyter.org/>

⁹<https://microbit.org/about/>

¹⁰<https://www.microsoft.com/en-us/makecode>

¹¹<https://codewith.mu/>

¹²<https://www.pokemon.com>

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 [17]. 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 [24], Python é a terceira linguagem mais popular do mundo, perdendo apenas para 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 Santana et al. [22], Rodrigues [21] e Mannila et al. [14]. 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 [20].

Jupyter¹³ 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 Arduino¹⁴ 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 alunos 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 nas Figuras 4 e 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, desde o entendimento do que são algoritmos, estruturas condicionais, estruturas de repetição até o modularização de código. Já o módulo avançado, além de entrar em alguns conceitos mais elaborados de programação, como orientação a objeto, ele aborda temas com grande apelo pelos jovens atuais, como programação de jogos e inteligência artificial.

Ressaltamos que o desenvolvimento dos notebooks é contínuo. Desde que começamos a utilizar o Jupyter, para cada aula, ouvimos os *feedbacks* dos alunos e também dos instrutores. 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 alunos frente ao conteúdo (nível de dificuldade), etc.; (ii) a do aluno, 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.

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 aluno 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 aluno: (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

¹³<https://jupyter.org>

¹⁴<https://www.arduino.cc/>



O que vamos aprender nessa aula:

1. [O que é um programa?](#)
2. [Python](#)
 - 2.1 [Histórico](#)
 - 2.2 [Programas em Python](#)
 - 2.3 [Documentação](#)
3. [Variáveis](#)
4. [Atribuição de valores](#)
5. [Tipos de dados](#)
6. [Operações com números](#)

Figura 4: Trecho de um notebook com o cabeçalho de uma aula.

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.

```

1: # 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).

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 curso para alunos do ensino fundamental e para professores da rede de ensino pública estadual e federal do estado (visando formar multiplicadores do curso). Com a ajuda de integrantes externos, denominados internamente de *trainees*, seriam gerenciados três cursos simultaneamente.

Todavia, com o contexto da pandemia tivemos, aos poucos, que repensar nossos planos. Inicialmente o calendário foi suspenso e aguardamos uma posição formal da Ufes e da SEDU. A incerteza ao longo do tempo acarretou no cancelamento primeiramente do curso de formação de professores (devido ao provável cenário conturbado da educação na volta às aulas) e na realização do módulo básico (para o ensino médio) de forma *online*, enquanto o curso especial para alunos do fundamental e módulo básico para o ensino médio seriam mantidos presenciais. Nesse cenário, o principal desafio seria reformatar o módulo avançado, e a nova experiência em uma edição especial para alunos do fundamental.

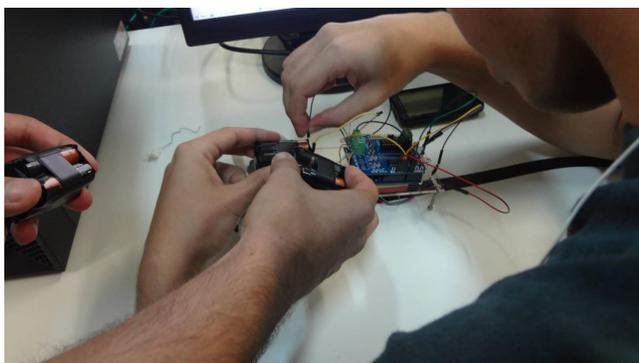


Figura 6: Alunos configurando o dispositivo Micro:bit.



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

Em meados de junho ainda não existia nenhuma previsão concreta para a volta às aulas presenciais e o cenário se tornara cada vez menos favorável. Por conseguinte, decidimos cancelar os cursos que ocorreriam presencialmente e concentrar os esforços em um curso totalmente *online* e remoto, atendendo o público tradicional – alunos 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: Jitsi¹⁵; BigBlueButton¹⁶; Webex¹⁷; e Riot¹⁸. Em seguida, definimos alguns critérios de interesse para nosso caso de

¹⁵<https://jitsi.org/>

¹⁶<https://bigbluebutton.org/>

¹⁷<https://www.webex.com/>

¹⁸<https://riot.im/>

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 entretenimento¹⁹, 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 visualizações²⁰. 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.

Nesse contexto, consideramos a plataforma Twitch²¹, principalmente por seu foco em *live streamings* de video 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 [6]. 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 instrutores e alunos. 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 Discord²², 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 alunos. 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, utilizamos o Streamlabs OBS²³, um programa que permite personalizar o *stream* ao vivo com a seleção de cenas, recursos audiovisuais, etc. de maneira fácil e intuitiva. Além disso, essa ferramenta permite inserir

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

²⁰<https://www.youtube.com/channel/UCG-eghyORxNrzSG8FkAWKyw>

²¹<https://www.twitch.tv/>

²²<https://discord.com/>

²³<https://streamlabs.com/>

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	✓	✓	✓	✓	✓	✓	✓	✓
BigBlueButton	✓	✓	✓	□	✓	✓	✓	✓
Webex	✓	□	□	□	✓	✓	✓	✓
Riot	✓	✓	✓	□	□	✓	✓	□

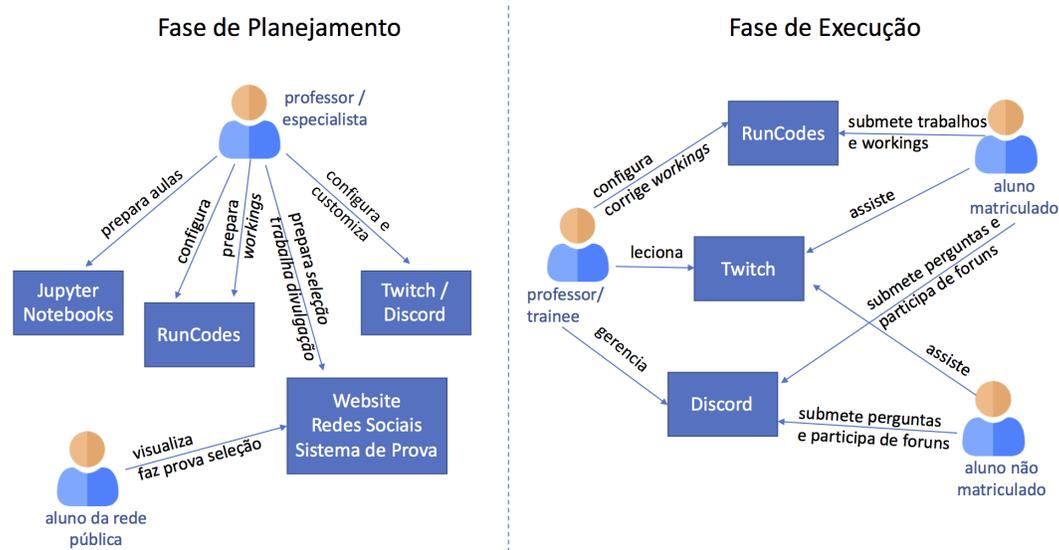


Figura 8: Visão geral das tecnologias envolvidas e os papéis dos usuários.

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 alunos com dúvidas em um dado instante. Com isso, poderemos montar uma transmissão mais elaborada que a tradicional câmera com compartilhamento de tela.

Com tais recursos, projetamos que durante a aula o instrutor permanece em um canal de voz limitado apenas a colaboradores do curso. Abrimos inicialmente um número limitado de canais de voz (na primeira aula foram três), cada qual com no máximo dois estudantes e um colaborador 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, os colaboradores podem eventualmente solicitar ao instrutor uma pausa na aula, seja para os alunos processarem melhor o conteúdo, ou para explicar novamente algo que não tenha ficado claro. Os colaboradores também estabeleceram horários de plantão para atender os alunos e esclarecer dúvidas fora do horário convencional das aulas. Ainda criamos canais destinados à interação entre os próprios alunos do curso e, com auxílio de um

bot, bonificação para os mais ativos. Essa bonificação é uma pontuação 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, acreditamos que esse incentivo poderá ajudar na formação de uma comunidade mais participativa. Apesar da limitação no número de usuários que podem ser registrados no run.codes, compartilhamos os *workings* com todos os alunos e os resolvemos na aula subsequente, de maneira a tornar o curso mais inclusivo e atraente para o público-alvo.

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 duas fases do desenvolvimento do curso: (i) na Fase de Planejamento, na qual o conteúdo do curso é preparado, as ferramentas são customizadas e configuradas, os meios de divulgação são criados e atualizados, a articulação com a SEDU e escolas públicas do estado é efetivada e, finalmente, o processo seletivo é realizado; e (ii) na Fase de Execução, na qual o curso é lecionado aos sábados, o suporte aos alunos é oferecido durante toda a semana, os trabalhos e exercícios são corrigidos e a equipe se reúne semanalmente para acompanhar o andamento do curso.

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 alunos. 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 Instagram²⁴ aumentou mais de 50%.

Ao final do processo seletivo alcançamos o número de 867 inscrições, um aumento de 44,7% em relação ao ano passado. 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 alunos 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 alunos 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 aluno não só se sente confortável em interagir durante a aula, como também valoriza o conteúdo e se concentra.

Com relação à segunda etapa do processo seletivo, coincidentemente, desde 2019 vinha sendo desenvolvido um novo sistema de provas com base em uma arquitetura mais escalável, robusta e portátil. Essa evolução foi planejada tendo em vista uma futura ampliação do projeto, e possível expansão para o interior do estado. Seria necessário com isso que o sistema também pudesse ser executado remotamente (em sua versão original, o sistema só podia ser executado em rede local).

Para a implementação do novo servidor foi decidido utilizar a tecnologia Docker²⁵ para garantir maior escalabilidade e portabilidade, em que foram criadas duas imagens separadas: 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 AdonisJS²⁶, 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 virtual²⁷ 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).

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.

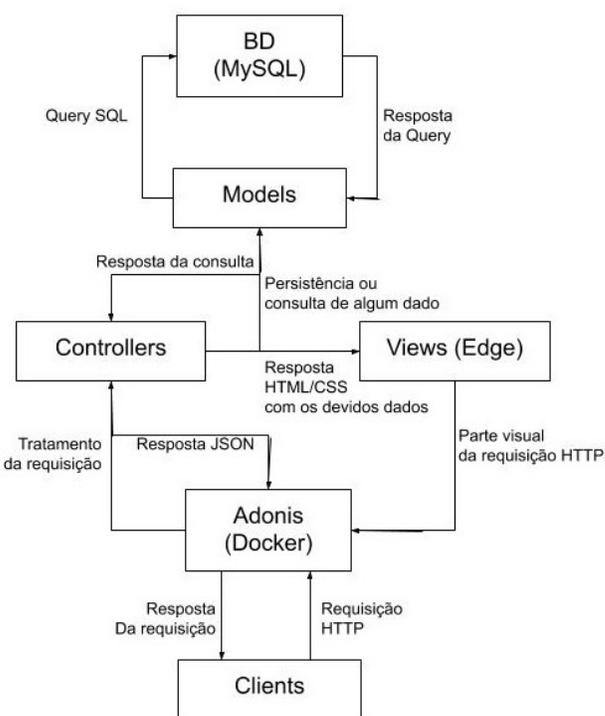


Figura 9: Arquitetura geral do sistema de provas

7 DESAFIOS E EXPECTATIVAS

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. Ao invés de respostas, trazemos ao final deste artigo questões e reflexões sobre o futuro do Introcomp, sintetizadas pela indagação: *e aí, vai dar certo?* Antes de mais nada, é importante reconhecer que são vários (e mesmo desconhecidos) os desafios do curso nesse contexto da pandemia. Primeiramente, a equipe de organizadores, formada em grande parte por alunos voluntários, não possui experiência com *live streaming*. As aulas que antes eram marcadas pelo ambiente informal, descontraído e com alta interação entre alunos e instrutores, agora precisam ser reavaliadas e novas estratégias e métodos de ensino centrados no usuário aplicados para que o curso seja bem

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

²⁵<https://www.docker.com/>

²⁶<https://adonisjs.com/>

²⁷Xeon(R) CPU 2.53GHz, 4 cores, 4GB RAM

sucedido [26]. Os equipamentos que outrora eram disponibilizados pelo projeto na universidade agora não o são mais, exigindo assim de cada aluno que tenha ao menos um computador e uma boa conexão de internet. Há de se considerar também as proporções que podem ser atingidas. Até 2019, o curso era limitado a 80 alunos por ano, porém nesta edição o alcance está sendo muito maior dado que a Twitch e o Discord estão abertos a qualquer interessado (à exceção do run.codes, como mencionado anteriormente). Ademais, apesar de todo o planejamento até o momento, a atração e retenção de alunos certamente nos demandará flexibilidade e aplicação de um conjunto de estratégias ainda a serem definidas.

Apesar da situação parecer não tão favorável quanto em anos anteriores, nossas expectativas para esta nova fase são boas. Nossos colaboradores têm explorado as plataformas adotadas visando ganhar familiaridade com os recursos que podem vir a ser aplicados para melhorar a qualidade do curso. A manutenção do processo seletivo para alunos do ensino médio visa assegurar assistência direta pelo menos no padrão de anos anteriores. Mesmo assim, o fato de as aulas ocorrerem em um ambiente acessível, deixa as portas abertas para o público em geral. Além disso, a possibilidade de ampliação do público-alvo, que antes era restrito a alunos de escolas da Grande Vitória, também nos oferece a oportunidade de alcançar alunos do interior e até mesmo de outros estados. O debate cada vez mais presente sobre o que será o “novo normal” nos faz ponderar e investir ainda mais energias na edição atual, sem obviamente pensar em abolir o modelo presencial pelo papel social que exerce no estado do Espírito Santo. De todo modo, temos certeza que essa experiência 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.

AGRADECIMENTOS

Agradecemos à Pró-reitoria de Extensão da UFES (ProEX/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. Os autores também agradecem aos estudantes André Oliveira Cunha e Gabriel Castro de Rezende pelas contribuições para a realização deste trabalho.

REFERÊNCIAS

- [1] Gersica Agripino Alencar, Murilo Rebouças Moura, and Ricardo Barbosa Biten-court. 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 (2013), 86–93.
- [2] Viviane Cristina Oliveira Aureliano and PCAR Tedesco. 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*. 10.
- [3] Lediana Dalla Costa, Daniela Tavares Camera, Ana Maria Zeferino, Anderson Kaliniski, Marcela Gonçalves Trevisan, and Franciele do Nascimento Santos Zonta. 2020. Análise da vulnerabilidade entre estudantes da rede pública e privada. *Rev. Saúde Pública Paraná (Online)* (2020), 108–119.
- [4] Emanuel Coutinho, Mara Bonates, and Leonardo Oliveira Moreira. 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. 689.
- [5] Carina Machado de Farias, Anderson S. de Oliveira, and Everton Dias de A. Silva. 2018. Uso do Scratch na Introdução de Conceitos de Lógica de Programação: relato de experiência. In *Anais do XXVI Workshop sobre Educação em Computação* (Natal). SBC, Porto Alegre, RS, Brasil. <https://doi.org/10.5753/wei.2018.3491>
- [6] Travis Faas, Lynn Dombrowski, Alyson Young, and Andrew D Miller. 2018. Watch me code: Programming mentorship communities on twitch. tv. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–18.
- [7] Ana Carolina Ferreira, Juliana Santos, Raul Silva, Allan Thales Ramos Oliveira, Diego Zobot, Débora Abdalla, and Eivaldo Matos. 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. 1306.
- [8] Luiz Otavio Gerhardt, Isabel MR Bustamante, Luiz Felipe F Mai, Nicole R Nunes, Lucio Sandrini, Gabriel C Rezende, Gustavo S Alochio, and Roberta L Gomes. 2018. Estudo de Ferramentas de Apoio à Correção de Atividades de Programação no Contexto do Projeto IntroComp. (2018), 10.
- [9] Mark Guzdial. 2009. Education Teaching Computing to Everyone. *Commun. ACM* 52, 5 (May 2009), 31–33. <https://doi.org/10.1145/1506409.1506420>
- [10] Dianne Hagan and Selby Markham. 2000. Does It Help to Have Some Programming Experience before Beginning a Computing Degree Program? *SIGCSE Bull.* 32, 3 (July 2000), 25–28. <https://doi.org/10.1145/353519.343063>
- [11] Caitlin Kelleher and Randy Pausch. 2005. Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers. *ACM Comput. Surv.* 37, 2 (June 2005), 83–137. <https://doi.org/10.1145/1089733.1089734>
- [12] A. Kemppainen, M. Fraley, A. Hamlin, and G. Hein. 2016. How important is high-school computing experience for first-year engineering student success?. In *2016 IEEE Frontiers in Education Conference (FIE)*. 1–5.
- [13] Hong Cheong Looi and Afzaal H Seyal. 2014. Problem-based learning: An analysis of its application to the teaching of programming. *International Proceedings of Economics Development and Research* 70 (2014), 68.
- [14] Linda Mannila, Mia Peltomäki, Ralph-Johan Back, and Tapio Salakoski. 2006. Why Complicate Things? Introducing Programming in High School Using Python. *Conferences in Research and Practice in Information Technology Series* 52 (01 2006).
- [15] Luana Müller, Milene Selbach Silveira, and Clarisse Sieckenius de Souza. 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 (Belém, Brazil) (IHC 2018)*. ACM, New York, NY, USA, Article 17, 10 pages. <https://doi.org/10.1145/3274192.3274209>
- [16] Sin Chun Ng, Steven O Choy, Reggie Kwan, and SF Chan. 2005. A web-based environment to improve teaching and learning of computer programming in distance education. In *International Conference on Web-based Learning*. Springer, 279–290.
- [17] Uolevi Nikula, Jorma Sajaniemi, Matti Tedre, and Stuart Wray. 2007. Python and roles of variables in introductory programming: experiences from three educational institutions. *Journal of Information Technology Education: Research* 6, 1 (2007), 199–214.
- [18] Elizabeth Odekir-Hash and Joseph L Zachary. 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*. 55–59.
- [19] Walter Priesnitz Filho, Ilse Abegg, and Eugenio de Oliveira Simonetto. 2012. Uma abordagem diferenciada no ensino de algoritmos através da utilização de uma lousa digital. *Revista GEINTEC-Gestão, Inovação e Tecnologias* 2, 2 (2012), 129–137.
- [20] Atanas Radenski. 2006. “Python first”: a lab-based digital introduction to computer science. In *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ITiCSE 2006, Bologna, Italy, June 26–28, 2006*, Renzo Davoli, Michael Goldweber, and Paola Salomoni (Eds.). ACM, 197–201. <https://doi.org/10.1145/1140124.1140177>
- [21] Rivanilson S. Rodrigues. 2014. Ensino de algoritmos e linguagem de programação no nível médio: um relato de experiência.
- [22] Bianca Santana, José Solenir Figuerêdo, and Roberto A. Bittencourt. 2017. Motivação de Estudantes Non-Majors em uma Disciplina de Programação. In *Anais do XXV Workshop sobre Educação em Computação* (São Paulo). SBC, Porto Alegre, RS, Brasil. <https://doi.org/10.5753/wei.2017.3545>
- [23] Rafael Santos, Breno Sousa, Alberto Raiol, Paulo Cerqueira, and Fábio Bezerra. 2019. Uma Proposta de Método de Ensino e Relatos de Experiências com a Robótica Educacional. In *Anais do XXVII Workshop sobre Educação em Computação* (Belém). SBC, Porto Alegre, RS, Brasil, 111–120. <https://doi.org/10.5753/wei.2019.6622>
- [24] TIOBE.com. 2020 (acessado em 23 de setembro de 2020). *TIOBE index*. <https://www.tiobe.com/tiobe-index>
- [25] Flávio Varejão. 2016. *Introdução à Programação: Uma Nova Abordagem Usando C*. Elsevier Brasil.
- [26] Jamie Woodcock and Mark R Johnson. 2019. Live streamers on Twitch. tv as social media influencers: Chances and challenges for strategic communication. *International Journal of Strategic Communication* 13, 4 (2019), 321–335.