

TuPy Online: Uma Ferramenta para Visualização de Algoritmos

Jorge L. J. Goulart¹, Fabiano S. Oliveira¹, Paulo E. D. Pinto¹,
Giancarlo F. Roberto¹, Vinicius Sathler¹

¹Instituto de Matemática e Estatística
Universidade do Estado do Rio de Janeiro (UERJ)
Rio de Janeiro - RJ - Brazil

jorge_goulart@outlook.com, {fabiano.oliveira, pauloedp}@ime.uerj.br,
{giandroberto, vinicius.sathler2}@gmail.com

Abstract. *Considerable effort has been made to create tools to support the teaching of programming logic, such as algorithm visualizers. One of them is TuPy Online which, besides having features for the representation of the state of variables for each step of the execution of a program, allows visualization of complex data structures, such as graphs or trees, in a convenient high level of abstraction. Moreover, the pseudo language employs a small set of commands, all in Portuguese language, complete enough to express any algorithm. In this work, we present TuPy Online and a study on the usability of this tool. We also measure its influence in the learning process of algorithms.*

Resumo. *Um esforço considerável tem sido feito em se criar ferramentas para apoiar o ensino da lógica de programação, como os visualizadores de algoritmos. Um deles é o TuPy Online que, além de possuir recursos para a representação do estado das variáveis em cada etapa da execução de um programa, permite a visualização de estruturas de dados complexas, como grafos ou árvores, em um nível de abstração conveniente. Além disso, a pseudolinguagem emprega um pequeno conjunto de comandos, todos em língua portuguesa, completo o suficiente para expressar qualquer algoritmo. Neste trabalho, apresentamos o TuPy Online e um estudo sobre a usabilidade dessa ferramenta. Também medimos sua influência no processo de aprendizagem de algoritmos.*

1. Introdução

É notória a migração de métodos tradicionais de ensino para métodos alternativos. Com o grande avanço tecnológico e a popularidade da Internet, ambientes digitais dão oportunidade a uma nova forma de aprendizagem, tornando-se frequente o desenvolvimento de *softwares* projetados para o complemento do processo de ensino. No entanto, trabalhar com novas tecnologias no ensino não é trivial tanto quanto se poderia presumir, carecendo de preparação, habilidades básicas e materiais disponibilizados. Além disso, ao passo que alguns educadores empenham-se em ampliar o ensino e a aprendizagem com o uso de computadores, outros resistem à ideia do emprego de novas metodologias.

Em 2018, um grupo de pesquisadores da Universidade do Estado do Rio de Janeiro (UERJ), idealizou a ferramenta TuPy Online¹, como um software livre, para apoiar e auxiliar estudantes na aprendizagem de algoritmos, e com três objetivos explícitos:

¹Disponível em <https://gvirtu.github.io/TuPyOnline/>

(i) introduzir uma pseudolinguagem com um número reduzido de palavras-chave para construção de algoritmos; (ii) basear-se na língua portuguesa com comandos e identificadores escritos em português, atendendo um público-alvo sem domínio da língua inglesa; (iii) permitir ao usuário uma visualização descomplicada do progresso da estrutura de dados frente à depuração do código linha por linha.

Após o desenvolvimento da ferramenta, ela foi experimentalmente utilizada em algumas turmas. Mesmo tendo havido grande aceitação da ferramenta por parte de alunos e professores, foi planejada uma avaliação estatística sobre sua eficácia na aprendizagem, bem como uma análise sistemática de usabilidade, para uma validação mais profunda. Neste trabalho, apresentamos o *software* TuPy Online e o estudo da avaliação de aprendizagem e de usabilidade com essa ferramenta, na Universidade de origem do mesmo. Resultados parciais estão apresentados em [Roberto et al. 2018, Goulart et al. 2019].

O trabalho está organizado da seguinte forma. Na Seção 2, apresenta-se o TuPy Online. Na Seção 3, apresentam-se as metodologias utilizadas para avaliações de aprendizagem e de usabilidade. Na Seção 4, descreve-se como o experimento de avaliação foi realizado e, na Seção 5, seus resultados. Na Seção 6, conclui-se com trabalhos futuros.

2. TuPy Online

2.1. Arquitetura utilizada

A ferramenta possui quatro componentes principais: OPT, Graphviz, ANTLR4 e Interpretador para TuPy. O OPT² é uma ferramenta de apoio ao aprendizado de Python e utilizada em inúmeras universidades em todo o mundo. O Graphviz é um pacote de código aberto dedicado a representação de informação estruturada na forma de grafos e redes. Foi criado por pesquisadores da AT&T Labs em 1991 e possui apoio da comunidade até hoje. O ANTLR4 é um gerador de analisadores sintáticos a partir de definições de gramáticas. É, por exemplo, usado no mecanismo de buscas do Twitter. Finalmente, o interpretador da linguagem TuPy foi desenvolvido em Python e é responsável pelo processamento do fluxo de execução do programa. Para o melhor entendimento da arquitetura, dois dos componentes são detalhados a seguir: o OPT e o interpretador da linguagem TuPy.

OPT (Online Python Tutor)

O OPT foi criado para uso com o Python, mas, ao longo dos anos, recebeu suporte a novas linguagens. Um aspecto de sua arquitetura que favoreceu tal implementação é o fato da visualização não ser construída diretamente a partir da interpretação do programa Python, mas sim da interpretação de uma estrutura de representação intermediária, a qual é gerada a partir da execução íntegra do programa. A questão de estender o suporte a novas linguagens se dá, portanto, pela implementação de um mecanismo para geração dessa representação intermediária. Para as linguagens atualmente suportadas, existem componentes separados encarregados de executar o código com restrições de privilégios e implementar alguma estratégia para produzir a estrutura desejada.

A representação utilizada é um arquivo JSON que descreve o rastro da execução, isto é, contém uma sequência de informações acerca do estado de variáveis e da pilha de ativação para cada passo da execução do programa. Como essa execução acontece

²Disponível em <http://pythontutor.com/>

integralmente de forma não interativa, o usuário pode, em posse do arquivo retornado, navegar livremente pela visualização da execução, em passos para a frente ou para trás.

No OPT, o rastro da execução de um programa Python é obtido através da análise de dados fornecidos pelo depurador `bdb` ao executá-lo. No TuPy Online, a concepção de um interpretador permitiu a integração de um módulo que acompanha a execução dos programas e é capaz de produzir o arquivo de rastro de execução no formato desejado.

O Interpretador da Linguagem TuPy

Foi utilizada a ferramenta ANTLR4 para que, a partir de uma gramática EBNF³, houvesse geração de código para cumprir as etapas de análise léxica e sintática do interpretador, gerando uma árvore de sintaxe abstrata (AST).

Cabe ao interpretador, em seguida, percorrer a AST e registrar a evolução de estado do programa. Para isso, foram implementadas abstrações para a tabela de símbolos e a pilha de ativação, que contém as informações necessárias não somente à execução mas também à produção do rastro de execução.

2.2. Principais funcionalidades da ferramenta

A interface do TuPy Online é totalmente baseada no navegador e herda diversos aspectos do OPT. Conforme mostra a Figura 1, a página principal consiste do campo de edição de texto e dos botões de ação. A parte inferior da página também conta com um manual da linguagem TuPy e uma coleção de exemplos representativos de diferentes algoritmos e estruturas de dados. Durante a visualização, um painel de código situa-se à esquerda com destaque às linhas atual e seguinte da sequência de execução, junto aos botões de navegação. Assim como no OPT, é possível configurar pontos de parada por meio de um clique na linha de código desejada.

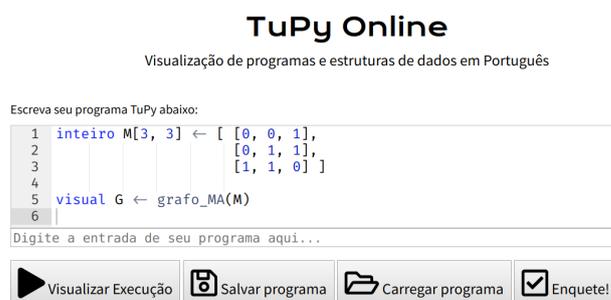


Figura 1. A interface de edição de código do TuPy Online.

Alguns pontos de destaque implementados são a possibilidade de fornecer dados de entrada do usuário a serem lidos pelo programa, assim como botões para salvar e carregar o código-fonte de um arquivo de texto. O editor de código também foi adaptado para habilitar conveniências como preenchimento automático, expansão ou contração de blocos (*code folding*) e uma fonte⁴ com suporte a ligaduras voltadas à programação, isto é, a capacidade de unir símbolos consecutivos de um código em um novo símbolo coerente automaticamente. A digitação dos símbolos correspondentes ao comparador de

³Uma extensão do Formato Backus–Naur (BNF) que, dentre outras características, possui notações que facilitam opcionalidade e repetições em regras.

⁴Fira Code. Disponível em <https://github.com/tonsky/FiraCode>. Acessado em 18 fev. 2018.

não-igualdade de operandos (\neq), por exemplo, resulta na exibição do símbolo \neq . Com isso, espera-se que o código TuPy possa se aproximar visualmente de notações de pseudocódigo manuscrito. Além disso, foi disponibilizada uma versão que pode ser executada localmente, sem necessidade de conexão com a internet, minimizando problemas de escalabilidade oriundos do compartilhamento de recursos do servidor.

2.3. Visualização de Estruturas de Dados

O projeto do TuPy Online providenciou uma biblioteca de funções para criar representações visuais extensíveis a partir de estruturas de dados.

A Figura 2 mostra as principais visualizações já prontas para diversas estruturas de dados: vetores, matrizes, listas encadeadas, pilhas, filas, árvores enraizadas, grafos e digrafos valorados ou não. Internamente, as funções encapsulam lógica para percorrer as estruturas convertendo-as para descrições textuais, que são consumidas e transformadas em imagens durante a execução no navegador do usuário pelo Graphviz, uma ferramenta para visualização de grafos e produção de diagramas estáticos descritos pela linguagem DOT.

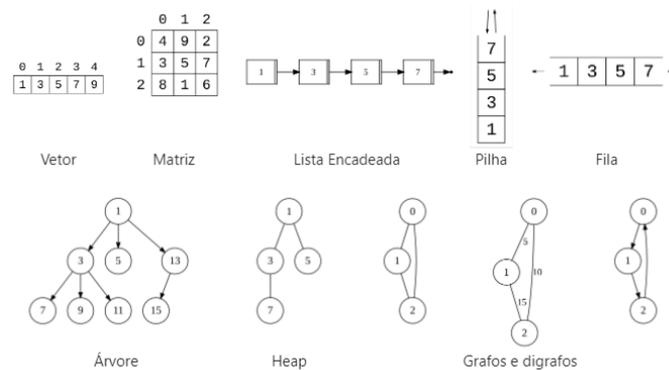


Figura 2. Exemplos de visualizações para estruturas de dados no TuPy Online

Visualizações específicas também podem ser criadas, como é o exemplo da Figura 3. Isto pode ser feito com o uso da linguagem DOT, que possui uma extensa gama

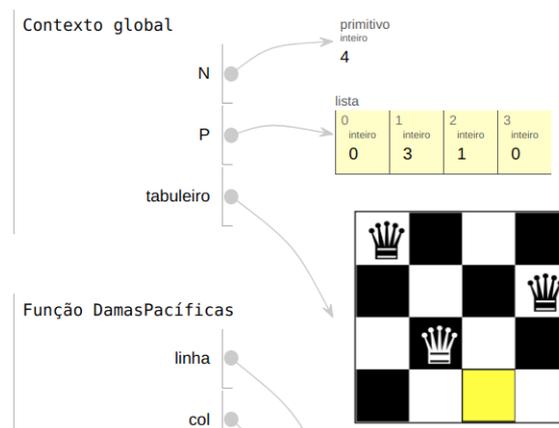


Figura 3. Tabuleiro de xadrez gerado por uma função TuPy durante a execução do algoritmo de backtracking para o problema N Rainhas ($N=4$)

de opções para a customização dos formatos, das cores e do posicionamento dos elementos visuais, o que permite gerar visualizações parametrizadas pelo estado do programa. Isso significa que, além das funções predefinidas para estruturas de dados convencionais, o programador pode criar novas funções capazes de exibir imagens com as características que desejar, complementando a visualização de forma integrada. Essa funcionalidade é observada na demonstração do problema das N rainhas em um tabuleiro de xadrez, incluído como um dos exemplos da ferramenta para ilustrar a customização da visualização para problemas específicos.

Exemplo da proximidade da visualização com a abstração da estrutura de dados

Considere o problema da obtenção de uma Árvore Geradora Mínima (AGM) em um grafo valorado. Essa é uma importante aplicação de grafos na interconexão de redes, pois uma AGM indica a forma mais barata de se obter interconexão dos diversos nós de uma rede, modelada como um grafo valorado.

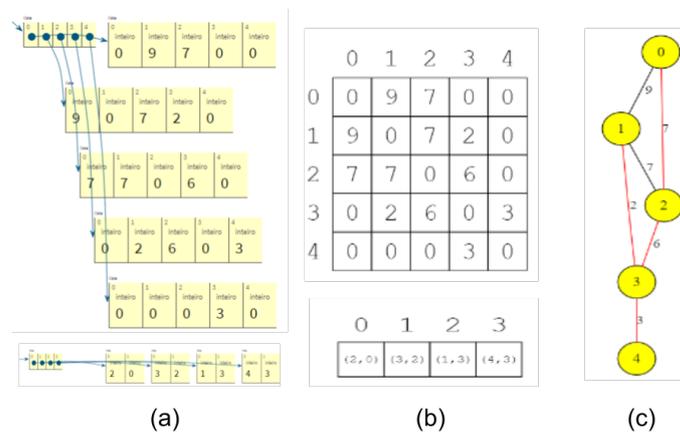


Figura 4. Representações em forma de listas (a), matrizes/vetores (b) e grafo (c)

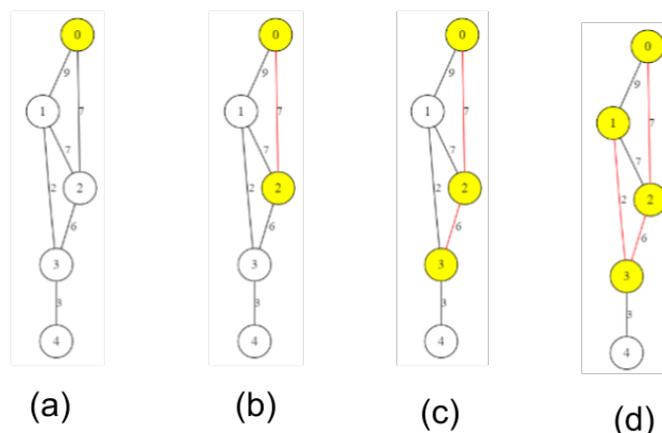


Figura 5. Visão parcial da escolha das arestas: (b): aresta (0,2); (c): aresta (2,3) e (d): aresta (1,3).

A Figura 4 ilustra três diferentes visualizações de uma AGM, obtida pelo algoritmo de Prim. Esse é um algoritmo guloso que constrói a AGM passo a passo, iniciando

com um vértice qualquer. A cada passo um novo vértice e uma nova aresta do grafo original são escolhidos e integrados à árvore em construção. A aresta escolhida deve ser aquela de custo mínimo, sujeita à restrição de que um dos vértices incidentes já esteja na árvore e o outro não. A parte esquerda da figura ilustra duas listas na memória. A da parte superior contém um conjunto de listas da representação interna da matriz de adjacências do grafo. A parte inferior mostra uma lista das arestas escolhidas: $(2, 0)$, $(3, 2)$, $(1, 3)$ e $(3, 4)$. Essas representações são aquelas feitas pelo OPT, bastante básicas. Na parte central da figura, o grafo é exibido como uma matriz de adjacências e as arestas escolhidas como um vetor. Pode-se notar que a visualização já torna mais fácil o entendimento do resultado. Finalmente, na parte da direita, a representação é em forma gráfica usual de um grafo, muito mais próxima da abstração que normalmente se deseja sobre tal estrutura de dados. Aqui, as arestas escolhidas são coloridas de vermelho, permitindo comparar diretamente a sobreposição das duas estruturas: o grafo e a AGM. Observe que os vértices estão coloridos com amarelo. A possibilidade de colorir os vértices e as arestas permite uma melhor compreensão do desenrolar do algoritmo. A Figura 5 ilustra vários passos do algoritmo onde os vértices já escolhidos para a AGM são coloridos e os demais continuam em branco.

3. Metodologias para Avaliação de Aprendizagem e Usabilidade

Em linhas gerais, a grande maioria das criações de modelos analíticos de aprendizagem empenha-se em tratar um problema específico, escorando todo o planejamento do experimento para atender as características do ambiente de aprendizagem utilizado, seja ele um jogo, *software* educacional, ou outro método computacional.

Para avaliação do TuPy Online, foi escolhida a metodologia proposta por Savi et al. que sugere uma rápida e fácil aplicação, não exigindo competências avançadas na área de educação, medição e estatística. O questionário, chamado de *MEEGA*, apoia-se nos 4 níveis de Kirkpatrick para aprendizado: Reação, Aprendizagem, Comportamento e Resultados. Savi et al. utiliza-se apenas do nível 1 e o divide em 3 partes: Motivação, Experiência e Aprendizagem. Estas 3 partes são descritas por outras metodologias de ensino: Taxonomia de Bloom, Modelo ARCS de Keller e Aprendizagens de Curto Prazo e Longo Prazo. Para consultar as referências para os 4 níveis de Kirkpatrick, Taxonomia de Bloom, Modelo ARCS de Keller e as Aprendizagens de Curto Prazo e Longo Prazo veja [Savi et al. 2011].

Além disso, optou-se por também utilizar a proposta feita por [Ferreira 2017] com o acréscimo do nível 2 de Kirkpatrick - Aprendizagem. Essa autora aplicou um pré-teste antes do uso de um *software* e um pós-teste após a utilização do mesmo. Por fim, realizou a comparação das duas avaliações para verificar se o *software* beneficiou o desempenho.

Em nenhum dos experimentos relatados foi inserido grupo de controle. Essa técnica permite o estudo experimental de um tratamento, comparando os resultados de um grupo sob o efeito do tratamento com outro grupo (grupo de controle) sem tal efeito.

Resumidamente, optou-se como metodologia para avaliação de aprendizagem uma adaptação da metodologia de Savi, a inclusão do nível 2 de Kirkpatrick e a utilização de grupos de controle. Para a análise de usabilidade da ferramenta, consideramos os seguintes aspectos: a facilidade em usar o sistema a primeira vez; rapidez para executar tarefas; o processo de lembrar como utilizar o sistema, após um tempo sem utilizar;

ausência de erros apresentados pelo sistema; e um *design* agradável. Ao contrário dos questionários de avaliação de aprendizagem, aqueles de usabilidade possuem mais referências na literatura. Entre os questionários considerados em nossa avaliação, estão os conhecidos: ASQ [Lewis 1995], PSQ [Lewis 1995], PSSUQ [Lewis 1995], CSUQ [Lewis 1995], USE [Lund 2001] e SUS [Brooke 1996].

Apesar de todos os questionários atenderem o propósito para análise da usabilidade do TuPy Online, o SUS foi escolhido devido ao número menor de questões. Além disso, outro ponto positivo é a proposta de incluir ao questionário perguntas positivas e negativas de maneira alternada, facilitando a identificação de respostas incoerentes, o que permite a remoção de tais respostas da análise final. As 10 questões do SUS são avaliadas da seguinte forma: a satisfação do usuário; a facilidade de memorização; as inconsistências; a eficiência; e a facilidade de aprendizagem.

4. Experimento

O experimento foi realizado na Universidade do Estado do Rio de Janeiro, com alunos da graduação em Ciência da Computação, da seguinte forma: uma turma de Otimização em Grafos (OTG), uma turma de Algoritmos e Estrutura de Dados I (AED1) e uma turma de Algoritmo e Estrutura de Dados II (AED2). Além disso, foi utilizado também em uma turma no curso de mestrado da disciplina obrigatória de Algoritmos (ALG).

Inicialmente os alunos estudaram durante um mês com a metodologia tradicional. Após o primeiro mês, foram informados sobre o estudo e assinaram um termo de consentimento de participação no mesmo, além de preencher um questionário de caracterização de perfil. Realizou-se após, o experimento com o TuPy Online. O experimento foi projetado da seguinte forma: os alunos realizaram o pré-teste, em seguida utilizaram o TuPy Online por um período de duas semanas e, por fim, foi aplicado o pós-teste seguido dos questionários de percepção (Nível 1 de Kirkpatrick) e usabilidade (SUS).

O grupo de controle foi adotado para avaliar o aprendizado, mantendo o foco no uso da ferramenta, e visando excluir outras variáveis tais como: modelos de provas distintas, qualidade de ensino intrínseco a cada professor, alunos com pré-conhecimento do conteúdo da disciplina. O grupo de controle foi formulado da seguinte forma: as provas do pré-teste foram divididas nos modelos *A* e *B* e cada turma dividida em dois grupos de mesmo tamanho (aleatoriamente). O grupo que fez no pré-teste a prova *A*, realizou a prova *B* no pós-teste. Já o outro, realizou o inverso, prova *B* no pré-teste seguida da prova *A* no pós-teste.

Utilizaram-se os testes *t* pareado, Wilcoxon pareado e *U* de Mann-Whitney para as análises entre pré-teste e pós-teste, além dos grupos de controle. Os resultados são apresentados com nível de confiança de 95%.

O questionário de percepção foi elaborado com 19 questões, além de uma auto-avaliação, seguindo os princípios da Taxonomia de Bloom. Por fim, o aluno respondeu o questionário *SUS* para avaliar a usabilidade do *software*. Os modelos de questionário para avaliação do TuPy Online estão publicamente disponíveis ⁵.

⁵Disponível em <https://drive.google.com/open?id=16jzj6chA3M1DXQLJvsNiDjKWEXA-aHmg>

5. Apresentação dos Resultados

Nas Figuras 6 e 7, apresentam-se os resultados obtidos com o questionário de percepção. A Figura 6 representa as dimensões Motivação e Experiência do nível 1 de Kirkpatrick. Todos os critérios obtiveram índices de concordância (Concordo plenamente ou Concordo parcialmente) entre 55% a 80%, e de discordância (Discordo plenamente ou Discordo parcialmente) abaixo dos 20%, o que indica uma percepção altamente positiva em relação à ferramenta em análise.

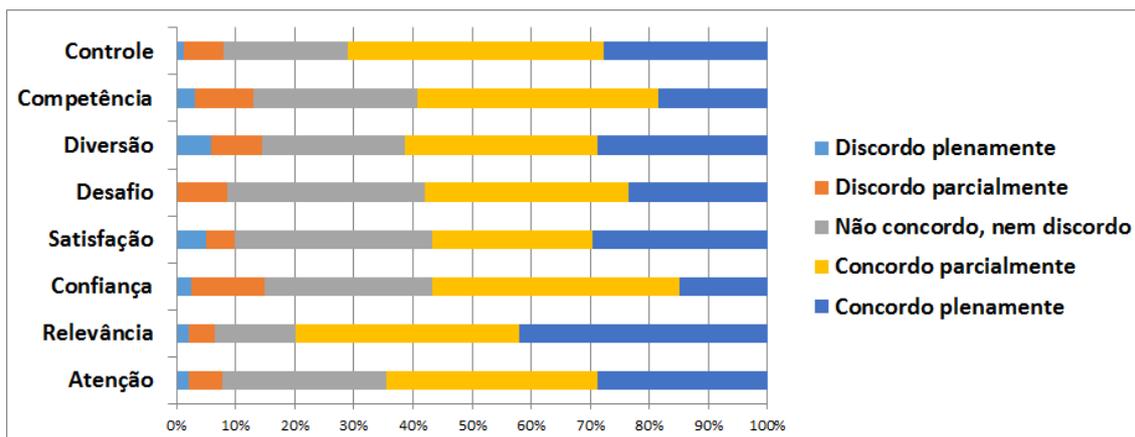


Figura 6. Avaliação do questionário de percepção.

Na dimensão Motivação, o item Relevância foi o que obteve maior avaliação. Em outras palavras, dentre os 4 componentes (Satisfação, Confiança, Relevância e Atenção), os alunos estão atribuindo grande importância aos conteúdos oferecidos pelo TuPy Online. Sobre a Experiência do usuário, foram avaliados 4 itens (Desafio, Diversão, Competência e Controle). A dimensão Controle foi a melhor avaliada, ou seja, dada a experiência do aluno com o TuPy Online, a facilidade para usar os controles e interfaces do *software*, bem como as boas respostas para realizar ações obtiveram destaque perante aos usuários. Em uma análise geral, todos os itens do questionário de percepção apresentaram índice com mais de 50% de concordância.

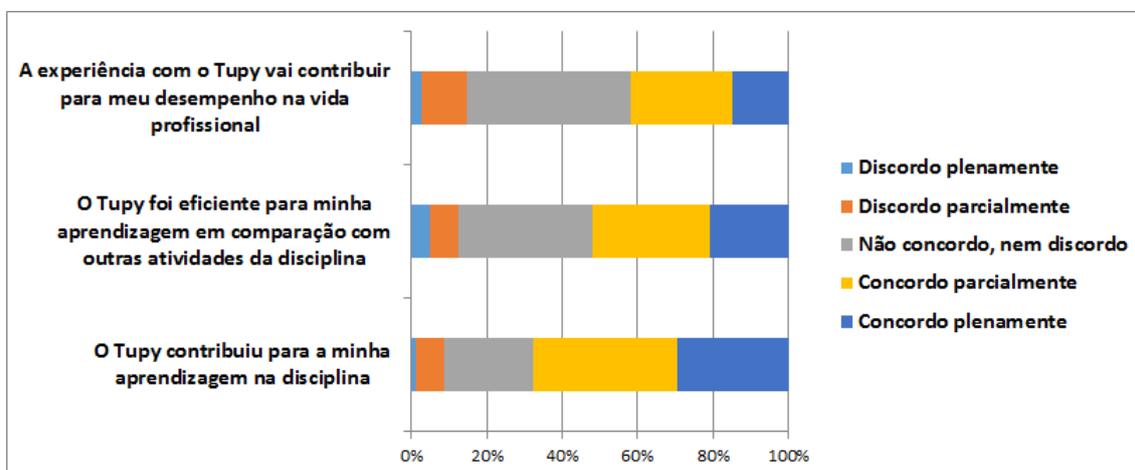


Figura 7. Aprendizagem de curto e longo prazo.

AED1	Lembrança		Compreensão		Aplicabilidade	
	Antes	Depois	Antes	Depois	Antes	Depois
Listas Lineares	2,61	3,00	2,55	3,38	2,11	2,61
Pilhas	3,00	3,44	2,50	3,22	2,27	2,72
Filas	3,06	3,44	2,77	3,38	2,50	2,83
OTG	Lembrança		Compreensão		Aplicabilidade	
	Antes	Depois	Antes	Depois	Antes	Depois
Busca em Grafos	2,66	3,12	2,66	3,20	2,37	3,12
Ordenação Topológica/COM	2,66	3,04	2,58	3,12	2,37	3,00
Componentes Fortemente Conexas	2,37	3,04	2,37	3,08	2,33	2,95
AED2	Lembrança		Compreensão		Aplicabilidade	
	Antes	Depois	Antes	Depois	Antes	Depois
Divisão e Conquista	3,00	3,55	2,88	3,77	2,66	3,33
Recursão e Memorização	3,33	3,77	3,55	3,77	3,11	3,44
Programação Dinâmica	3,44	3,77	3,55	3,77	3,11	3,44
ALG	Lembrança		Compreensão		Aplicabilidade	
	Antes	Depois	Antes	Depois	Antes	Depois
Listas Lineares	2,36	2,86	2,56	3,13	2,10	2,66
Pilhas	2,53	3,06	2,60	3,26	2,33	2,86
Filas	2,20	2,73	2,33	2,83	2,20	2,63

Tabela 1. Resultados da avaliação pela Taxonomia de Bloom.

Quando questionados sobre a influência do TuPy Online na aprendizagem (Figura 7), verificou-se que os alunos acreditam que o *software* tem grande importância e influência para o aprendizado na disciplina. No entanto, apesar da boa avaliação, não o consideram como uma ferramenta que contribua para o desempenho profissional. Tais percepções coincidem com os objetivos do TuPy Online, em ser uma ferramenta educacional no ensino de programação.

A Tabela 1 apresenta os resultados da auto-avaliação dos alunos. Verificou-se que para todos os assuntos abordados nas disciplinas, os alunos se auto-avaliaram melhores ao lembrar, compreender e aplicar a metodologia estudada após a utilização do TuPy Online.

Na avaliação de aprendizagem pelas notas obtidas, a hipótese nula adotada é a não influência do TuPy Online. Desta forma, tem-se os resultados a seguir.

Considerando todos os alunos de todas as disciplinas, em que 85 alunos participaram, a análise indicou a rejeição da hipótese nula para um nível de confiança de 95%, para o p-valor de 0,0101. Em outras palavras, houve uma melhora significativa no aprendizado dos alunos com a utilização do TuPy Online. Já analisando turma por turma, obteve-se os seguintes resultados. Na disciplina AED1, nenhum dos 2 Grupos rejeitou a hipótese nula. Ou seja, não foram encontradas evidências estatísticas para comprovar que o TuPy Online influenciou no desempenho dos alunos. Em OTG e AED2, apenas 1 Grupo rejeitou a hipótese nula, indicando melhora no desempenho nas provas. Já na disciplina de ALG, o TuPy demonstrou diferença apenas no primeiro grupo, rejeitando a hipótese nula. No segundo Grupo, rejeita-se a hipótese nula, contudo, se desconsiderarmos os alunos que reportaram não terem estudado em casa com o TuPy Online.

Em relação aos grupos de controle, criados para estudar o efeito das provas no desempenho dos alunos, o resultado foi inconclusivo em mostrar que elas não tiveram influência. Sendo assim, há a possibilidade de que a melhoria observada seja parcialmente devida a provas com níveis de dificuldades distintas.

Um primeiro resultado da análise sobre usabilidade, considerando as respostas de maneira global, foi o percentil de respostas positivas encontrado, de 66%. Comparando

com a análise feita por [Sauro 2011], o resultado é bem próximo à média de 68%, relativa a cerca 500 análises. Essa média está homogeneamente distribuída entre os itens analisados, sendo o de pior avaliação a satisfação, com 60% do total de pontos possíveis e o melhor avaliado a eficiência, com 72%. Esses resultados sugerem que os usuários consideram as funcionalidades do *software* bem integradas, de execução rápida, mas ainda necessitam de ajuda técnica para o uso da ferramenta. Foram dadas inúmeras sugestões de melhorias, que serão consideradas em futuras versões.

6. Conclusão

Neste trabalho, apresentamos a ferramenta TuPy Online, criada com o objetivo principal de suportar a visualização das estruturas de dados em alto nível de abstração durante a execução de algoritmos. Apresentamos também avaliações de usabilidade da ferramenta e da sua eficiência na aprendizagem de algoritmos. Quanto à avaliação de aprendizagem, o questionário de percepção demonstrou alto índice de concordância dos entrevistados em relação tanto à relevância da proposta da ferramenta, quanto à praticidade de sua interface. A avaliação sistemática é conclusiva em relação à influência positiva da ferramenta na aprendizagem. A auto-avaliação demonstrou a satisfação por parte dos alunos de todas as turmas. Em relação à usabilidade, em uma primeira avaliação, o resultado foi positivo identificando possíveis melhorias e aperfeiçoamento do TuPy Online. Como trabalho futuro, além de melhorias identificadas pela avaliação de usabilidade, pretendemos integrar outras ferramentas, tal como a visualização de gráficos, e também internacionalizá-la, pelo reconhecimento de pseudocódigos em outros idiomas.

Referências

- Brooke, J. (1996). SUS- A quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7.
- Ferreira, A. (2017). MA-AVA: Modelo de Avaliação da Aprendizagem em Ambientes Virtuais. *Dissertação de Mestrado. Universidade do Estado do Rio de Janeiro*.
- Goulart, J., Oliveira, F., and Pinto, P. (2019). Avaliação Sistemática de Eficácia na Aprendizagem de Algoritmos com o uso do Tupy Online. *27º Workshop sobre Educação em Computação (WEI)*.
- Lewis, J. R. (1995). IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*, 7(1):57–78.
- Lund, A. M. (2001). Measuring usability with the use questionnaire12. *Usability interface*, 8(2):3–6.
- Roberto, G., Oliveira, F., Pinto, P., and Coelho, I. (2018). Tupy Online - Programação em Português com Visualização de Execução e Abstrações de Estruturas de Dados na Web. *26º Workshop sobre Educação em Computação (WEI)*.
- Sauro, J. (2011). Measuring usability with the system usability scale (sus). 2011. URL: <http://www.measuringusability.com/sus.php> [accessed 2019-09-25].
- Savi, R., Wangenheim, C., and Borgatto, A. (2011). Um Modelo de Avaliação de Jogos Educacionais na Engenharia de Software. *25º Simpósio Brasileiro em Engenharia de Software (SBES)*.