



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ
CURSO DE DESIGN DIGITAL

JOÃO PEDRO DEDÊ DA SILVA

EFEITOS DO USO DA FERRAMENTA PYTHON TUTOR EM DISCIPLINAS
INTRODUTÓRIAS DE PROGRAMAÇÃO

QUIXADÁ
2023

JOÃO PEDRO DEDÊ DA SILVA

EFEITOS DO USO DA FERRAMENTA PYTHON TUTOR EM DISCIPLINAS
INTRODUTÓRIAS DE PROGRAMAÇÃO

Monografia apresentada no curso de Design Digital da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Design Digital. Área de concentração: Programas interdisciplinares e certificações envolvendo Tecnologias da Informação e Comunicação (TIC).

Orientador: Profa. Dr^a. Ingrid Teixeira Monteiro

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S58e Silva, João Pedro Dedê da.
Efeitos do uso da ferramenta python tutor em disciplinas introdutórias de programação / João Pedro Dedê da Silva. – 2023.
41 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Design Digital, Quixadá, 2023.

Orientação: Prof. Dr. Profa. Dr^a. Ingrid Teixeira Monteiro.

1. Python (Linguagem de programação de computador). I. Título.

CDD 745.40285

JOÃO PEDRO DEDÊ DA SILVA

EFEITOS DO USO DA FERRAMENTA PYTHON TUTOR EM DISCIPLINAS
INTRODUTÓRIAS DE PROGRAMAÇÃO

Monografia apresentada no curso de Design Digital da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Design Digital. Área de concentração: Programas interdisciplinares e certificações envolvendo Tecnologias da Informação e Comunicação (TIC).

Aprovada em: __/__/__.

BANCA EXAMINADORA

Profa. Dr^a. Ingrid Teixeira Monteiro (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Jefferson de Carvalho Silva
Universidade Federal do Ceará (UFC)

Prof. Dr. David Sena Oliveira
Universidade Federal do Ceará (UFC)

Aos meus amigos, minha família e meus orientadores.

AGRADECIMENTOS

Deixo meus sinceros agradecimentos àqueles que me acompanharam nessa jornada, sempre dedicando seu apoio.

Agradeço aos meus orientadores, Prof. Aníbal Cavalcante de Oliveira e a Profa. Ingrid Teixeira Monteiro, que me guiou nesse caminho. Deixo um especial agradecimento em memória do Prof. Aníbal Cavalcante de Oliveira, que descanse em paz e que esteja vendo que finalmente alcançamos essa defesa.

Agradeço aos professores que acompanharam as bancas, sempre aperfeiçoando meus conhecimentos e habilidades sobre os conteúdos, muito obrigado Prof. Jefferson de Carvalho Silva, Profa. Rainara Maia Carvalho e Prof. David Sena Oliveira.

E por fim, mas não menos importante, aos meus amigos e família, que me apoiaram em continuar nessa jornada quando tudo parecia obscuro.

RESUMO

A área de desenvolvimento vem se popularizando mais e mais no mercado, e a necessidade de profissionais qualificados é cada vez maior, porém é visto uma grande dificuldade no aprendizado da lógica inicial de programação, causando um grande número de reprovações. Para prevenir tal efeito, muitos estudos se voltam para métodos que possam auxiliar no ensino de programação, diminuindo essa barreira de aprendizagem que muitos enfrentam, um desses métodos é a utilização de ferramentas de visualização de dados para deixar mais figurativo o entendimento sobre o conteúdo. Essa pesquisa tem como objetivo enxergar os efeitos que tais ferramentas podem causar no cenário de disciplinas introdutórias de programação. Outros estudos na área, apontam que principalmente nas turmas iniciais de programação, onde o conhecimento ainda está sendo moldado, ferramentas de visualização de dados têm maior efeito, sendo apresentado como um método de ensino mais assertivo para combater a dificuldade de entendimento da lógica de programação.

Palavras-chave: lógica de programação; visualização de dados; ferramentas visuais;

ABSTRACT

The field of development is increasingly gaining popularity in the market, and the demand for qualified professionals is on the rise. However, there exists a significant challenge in grasping the fundamentals of initial programming logic, leading to a high rate of academic setbacks. To mitigate this issue, numerous studies are exploring methods to facilitate programming education, aiming to reduce the learning barriers encountered by many learners. One such approach involves leveraging data visualization tools to enhance the conceptual understanding of programming content. This research seeks to examine the impact of these tools in the context of introductory programming courses. Previous studies in the field highlight that data visualization tools are particularly effective in the early stages of programming education when foundational knowledge is being established, and serve as a more robust instructional method to address challenges in comprehending programming logic.

Keywords: logic programming; data visualization; visualization tools.

LISTA DE FIGURAS

Figura 1 –	Exemplo de visualização do código no Python Tutor, presente na página inicial da aplicação	13
Figura 2 –	Processo de interação com o código sendo demonstrado no Python Tutor	14
Figura 3 –	Processo subsequente de interação com o código sendo demonstrado no Python Tutor	22

LISTA DE ABREVIATURAS E SIGLAS

H0 Hipótese nula

H1 Hipótese 1

SUMÁRIO

1	INTRODUÇÃO	13
2	OBJETIVOS	15
2.1	Objetivo Geral	15
2.2	Objetivos Específicos	15
3	TRABALHOS RELACIONADOS	16
3.1	Online Python tutor: Embeddable web-based program visualization for CS education	16
3.2	Uso do Scratch na Introdução de Conceitos de Lógica de Programação: relato de experiência	17
3.3	Aprendizagens em movimento: Um relato de experiência de Prática Docente do Pensamento Computacional através de M-Learning e U-Learning	18
3.4	Comparação	18
4	FUNDAMENTAÇÃO TEÓRICA	20
4.1	Ferramentas de programação visual	20
4.2	Python Tutor	21
4.3	Testes experimentais no contexto do Python Tutor	24
5	PROCEDIMENTOS METODOLÓGICOS	25
5.1	Definição da hipótese	25
5.2	Definição das variáveis do experimento	25
5.3	Riscos envolvidos na hipótese	26
5.4	Design do experimento	27
5.5	Estabelecer comunicação com um professor de uma turma de programação introdutória	27
5.6	Realizar teste piloto	28
5.7	Fazer uma pesquisa inicial para entender o perfil da turma	29
5.8	Definir um conteúdo para aplicar com a turma	29
5.9	Dividir os grupos em que serão aplicados os testes	29
5.10	Aplicar um teste para colher o resultado da turma	30

6	RESULTADOS.....	31
7	DISCUSSÃO DOS RESULTADOS	38
8	CONCLUSÃO.....	39
9	TRABALHOS FUTUROS	39
	REFERÊNCIAS	40

1 INTRODUÇÃO

Com o aumento da demanda no campo da tecnologia e informação, profissionais capazes de resolver problemas no âmbito computacional são cada vez mais necessários (Santos, 2006, Wangenheim, Nunes e Santos, 2014), porém, até a atualidade, disciplinas introdutórias de programação apresentam altos índices de reprovação (Oliveira et al, 2021, Bosse e Gerosa, 2015, Carvalho e Alves, 2018, Guedes 2014).

Uma das possíveis causas levantadas para essas ocorrências é a dificuldade de aprendizagem da lógica de programação, devido à natureza abstrata das matérias e a falta de preparo básico em lógica matemática por parte dos alunos. (Farias, Oliveira e Silva, 2018).

Buscando soluções para o problema, estudos na área defendem o uso de ferramentas que auxiliem o estudante a ter uma visualização e prática mais amigáveis das linguagens de programação aplicadas da forma tradicional (Farias, Oliveira e Silva, 2018, Gomes e Melo, 2013).

A programação visual, por exemplo, é uma das formas apontadas em diversas pesquisas para auxiliar o aprendizado da lógica de programação, e se constitui em um método que apresenta as sequências lógicas de um algoritmo de forma visual (Sonza, Batista e Barbosa 2016). A ferramenta Python Tutor¹ é um exemplo de como a escolha de uma ferramenta de programação visual pode influenciar nas dinâmicas de ensino de programação em matérias introdutórias (Guo, 2013).

De acordo com Guo (2013), a ferramenta Python Tutor busca resolver o problema de professores e alunos em disciplinas introdutórias de programação nos cursos de ciência da computação, trazendo para esse cenário uma ferramenta que apresenta algoritmos escritos na linguagem de programação Python de forma gráfica. Além disso, ela pode ser usada a partir de um navegador.

Em seus estudos, Guo (2013) apresenta que após aplicação em turmas de nível introdutório de ciência da computação, obtiveram-se resultados expressivos de uso da ferramenta Python Tutor.

¹ Disponível em: Acesso em: 22 de nov, 2023.

Outras ferramentas presentes em estudos na área (Farias, Oliveira e Silva, 2018, Gomes e Melo, 2013) são o Scratch² e o App Inventor³, que têm como princípio a utilização de blocos de programação para facilitar o uso e aprendizado da linguagem de programação pelos usuários (Papadakis et al, 2014). Papadakis et al. (2014) explicam que ambas as alternativas são viáveis para auxiliar na metodologia de ensino de programação, pontuando que o Scratch mostra-se melhor para ser apresentado em turmas com alunos com menos experiência, enquanto o App Inventor pode ser mais bem utilizado com usuários que estão mais próximos de migrar para uma linguagem de programação mais tradicional.

Nas pesquisas de Farias, Oliveira e Silva (2018), com a aplicação do Scratch em turmas introdutórias, são apresentados testes relevantes sobre o uso da ferramenta, que apesar de não ter atingido os resultados desejados, mostra a importância e interesse nos estudos voltados a esse campo de pesquisa.

Em uma revisão sistemática de literatura sobre o App Inventor, Costa e Piedade (2021) pontuam que apesar de terem identificado muitos estudos, poucos apresentam pesquisas com o público do ensino médio e secundário, focando mais no ensino superior, além da ausência de trabalhos em âmbito nacional, mostrando a preocupação e interesse em elaborações de pesquisas com essa temática.

Para obter resultados que comprovem os efeitos de ferramentas de visualização de dados no ensino de programação, é indicado o uso de pesquisas experimentais, que tem como foco a elaboração de experimentos controlados (Lazar et al, 2017).

Lazar et al (2017) indicam que pesquisas experimentais têm como característica entender o efeito de um elemento “X” que interage com outro elemento “Y” de modo a causar um efeito. É possível apresentar como exemplo dessa relação o uso de uma ferramenta como o Python Tutor, que introduzida em uma turma de programação, pode ou não gerar um efeito positivo.

Sendo assim, para a aplicação de uma pesquisa experimental, é preciso inicialmente se basear em uma hipótese, que possa ser defendida através do experimento (Lazar et al, 2017).

Tendo em vista os projetos apresentados (Farias, Oliveira e Silva, 2018, Gomes e Melo, 2013, Guo, 2013), nota-se o interesse em abordar pesquisas com ferramentas que se distanciam dos métodos tradicionais de ensino de programação, tal interesse também é

2 Disponível em: <https://appinventor.mit.edu/>. Acesso em: 22 nov. 2023.

3 Disponível em: <https://scratch.mit.edu/>. Acesso em: 22 nov. 2023.

transmitida por Costa e Piedade (2021), que pontua a falta de análises experimentais em âmbito nacional, referente a tais ferramentas.

Dessa forma, salientando a necessidade de pesquisas que apresentem os efeitos de ferramentas de visualização de código usadas com alunos de programação, esse projeto busca atingir o objetivo de apresentar resultados que mostram os efeitos da ferramenta Python Tutor, inserida no contexto de aulas introdutórias de programação, utilizando um experimento para a obtenção de resultados que comprovam ou refutam uma determinada hipótese.

2 OBJETIVO

2.2 Objetivo Geral

Aplicar um experimento focado no uso da ferramenta Python Tutor, no contexto de disciplinas introdutórias de programação no Campus da UFC. Nossa hipótese é que o Python Tutor como ferramenta de visualização de dados consegue melhorar o entendimento dos alunos em relação a alguma estrutura de programação básica, como arrays, strings e laços.

2.2 Objetivos Específicos

- Entender se os conteúdos escolhidos para o experimento são bem exemplificados com o Python Tutor.
- Identificar dificuldades enfrentadas pelos alunos durante as explicações do conteúdo.
- Realizar um experimento controlado, com a implantação do Python Tutor em sala de aula.
- Reconhecer se existem efeitos positivos na aplicação do Python Tutor em conteúdos introdutórios de programação.

3 TRABALHOS RELACIONADOS

3.1 Online Python tutor: Embeddable web-based program visualization for CS education.

O Python Tutor é uma ferramenta que busca tornar fácil a visualização de códigos para o auxílio em aulas de programação, tentando oferecer um uso prático e intuitivo (Guo, 2013). De acordo com Guo (2013), devido a sua proximidade com cursos de CS1 (introdução à ciência da computação), seus estudos com a ferramenta foram voltados a turmas iniciais desse curso, onde ele relata ter atingido resultados promissores, evidenciando o uso da ferramenta por mais de 200.000 pessoas.

Para Guo (2013), a abstração do código, a dificuldade de professores em criar conteúdos didáticos para as disciplinas introdutórias, a dificuldade dos alunos de entenderem os conteúdos foram fatores motivacionais para a criação do Python Tutor. Com isso Guo (2013) descreve três elementos base que tornam o Python Tutor uma boa alternativa para ser utilizada no ambiente de estudo:

“...**Python**: In recent years, Python has been gaining traction as a preferred language for CS1 courses across many universities...**Web-Based**: Online Python Tutor is the only Python program visualization tool that runs within a web browser without any required software or plugin installation...**Embeddable**: Finally, Online Python Tutor is the only program visualizer that can be seamlessly embedded within webpages...” (Guo, 2013)

Apesar de Guo (2013) mostrar diversos indícios de que sua metodologia e aplicação alcançaram resultados positivos, não estão claros os métodos de validação do potencial da aplicação. Mesmo que Guo (2013) cite em sua pesquisa um tópico de “DESIGN METHODOLOGY”, ele apenas relata um pouco do processo de descobrimento de problemática, escolhas de elementos e ao falar da aplicação de sua pesquisa em contextos reais, ele não aponta uma descrição clara do processo tomado para essa validação:

“We refined our initial design and implemented new features (e.g., nested function and lambda support) by consulting with our users, which includes a diverse group of CS instructors—teaching

assistants, professors at both small and large universities, and an instructor of several free online CS classes with over 200,000 total enrolled students” (Guo, 2013).

Em suma, Guo (2013), baseia-se nos resultados obtidos com o uso de sua aplicação, mas não apresenta de forma prática um método que prove sua hipótese de que o Python Tutor oferece reais benefícios no aprendizado de programação.

3.2 Uso do Scratch na Introdução de Conceitos de Lógica de Programação: relato de experiência.

Nas pesquisas de Farias, Oliveira E Silva (2018), é evidenciada a importância de metodologias que auxiliem a reduzir o alto nível de reprovação em cursos de programação. No projeto em questão, Farias, Oliveira E Silva (2018) dão continuidade à aplicação de uma metodologia de ensino de lógica de programação, em turmas do ensino básico, aplicada no ano de 2016 (Farias et al, 2017). Nesta continuidade do projeto, é utilizado o software Scratch, que é colocado como uma alternativa popular para ser aplicado com pessoas com pouco conhecimento prévio na área de programação (Farias, Oliveira e Silva, 2018).

Ao descrever o processo de aplicação da pesquisa, Farias, Oliveira e Silva (2018) pontuam ter aplicado a primeira etapa do projeto com uma turma de 13 pessoas, ingressantes do primeiro semestre do curso Técnico Subsequente de Informática, do qual 8 alunos concluíram a disciplina com as dinâmicas propostas usando o SCRATCH.

Em outro momento, Farias, Oliveira e Silva (2018) investigam os efeitos da aplicação da metodologia com o SCRATCH no grupo de 8 pessoas participantes, em contraste com um outro grupo de alunos, que juntos, estão ingressando em uma disciplina do segundo semestre, totalizando uma turma de 31 alunos, dos quais foram avaliados 23, já que 8 dos alunos não compareceram às aulas.

Farias, Oliveira e Silva (2018) dividiram a turma em dois grupos, Grupo A, com os 8 alunos que participaram anteriormente do experimento com o SCRATCH e Grupo B, com 15 alunos que ingressaram na turma sem ter participado das dinâmicas com o software.

Por fim, o projeto de Farias, Oliveira e Silva (2018) mostrou notas médias maiores no Grupo B, porém Farias, Oliveira e Silva (2018) defendem que apesar dos dados apontarem isso, o uso da ferramenta SCRATCH ainda deve ser investigada para se obter mais resultados.

3.3 Aprendizagens em movimento: Um relato de experiência de Prática Docente do Pensamento Computacional através de M-Learning e U-Learning

No estudo de Greff, Peres e Bertagnolli (2018), é mostrado um interesse na forma de como o pensamento computacional é abordado nas dinâmicas pedagógicas, tendo como foco o uso do APP Inventor como forma de avaliar a aplicação de metodologias de ensino em turmas de sistemas.

Para seu experimento (Greff, Peres e Bertagnolli 2018) utilizou o critério qualitativo, em um grupo de 15 pessoas adultas, com o intuito de apresentar as dinâmicas pedagógicas possíveis com o App Inventor, e coletar as impressões dos envolvidos. A abordagem focou principalmente na obtenção de resultados com professores, por querer se aproximar de critérios pedagógicos.

Em seus resultados, Greff, Peres e Bertagnolli (2018), perceberam que boa parte do grupo onde foi aplicado a didática não possuía conhecimento prévio sobre a ferramenta APP Inventor, evidenciando que é uma ferramenta pouco utilizada nas práticas pedagógicas dos participantes.

Greff, Peres e Bertagnolli (2018), em suas conclusões, apesar de identificar os fatores positivos da aplicação da dinâmica proposta, pontuam que a diferença de conhecimentos entre o grupo, além do insuficiente tempo para a aplicação, acabou afetando a absorção por parte dos que possuíam menos conhecimento prévio sobre as ferramentas e práticas de programação.

3.4 Comparação

Na Tabela 1, é apresentada uma breve comparação sobre características inerentes aos trabalhos apresentados, tendo em contraponto as definições básicas do presente trabalho. É

possível observar a preocupação com a aplicação em turmas introdutórias de programação, sendo um foco na maioria dos trabalhos.

Além disso, é de comum interesse, metodologias voltadas a utilização de visualização de dados para buscar resultados na melhoria de aprendizado introdutório. Para isso, as ferramentas que se mostram mais comuns se tratam do Scratch e APP Inventor, tendo poucos trabalhos atuais, principalmente nacionais, baseados na aplicação com o Python Tutor.

Tabela 1 - Comparativo entre características dos trabalhos relacionados.

	Trabalho relacionado 1	Trabalho relacionado 2	Trabalho relacionado 3	Este trabalho
Abordou turmas introdutórias de programação	x	x		x
Aplicou um experimento		x	x	x
Apoia a dinâmica de visualização de dados ou pensamento computacional	x	x	x	x
Abordagem qualitativa	x		x	x
Utilizou Python Tutor	x			x

Fonte: Elaborada pelo autor, 2023.

4 FUNDAMENTAÇÃO TEÓRICA

No contexto atual, muitas são as fontes de pesquisa que abordam o assunto de programação visual como uma forma de auxiliar no ensino de programação introdutório (Costa e Piedade, 2021). A aplicação dessa metodologia em turmas que estão começando a tentar entender a lógica de programação é a forma que é defendida por pesquisadores, para que os alunos consigam absorver melhor o conhecimento (Farias et al, 2017).

Aplicações como o Scratch, App Inventor e Python Tutor são ferramentas que tem como intuito, tornar mais amigável a forma como a lógica de programação é tratada (Guo, 2013, Papadakis et al, 2014, Farias, Oliveira e Silva, 2018). De acordo com Guo (2013), o Python Tutor é fruto de uma preocupação observada no contexto de disciplinas introdutórias de ciência da computação, onde professores tinham dificuldade em criar conteúdos e apresentá-los de forma prática para os alunos, e em contrapartida, os alunos possuíam dificuldade de entender a lógica de como os algoritmos funcionam.

Porém em seus estudos, Guo (2013) não tem foco em elaborar experimentos que validem o ponto de vista do usuário. Sendo assim, um ponto que pode reforçar as teses de Guo (2013) seria a elaboração de um experimento que vise apresentar os resultados da ferramenta Python Tutor, em um contexto real, para assim testar uma possível hipótese de que o Python Tutor auxilia no aprendizado de programação em turmas introdutórias.

4.1 Ferramentas de programação visual

A programação visual, como forma de auxílio nas metodologias de ensino, é um fator que atualmente vem sendo muito utilizado e incentivado como uma alternativa para combater os altos índices de reprovações em disciplinas iniciais de programação (Gomes e Melo, 2013, Farias, Oliveira e Silva, 2018). Aplicativos que tem como intuito o uso desse método estão se tornando muito populares e existe um grande anseio por estudos que envolvam essa tecnologia para avaliar os impactos na educação (Papadakis et al, 2014, Farias, Oliveira e Silva, 2018).

Ferramentas como o SCRATCH e APP Inventor já são amplamente utilizadas em pesquisas, já o SCRATCH, em específico, é muito presente em pesquisas nacionais (Papadakis et al, 2014; Farias, Oliveira e Silva, 2018; Costa e Piedade, 2021; Wangenheim, Nunes e Saltos,

2014). Muitos dos pesquisadores defendem o uso do SCRATCH para os níveis de ensino mais introdutórios, já em relação ao APP Inventor, por ser uma plataforma com linguagem em inglês, acaba apresentando menos uso em estudos similares no âmbito nacional, apesar de que muitos estudiosos mostram a importância de experimentos com o APP Inventor, por apresentar uma forma de apresentar os dados diferente do SCRATCH (Gomes e Melo, 2013, Papadakis et al, 2014, Costa e Piedade, 2021).

Em análise dessas ferramentas, Papadakis (2014) mostra que ambas têm grande potencial no que se propõem, e apresentam fatores positivos em diferentes contextos de ensino. Colocando em contraste, a ferramenta Python Tutor apresenta ainda menos estudos no âmbito nacional, apesar de ser apresentada como uma alternativa que apresenta muitos resultados positivos no contexto que se propõe (Guo, 2013).

Nos relatos de Farias, Oliveira e Silva, 2018, é possível notar que apesar da metodologia proposta com o SCRATCH, não é atingido resultado expressivamente positivo, pois na média geral, a turma que teve interação com o SCRATCH, acabou atingindo notas menores que o grupo de controle. Esse cenário foi diferente ao se analisar as notas apresentadas nos conteúdos iniciais, onde o grupo que seguiu a metodologia proposta teve mais destaque. Nos estudos de Papadakis et al (2014), é pontuado que a ferramenta SCRATCH mostra melhores resultados com usuários com menos noção de programação, por ter uma dinâmica mais lúdica, em comparação ao APP Inventor, que já se aproxima mais da lógica de programação em linguagens convencionais.

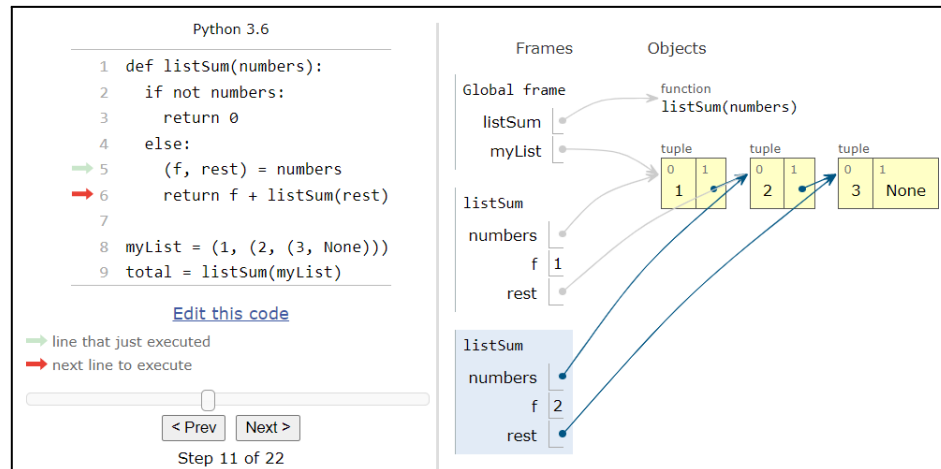
O Python Tutor, por sua vez, tem maior foco em compilação de códigos na linguagem de programação Python, tornando as estruturas codificadas em elementos gráficos, sendo apresentada como uma alternativa para criação de conteúdos didáticos em matérias de programação (Guo, 2013). Sendo assim, sua utilização é muito mais próxima da linguagem convencional, no caso, de PYTHON, que de acordo com Guo (2013), baseou essa escolha por ser uma linguagem muito difundida atualmente.

4.2 Python Tutor

A ferramenta Python Tutor foi desenvolvida por Guo (2013), tendo como foco inicialmente a linguagem Python, tendo como função apresentar a compilação do código de

forma visual e interativa. Sua forma de visualização das estruturas do código é apresentada com auxílio de vetores indicativos (Figura 1).

Figura 1 - Exemplo de visualização do código no Python Tutor, presente na página inicial da aplicação.



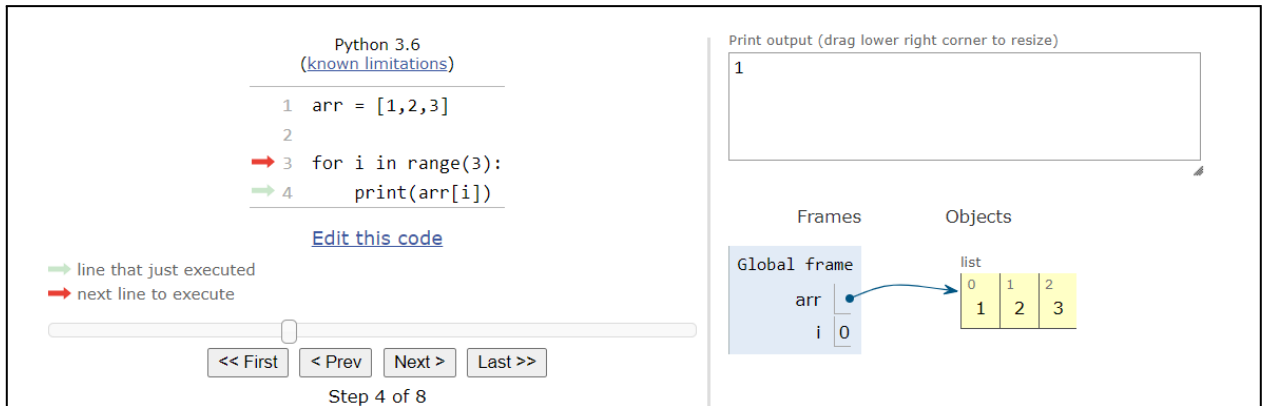
Fonte: Site [Pythontutor.com](http://pythontutor.com)¹, 2023.

De acordo com Guo (2013), o Python Tutor gerou ótimos resultados em turmas introdutórias de ciências da programação, em sua tese ele mostra isso elencando o quão difundido foi seu trabalho, apresentando que aproximadamente 200.000 pessoas utilizaram sua aplicação em um período de três anos. Guo (2013) também pontua que sua aplicação foi usada em contextos reais de sala de aula, por professores e alunos, servindo como ferramenta de aprendizagem.

O Python Tutor tem o conceito de ser facilmente introduzido, por sua metodologia de programação ser baseada em uma ferramenta Web, podendo ser utilizada no próprio navegador, tornando acessível ao usuário começar a programar em Python e já executar seu código (Guo, 2013). Guo (2013), defende que a linguagem Python auxilia na popularização da ferramenta, por ser uma linguagem amplamente utilizada nos dias atuais, apesar que atualmente a ferramenta conta com suporte a outras linguagens, como o JavaScript, C, C++, e Java.

Para Guo (2013), a forma que o Python tutor torna o código Python uma representação visual é uma importante funcionalidade do sistema. A partir da compilação do código, a ferramenta apresenta um componente de interação que permite avançar e retroceder em cada parte da execução do código, permitindo assim a melhor visualização das etapas de compilação (Figura 2).

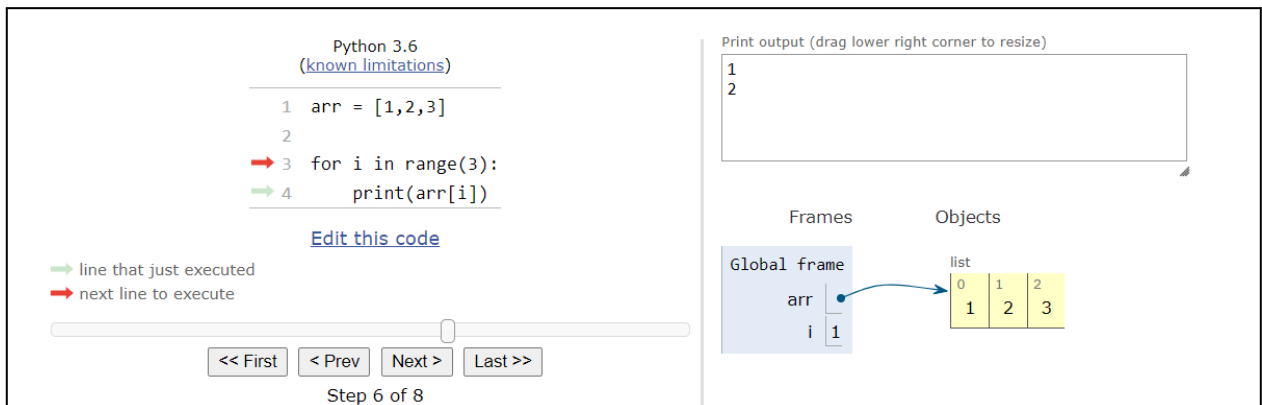
Figura 2 - Processo de interação com o código sendo demonstrado no Python Tutor.



Fonte: Site Pythontutor.com¹, 2023.

Na Figura 3 é mostrada a sequência lógica apresentada pela aplicação Python Tutor para representar a leitura do código em sequência à Figura 2.

Figura 3 - Processo subsequente de interação com o código sendo demonstrado no Python Tutor.



Fonte: Site Pythontutor.com¹, 2023.

Diferente de ferramentas como o SCRATCH e APP Inventor, que tem suas funcionalidades voltadas à programação em blocos (Papadakis et al, 2014), o Python tutor usa uma caixa de texto que compila o código textual de forma convencional, aproximando o usuário da linguagem de programação (Guo, 2013).

Para Guo (2013), sua aplicação apresenta resultados produtivos no ensino de programação, porém em seu projeto, Guo não elenca os métodos utilizados para suas pesquisas, apresentando os resultados como consequência do uso de sua aplicação, porém sem demonstrar os testes baseados em experimentos.

4.3 Testes experimentais no contexto do Python Tutor

Para entender melhor os efeitos que a ferramenta Python tutor pode ter no usuário, é importante a execução de testes experimentais, para adquirir dados que provem ou neguem o efeito da ferramenta (Lazar et al, 2017).

De acordo com Guo (2003), o Python Tutor demonstra bons resultados no auxílio do aprendizado de programação, de forma que seu método de tornar a compilação do código em algo visual facilita o entendimento por parte do usuário. Para comprovar os reais efeitos da ferramenta Python Tutor com o usuário, o primeiro passo é a elaboração da hipótese a ser defendida (Lazar et al, 2017).

Para provar a hipótese, de acordo com (Lazar et al, 2017), é importante que, junto da hipótese, exista uma hipótese nula, para que assim, o experimento possa buscar negar a hipótese nula e provar a hipótese. Aplicando ao contexto do Python Tutor, é importante entender se a ferramenta realmente gera efeitos positivos no aprendizado de programação, ou é indiferente.

Baseado nas argumentações de Guo (2003) o Python Tutor é uma ferramenta, que tem como intuito apresentar uma mudança de hábitos e assim gerar um efeito positivo a um grupo específico, sendo assim, para Lazar et al (2017), a pesquisa que mais se encaixa nesse caso seria a experimental, que tem como foco a criação de um experimento controlado, que aplicado a um grupo, busca encontrar resultados que comprovam ou refutam a hipótese.

Como apresentado por Farias, Oliveira e Silva (2018), para a elaboração de um experimento com o intuito de testar o Python Tutor, é importante abordar uma turma de conteúdo introdutório de programação. Dessa forma pode ser possível acompanhar o desempenho inicial do grupo ao serem apresentados à ferramenta e ao conteúdo.

É importante que o experimento seja feito de forma controlada, para consolidar um resultado claro, buscando mensurar as variáveis que independem do controle do pesquisador, como indica Lazar et al (2017). Para isso é importante ao experimento, entender como pode ser introduzida a ferramenta do Python Tutor, em uma dinâmica de aula.

Também é citado por Lazar et al (2017), que para a defesa da hipótese, é importante o planejamento em relação aos riscos sobre ela, para que o pesquisador consiga diminuir ao máximo as possibilidades de se negar a hipótese de forma errônea.

5 PROCEDIMENTOS METODOLÓGICOS

5.1 Definição da hipótese

O primeiro passo para a aplicação do experimento é a elaboração da hipótese a ser testada, e junto com a hipótese é importante a criação de uma hipótese nula, de forma que o experimento irá buscar testar a hipótese e negar a hipótese nula, para que o objetivo do experimento seja bem sucedido. Caso contrário, se a hipótese nula for provada como verdadeira, a hipótese terá falhado em ser constatada.

Dessa forma foram estabelecidas as seguintes hipóteses:

Hipótese 1(H1): O Python Tutor como ferramenta de visualização de dados consegue melhorar o entendimento dos alunos em relação a alguma estrutura de programação básica.

Hipótese nula(H0): O Python Tutor não apresentou efeitos práticos de melhora no entendimento dos alunos aos conteúdos de programação que serão testados.

5.2 Definição das variáveis do experimento

Para testar a H1 foi preciso identificar como variável o índice de percepção de aprendizagem dos participantes, sendo esse o fator determinante para saber se a ferramenta apresentou algum resultado na interpretação do conteúdo pelos alunos.

O teste terá teor qualitativo, porém para a percepção de aprendizagem foi utilizado uma questão de teor quantitativo para apoiar a avaliação da hipótese. O intuito é junto das experiências subjetivas de cada participante, coletar o dado de que nota eles dão ao conhecimento que eles obtiveram com a oficina e a nota do conhecimento que eles já possuíam antes, o resultado dessas duas questões deve gerar uma média para cada grupo.

Para o critério qualitativo, é importante entender a percepção subjetiva do que cada grupo achou a respeito das oficinas. Sendo assim, para a montagem do formulário foram levadas em consideração as seguintes questões qualitativas:

- A explicação fornecida na oficina, foi suficiente para realizar o exercício? (descreva um pouco)
- O que você achou da ferramenta Python Tutor?(Apenas grupo A)

- A ferramenta Python tutor te ajudou a entender melhor o conteúdo aplicado? (Apenas grupo A)
- Descreva um pouco a sua resposta anterior.(Apenas grupo A)
E para o critério quantitativo foi usado as seguintes questões:
- Você já conhecia o Python Tutor?
- Para fazer os códigos, você utilizou o Python Tutor?(Apenas grupo A)
- O quanto você conhecia sobre array antes da oficina?(em nota de 0 a 10)
- O quanto você acha que conhece agora sobre array depois da oficina?(em nota de 0 a 10)

5.3 Riscos envolvidos na hipótese

Para evitar a chance dos participantes avaliarem a percepção de aprendizado de forma negativa e invalidar a H1, tendo em vista que a amostragem de participantes é pequena para uma avaliação quantitativa, é importante que os critérios qualitativos da pesquisa sejam levados em consideração, levantando assim outras questões que possam apoiar os efeitos da ferramenta no grupo. Como o nível de participação dos integrantes, interesse no conteúdo, percepção sobre a ferramenta.

Para isso, junto com o formulário de percepção, foram elaboradas algumas questões de desafio, para visualizar se a turma se mostra disposta a estar aplicando o conteúdo da oficina. Neste caso foram feitas 4 questões sobre o conteúdo, sendo as três mais simples, e uma questão mais complexa.

- Supondo que possuímos uma lista de strings, que possuem os seguintes valores dentro: "hamburger", "queijo", "alface", "molho". Escreva um código que coloca uma string "pão", no início e no fim da lista.
- Em uma lista que tenha valores numéricos aleatórios, crie um código que remova todos os números menores que 5. Leve em consideração a seguinte lista: [1,5,2,6,3,5,4,19,2,20](QUESTÃO COM CONTEÚDO MAIS COMPLEXO)
- Quais os índices dos valores excluídos na questão anterior?
- Faça um código que imprima o valor e o índice apenas do terceiro valor da lista. Leve em consideração a seguinte lista: ["José", "Maria", "João", "Pedro", "Alfonso"]

5.4 Design do experimento

Após a definição da hipótese, foi dado início a elaboração dos passos que serão tomados para aplicar o experimento, que como indicado na literatura, para casos de pesquisas experimentais, é importante que o método que deve ser seguido seja um experimento controlado, que para essa pesquisa será feito da seguinte forma:

- Escolha de um grupo de alunos de uma disciplina introdutória de programação, que será dividido em dois subgrupos, para que se estabeleça um controle nos testes.
- Analisar o perfil dos integrantes do grupo.
- Elaboração de conteúdos que sejam baseados em array e laço, que é estudada em matérias introdutórias de programação.
- Elaborar uma forma de apresentar o conteúdo com a aplicação da ferramenta Python tutor, e outra forma sem a ferramenta.
- Aplicação de 4 desafios para entender o nível de participação e entendimento da turma.
- Aplicar um teste que visa entender os resultados do experimento no grupo de forma quantitativa e qualitativa.
- Sintetizar os resultados.

O experimento controlado tem como intuito testar a hipótese, enquanto busca negar a hipótese nula.

Foi necessário a divisão de dois grupos, A e B, onde o grupo A recebeu a oficina com o auxílio da ferramenta e o grupo B será o grupo de controle, que receberá a oficina sem a ferramenta.

5.5 Estabelecer comunicação com um professor de uma turma de programação introdutória

Para que o experimento ocorra de forma adequada, é necessário a interação com uma figura pedagógica, que irá acompanhar e aprovar o método que será utilizado com o grupo, de forma que não interfira no planejamento original de como a disciplina foi elaborada.

O intuito do experimento é colher resultados de como a ferramenta Python Tutor pode gerar resultados em um cenário real de aprendizado de programação, sem interferir na

metodologia de aula que seria previamente aplicada, apenas servindo de ferramenta para a interpretação dos usuários em relação ao código.

O contato com a figura do professor, é importante para ter a apresentação de quem será o grupo do experimento e entender qual metodologia será aplicada na disciplina, de forma que possa ser definido o momento em que será aplicado o experimento propriamente dito. Sendo mais adequado a aplicação após os alunos já

terem visto o conteúdo, para que não haja interferência na metodologia aplicada pelo professor.

No caso deste experimento, foram utilizadas duas turmas, uma de redes de computadores, e outra de engenharia mecânica para o teste final, com a média de 45 alunos entre elas.

5.6 Realizar teste piloto

Para prever erros, é importante a execução de um teste piloto, dessa forma será possível analisar melhor o comportamento dos grupos em relação a metodologia e ajustar detalhes que não haviam sido previstos.

O piloto deve consistir de uma aplicação nos mesmos padrões da aplicação real, e seu resultado deve ser logo após descartado, não servindo para a análise real da hipótese.

A execução do piloto seguiu com a turma de redes de computadores, um semestre antes da aplicação do teste final, buscando prever falhas que poderiam acontecer durante o experimento.

Foi testado parâmetros de aplicação e organização do experimento, como tempo da oficina, organização e separação dos grupos e o tempo para aplicação do exercício e formulário ao final da aplicação. Também foi testado a recepção dos integrantes do grupo ao formulário de definição de perfil.

5.7 Fazer uma pesquisa inicial para entender o perfil da turma

Após ter o entendimento de qual turma foi utilizada no experimento, e estar alinhado com a figura pedagógica que acompanhará o experimento, é importante estabelecer um primeiro momento de pesquisa, onde foi apresentado um questionário para o grupo.

O questionário consistiu de um conjunto de perguntas a respeito das experiências e conhecimentos prévios de cada membro do grupo, para que dessa forma fosse possível entender melhor o perfil de cada um e analisar se existiria algo que pudesse gerar alguma variável para o experimento.

Para a definição desse experimento, foi preciso coletar os seguintes dados:

- Em que curso está matriculado?
- Em que ano entrou no curso?
- Qual a experiência que possui com programação?
- Qual a idade?
- Qual o nome?

5.8 Definir um conteúdo para aplicar com a turma

Também após a interação com o professor responsável pelo grupo, foi definido o conteúdo da disciplina utilizado como apoio para o experimento, de forma que o experimento fosse aplicado apenas em uma parte da matéria, e não em todo o conteúdo.

Dessa forma foi analisado o efeito da ferramenta em um determinado cenário no grupo, permitindo que o teste tivesse menos variáveis que pudessem afetar o resultado, mas em contrapartida gerando um resultado mais específico.

Para a aplicação, foi decidido utilizar o conteúdo de Array e Laço, já que é um conteúdo relativamente complexo e onde o Python Tutor pode mostrar melhor as estruturas de programação a serem exemplificadas.

5.9 Dividir os grupos em que serão aplicados os testes

Para a execução do experimento, é preciso que exista a ideia de um grupo de controle, que consiste em um grupo em que a ferramenta Python tutor não será utilizada, e outro

que terá contato com a ferramenta, para que dessa forma seja entendido os efeitos de ambos os casos, e assim possibilitando ter resultados que possam gerar algum contraste.

A escolha de como os grupos foram separados foi tomada em cima do formulário que foi passado, onde se definia o perfil dos participantes definidos pela afinidade que eles tinham previamente com a programação. Dessa forma, foi buscado tentar deixar os dois grupos mais equilibrados possível em relação ao que já sabiam.

5.10 Executar uma oficina e aplicar formulário de coleta

Foi feita uma oficina com os dois grupos, ambas utilizando o mesmo slide com o conteúdo definido, porém uma se usufruindo da ferramenta Python Tutor para auxiliar nas explicações, e a outra usando apenas compilação de código básica.

Para finalizar o experimento, foi feita a aplicação das 4 questões de desafio e do formulário de percepção de aprendizado. É importante que para essa coleta, seja feita após a aplicação da oficina, e é necessário acompanhar os integrantes durante a execução, para tirar dúvidas e incentivar a resolução do formulário.

Só após a conclusão por parte do grupo A, deve ser seguido para a aplicação da oficina no grupo B.

6 RESULTADOS

A condução do experimento foi feita entre os meses de outubro e novembro de 2023, em uma turma do primeiro semestre de Engenharia Mecânica, tendo sido executado um teste piloto entre maio e junho de 2023, com uma turma de Redes de Computadores, com a maioria dos alunos do primeiro semestre. O teste para essa aplicação teve critério qualitativo, e focou em entender o entendimento da turma em relação a oficina aplicada baseada no conteúdo de *Array* e *While*, com e sem a ferramenta Python Tutor.

Ambas as turmas, do experimento e o piloto estavam utilizando da linguagem de Python previamente em suas aulas. Porém é importante pontuar que os professores de ambas as turmas não faziam uso da ferramenta Python Tutor.

Durante o piloto, inicialmente foram obtidos poucos resultados, tendo nas primeiras semanas, apenas 5 pessoas da turma respondido ao formulário inicial para a definição de perfil. Após ser interpretado a falta de interesse pela turma, foi solicitado a ajuda da figura do professor para motivar os alunos a responderem o formulário mesmo que fosse com a negativa de participação do teste, para que fosse constatado se o grupo estava demonstrando não querer aderir ao teste. Porém apenas poucos dias antes do teste que foi possível fechar 26 formulários preenchidos.

No dia da aplicação a turma foi dividida em dois grupos de 13 pessoas, além disso, durante a aplicação, mais alguns alunos preencheram o formulário de perfil com atraso, porém os mesmos não foram incluídos no resultado do experimento. No dia os professores não acompanharam o experimento.

No dia da aplicação, decorrente a problemas de pontualidade dos alunos e falta de apoio durante o teste, foi perdido parte do tempo da aula para organizar a turma. Então com a divisão dos grupos, foi iniciado a aplicação da oficina onde inicialmente foi feito com a turma que receberia o conteúdo com o Python Tutor, após a aplicação foi instruído a turma que fosse feito o exercício e formulário para coleta dos dados de percepção.

Contudo, devido ao tempo que foi perdido após a aplicação, a turma ficou sem orientação para executar o exercício enquanto era aplicada a oficina na outra turma. Por conta disso, foi reparado que os integrantes do grupo A se dispersaram, pegando suas coisas para se evadirem da sala de aula.

Após a aplicação no grupo B, foi tomada a atitude de ficar junto da turma, porém nenhum membro se mostrou disposto a executar o exercício apesar das orientações. Com isso, os integrantes solicitaram fazer o exercício e formulário em um momento posterior. No dia seguinte se obteve apenas 5 formulários concluídos, impossibilitando a análise dos dados por falta de amostragem significativa.

Após o piloto, na aplicação do teste real, foi utilizada uma turma de cerca de 50 alunos, do curso de Engenharia Mecânica, onde participaram 24 alunos, divididos em 14 para o grupo A que teve a utilização da ferramenta Python Tutor e 11 para o grupo B de controle. Para manter o interesse da turma a Figura do professor foi importante tendo em vista a aplicação do teste piloto.

Seguiu-se com a aplicação do formulário para a obtenção dos perfis do grupo, com o apoio do professor responsável, nessa aplicação foram alcançadas 28 respostas por parte da turma. Fizemos assim a divisão dos dois grupos, na Tabela 2 é exibido os resultados de uma das variáveis definidas para a divisão dos grupos.

Tabela 2 - Comparativo de perfil dos participantes que responderam o formulário inicial.

Participante	Grupo A	Grupo B
1	Já cursei ensino técnico	Já cursei ensino técnico
2	É a primeira vez que estou aprendendo programação.	É a primeira vez que estou aprendendo programação.
3	Já fiz ou estou fazendo outra(s) cadeira(s) de programação	Já cursei cadeira de programação antes, mas nunca um curso completo.
4	É a primeira vez que estou aprendendo programação.	É a primeira vez que estou aprendendo programação.
5	É a primeira vez que estou aprendendo programação.	É a primeira vez que estou aprendendo programação.
6	É a primeira vez que estou aprendendo programação.	É a primeira vez que estou aprendendo programação.

Participante	Grupo A	Grupo B
7	É a primeira vez que estou aprendendo programação.	É a primeira vez que estou aprendendo programação.
8	É a primeira vez que estou aprendendo programação.	É a primeira vez que estou aprendendo programação.
9	É a primeira vez que estou aprendendo programação.	É a primeira vez que estou aprendendo programação.
10	É a primeira vez que estou aprendendo programação.	É a primeira vez que estou aprendendo programação.
11	Já fiz cursos por fora da faculdade, Estudei por conta própria	Já fiz ou estou fazendo outra(s) cadeira(s) de programação
12	É a primeira vez que estou aprendendo programação.	Estou fazendo a cadeira de FUP novamente
13	Já cursei ensino técnico, Já fiz ou estou fazendo outra(s) cadeira(s) de programação	Já cursei ensino técnico, Já fiz ou estou fazendo outra(s) cadeira(s) de programação
14	Já fiz cursos por fora da faculdade	É a primeira vez que estou aprendendo programação.

Fonte: Elaborada pelo autor, 2023.

Com a divisão dos grupos feitos, executou-se a aplicação da oficina, que dessa vez foi feita com o auxílio do professor responsável e uma terceira pessoa para auxiliar na aplicação, prezando pela otimização do tempo disponível de aula. Cada aplicação consumiu pouco menos que uma hora de aula, sendo feita primeiramente a apresentação do conteúdo e se deixou um tempo para acompanhar a execução do exercício e preenchimento do formulário de percepção.

Foi feita primeiramente a aplicação com o grupo A, que se mostrou bem participativo, e logo em seguida com o grupo B que aguardou junto com o professor no lado de fora da sala.

Nos resultados coletados pelo formulário passado após a oficina, o grupo A apresentou que 20% dos participantes já conheciam a ferramenta Python Tutor e que 40% do grupo utilizou a ferramenta durante a resolução dos exercícios.

O grupo A apresentou em suas respostas, que a utilização da ferramenta foi instrutiva e que ajudou a entender melhor o conteúdo abordado. O grupo B apesar de demonstrar interesse na dinâmica não pontuaram nenhum fator excepcional que absorveram da oficina, diferente dos comentários do grupo A. Na Tabela 3 estão sendo mostradas as respostas da percepção de aprendizado a respeito do conteúdo passado na oficina, respondendo a pergunta se “A explicação fornecida na oficina, foi suficiente para realizar o exercício? (descreva um pouco)”.

Tabela 3 - Respostas a respeito da percepção subjetiva de aprendizado após a oficina.

Participante	Grupo A	Grupo B
1	sim	Sim
2	Sim, apresentou as ferramentas necessárias para resolução.	Infelizmente não consegui prestar muita atenção pois estava com alguns problemas pessoais.
3	foi bom para refrescar a memória e aprender coisas novas ao mesmo tempo	SIM
4	sim, aprendi baseado com o que foi apresentado no python tutor	Sim, foi. A oficina rápida e direta ajudou bastante no desenvolvimento da questão, consegui compreender completamente a função do "array" e do restante, foi ótimo. Programar até que é divertido quando você entende o que está fazendo.
5	Sim, houve informação suficiente para compreender os conceitos apresentados	Sem comentário

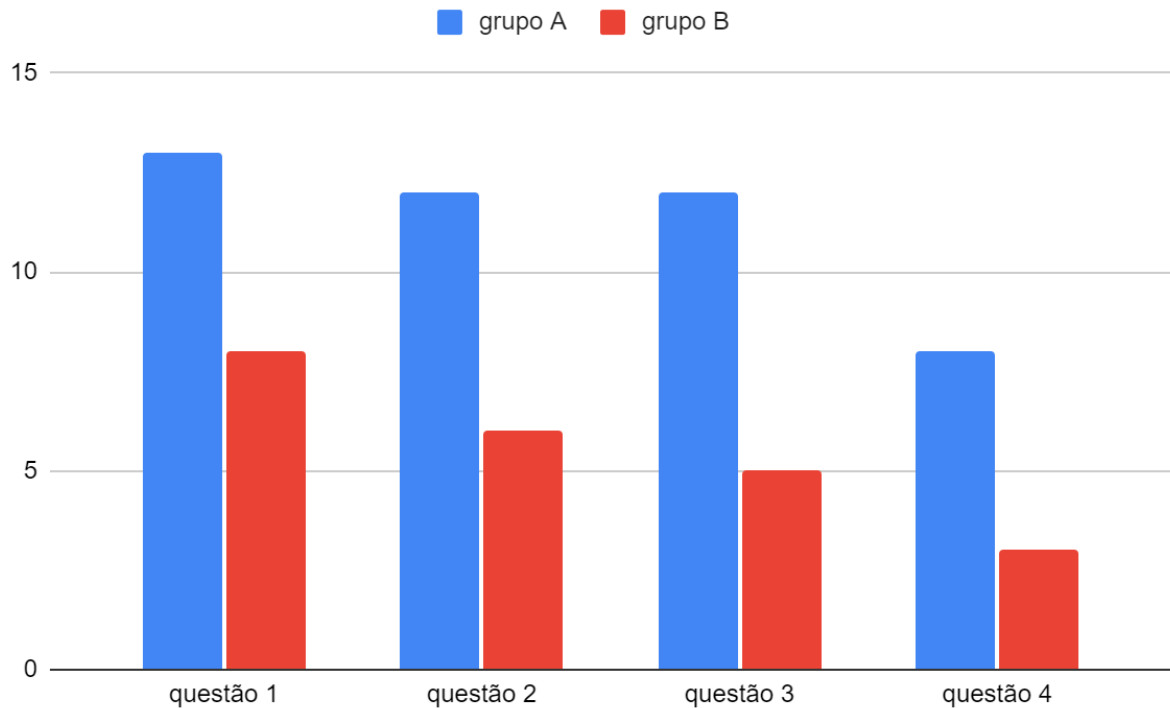
6	Foi o suficiente tendo em mente utilizações das ferramentas, facilitou muito.	Não, pois, em pequenos lapsos de desconcentração, perdi parte da conteúdo.
Participante	Grupo A	Grupo B
7	sim, foi o suficiente para entender	Sem comentário
8	Sim	Foi o suficiente, só achei que poderia ter disponibilizado o slide para consulta, caso alguém se esqueça. Oq foi o meu caso
9	as primeiras 2 sim, mas na 3º os comandos pareciam meio ambíguos	mais ou menos
10	Em partes sim, me fez entender algumas funções que tinha dificuldade	Sem comentário
11	sim	Sim, eu consegui entender.
12	sim	Sem comentário
13	Sim, ele demonstrou oque ocorreu ao longo do código, explicou a ferramenta e explicou a utilidade dos termos e tirou as dúvidas ao longo da atividade.	Sem comentário
14	Sim, foi um resumo suficiente das funções que eu precisaria usar pra responder as questões.	Sem comentário

Fonte: Elaborada pelo autor, 2023.

Na resolução dos exercícios, foi visto que nem todos os participantes resolveram todos os exercícios, na Tabela 4, podemos ver quais exercícios foram feitos por cada grupo e

participante. Os dados obtidos dos exercícios não serviram para constatar a resolução correta dos mesmos, apenas serviram de critério para observar o grau de engajamento da turma.

Gráfico 1 - Relação de conclusão dos exercícios feitos pelos dois grupos.



Fonte: Elaborada pelo autor, 2023.

No fator de percepção quantitativo, foi encontrada uma divergência de média mais positiva em relação ao grupo A, mostrando que a percepção desse grupo foi ligeiramente melhor que a percepção de aprendizagem do grupo B, dessa forma a diferença entre os valores, torna o resultado irrelevante entre os dois grupos. Na Tabela 5, é mostrado o resultado do cálculo da média de percepção de aprendizado entre os dois grupos.

Tabela 4 - Valores de média da percepção de aprendizado pelos dois grupos.

Grupo	Percepção antes da oficina	Percepção Depois da oficina
A	3.8	7.5
B	3.3	6.9

Fonte: Elaborada pelo autor, 2023.

E por fim, em relação a avaliação dos alunos do grupo A a respeito da própria ferramenta Python Tutor, na Tabela 6 é mostrado os comentários feitos após a oficina.

Tabela 5 - Comentários subjetivos do grupo A a respeito da ferramenta Python Tutor.

Participante	Comentário
1	Sem comentário.
2	“Ajudava a visualizar de forma visual o conteúdo demonstrado.”
3	“Fica melhor de compreender o que meu código vai executar e de que maneira ele vai executar”
4	“Sim, aprendi alguns conceitos que me ajudaram a resolver as questões propostas”
5	“A visualização dos dados ajudou a entender as transformações destes”
6	“Entendi melhor como era o funcionamento, melhorando assim a aplicação de algumas ferramentas.”
7	“eu conseguir entender como o python ler o código”
8	“Ajuda a visualizar”
9	Sem comentário.
10	“Ele mostra a execução parte a parte e a compreensão é melhor”
11	“ela me ajudou a entender o que acontecia em cada linha do código”
12	“uma forma melhor de aprender e ensinar”
13	“Não cheguei a utilizar a ferramenta, mas a demonstração do que acontece ao longo do código ajuda bastante.”
14	Sem comentário.

Fonte: Elaborada pelo autor, 2023.

7 DISCUSSÃO DOS RESULTADOS

Com o piloto, foi possível perceber falhas importantes na execução do experimento. Durante a etapa de coleta inicial do perfil dos estudantes, foi percebido uma falta de interesse no preenchimento do formulário, onde inicialmente apenas 5 estudantes preencheram. Além disso, no dia da aplicação da oficina, a otimização do tempo de aplicação se mostrou um fator importante, junto disso ficou claro a importância da presença do professor para incentivar a participação dos alunos.

Durante a segunda aplicação, foram corrigidos esses fatores, tornando os resultados mais consistentes, conseguindo a participação de quase todos os alunos que preencheram o primeiro formulário.

Durante a aplicação foi notado que o grupo A foi mais participativo em comparação ao grupo B, perguntando mais vezes durante a apresentação da oficina e resolvendo mais exercícios propostos. Foi possível notar por parte dos alunos do grupo A dessa forma uma boa aceitação com a metodologia, tanto nos comentários subjetivos da percepção do conteúdo aplicado com a ferramenta, quanto no indício de que o grupo A conseguiu resolver muito mais questões do exercício, principalmente a questão 4 que possuía mais complexidade que as outras.

Além da aceitação da turma com a ferramenta, e comentários positivos de que a aplicação com o Python Tutor foi mais fácil de se entender, o fator quantitativo de percepção apontou um melhor resultado no grupo A, porém o valor apresentado foi insignificante, não tendo como chegar a uma conclusão clara a respeito da percepção dos envolvidos.

Sendo assim, apesar de terem sido identificados resultados interessantes nos critérios subjetivos, não é possível chegar a resultados conclusivos, necessitando mais aprofundamento na pesquisa. Contudo é possível ver a importância de mais estudos na área, principalmente focados em critérios pedagógicos, que sustentam argumentações sobre os efeitos gerados nessas turmas.

Por fim, para critérios de aceitação da hipótese, apesar de ter sido levantados bons indicadores subjetivos sobre a aceitação da turma e da aplicação do conteúdo com a ferramenta, os critérios quantitativos não indicam uma conclusão clara a respeito da validação da H1, porém também não sustentam a H0 de forma sólida, sendo assim importante a obtenção de mais resultados, se possível com uma amostragem maior, de forma a provar de forma mais consistente os resultados da ferramenta.

8 CONCLUSÃO

A utilização de ferramentas de visualização de dados diante de metodologias de ensino de programação, é um ponto de estudo muito importante, tendo em vista o interesse e resultados apontados nesta área. Pesquisas como essa mostram o quanto ainda é possível se aprofundar nessa área, principalmente no ponto de vista pedagógico.

No cenário nacional ainda é pouco visto a presença de ferramentas como o Python Tutor, e tão notório é o desconhecimento tanto de professores quanto alunos a respeito dessa ferramenta. Muitos trabalhos se baseiam em ferramentas como o Scratch e o APP Inventor, que acabam sendo mais conhecidas atualmente.

Também é importante mencionar, a dificuldade de motivação e geração de interesse nas turmas de programação, que muitas vezes se mostram desestimuladas com os conteúdos e dinâmicas de aula. Os fatores que geram esses fenômenos são de suma importância a serem entendidos para combater esses efeitos, já que muitas vezes, turmas assim podem apresentar resultados negativos.

Dessa forma é incrivelmente rico a inserção de novas tecnologias que aguce o pensamento tanto de professores e alunos, de forma a combater tanto o desinteresse estudantil, como evasões e reprovações na área.

9 TRABALHOS FUTUROS

Para uma possível aplicação futura, seria interessante a utilização de um grupo de estudo maior, para uma validação mais clara dos critérios quantitativos. Por outro lado, uma abordagem de cunho mais pedagógico pode se aproveitar das questões subjetivas levantadas aqui, e se aprofundar com uma metodologia mais focada para tal intuito.

A aplicação com turmas que utilizem outros tipos de linguagem, como Java Script, pode ser válido, principalmente abordando o cenário de cursos como Design Digital.

Para padronização do experimento, pode ser adequados a aplicação da oficina com o auxílio de uma vídeo aula, otimizando o tempo de aplicação da explicação e diminuindo variáveis que podem influenciar o experimento, porém excluindo em parte o critério de participação da turma.

REFERÊNCIAS

- BERGIN, S.; REILLY, R. The influence of motivation and comfort-level on learning to program. In: **Proceedings of Workshop of the Psychology of Programming Interest Group**. [S.l], v. 17, p. 293–304, 2005.
- BOSSE, Yorah; GEROSA, Marco Aurélio. Reprovações e trancamentos nas disciplinas de introdução à programação da Universidade de São Paulo: um estudo preliminar. In: **Anais do XXIII Workshop sobre Educação em Computação**. SBC, 2015. p. 426-435.
- CARVALHO, Emerson; ALVES, Fábio Junior. A Eficiência do Ensino de Lógica de Programação na Modalidade a Distância. In: **Anais do XXIV Workshop de Informática na Escola**. SBC, 2018. p. 215-224.
- COSTA, Rosana Gomes; PIEDADE, João Manuel Nunes. Uso do aplicativo mit app inventor na aprendizagem de programação: uma revisão sistemática da literatura entre 2011 e 2020. **Intersaberes**. [S.l], v. 16, n. 37, 2021.
- FARIAS, Carina Machado de; OLIVEIRA, Anderson S. de; SILVA, Everton Dias de A. Uso do Scratch na Introdução de Conceitos de Lógica de Programação: relato de experiência. In: **Anais do XXVI Workshop sobre Educação em Computação**. SBC, [S.l], 2018.
- GOMES, Tancicleide CS; MELO, Jeane C. B. de. App inventor for android: uma nova possibilidade para o ensino de lógica de programação. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. [S.l], 2013.
- GREFF, Guaraci Vargas; PERES, André; BERTAGNOLLI, Silvia de Castro. Aprendizagens em movimento: um experimento de estímulo ao pensamento computacional de docentes com M-Learning e U-Learning. **Revista Thema**. [S.l], v. 15, n. 1, p. 312-322, 2018.
- GUEDES, Elloá B. Um estudo observacional sobre a disciplina introdutória de programação. In: **Anais do XX Workshop de Informática na Escola**. SBC, 2014. p. 552-561.
- GUO, Philip J. Online python tutor: embeddable web-based program visualization for cs education. In: **Proceeding of the 44th ACM technical symposium on Computer science education**. 2013. p. 579-584.
- LAZAR, Jonathan; FENG, Jinjuan Heidi; HOCHHEISER, Harry. **Research methods in human-computer interaction**. Morgan Kaufmann, 2017.
- PAPADAKIS, Stamatios et al. Novice programming environments. Scratch & app inventor: a first comparison. In: **Proceedings of the 2014 workshop on interaction design in educational environments**. 2014. p. 1-7.

PRATES, Raquel Oliveira; BARBOSA, Simone Diniz Junqueira. Avaliação de interfaces de usuário—conceitos e métodos. In: **Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação**. [S.l.], 2003. p. 28.

RAMALHO, Luciano. **Python Fluente**: Programação clara, concisa e eficaz. Novatec Editora, 2015.

RIBEIRO, Juliana Perez; MANSO, Marina Andrade; BORGES, Marcos. Dinâmicas com App Inventor no Apoio ao Aprendizado e no Ensino de Programação. In: **Anais do Workshop de Informática na Escola**. 2016. p. 271-280.

SANTOS, Rodrigo Pereira; COSTA, Heitor Augustus Xavier. Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos iniciantes em Computação e Informática. **INFOCOMP Journal of Computer Science**, v. 5, n. 1, p. 41-50, 2006.

SOUZA, Draylson Micael; BATISTA, Marisa Helena da Silva; BARBOSA, Ellen Francine. Problemas e dificuldades no ensino de programação: um mapeamento sistemático. **Revista Brasileira de Informática na Educação**, v. 24, n. 1, p. 39, 2016.

VON WANGENHEIM, Christiane Gresse; NUNES, Vinícius Rodrigues; DOS SANTOS, Giovane Daniel. Ensino de computação com scratch no ensino fundamental—um estudo de caso. **Revista Brasileira de Informática na Educação**, v. 22, n. 3, p. 115-125, 2014.