

Python e JavaScript como primeiras linguagens no ensino de programação

Python and JavaScript as first languages in teaching programming

RESUMO

Esta pesquisa objetiva realizar um levantamento sobre vantagens, desvantagens e considerações do uso de Python e JavaScript como linguagens introdutórias no processo de ensino-aprendizagem dos fundamentos de programação, considerado o contexto dos cursos do Câmpus Guarapuava da UTFPR. Conforme análise feita, as disciplinas introdutórias de programação do câmpus usam Java e C como linguagens e apresentam uma taxa de reprovação de 57,52% (dados de 2011 a 2019). Diante disso, procedeu-se com o estudo dos elementos básicos das linguagens Python e JavaScript, bem como consequente análise de vantagens e desvantagens para o ensino introdutório de programação. Foram desenvolvidos minicursos dessas linguagens para avaliação prática com alunos. A oferta dos minicursos (mesmo on-line) foi prejudicada pelo período de pandemia, entretanto tentativas subsequentes serão feitas. Os resultados dessa pesquisa foram úteis para o processo de atualização da matriz curricular do curso de Tecnologia em Sistemas para Internet, na escolha de JavaScript como primeira linguagem de programação. Para os cursos de engenharia, considerou-se Python como alternativa mais adequada.

PALAVRAS-CHAVE: Linguagem de Programação. Ensino Superior. Computação. Engenharia.

ABSTRACT

This research aims to survey the advantages, disadvantages, and considerations of using Python and JavaScript as introductory languages in the teaching-learning process of the fundamentals of programming, considered the context of Guarapuava campus courses (UTFPR). According to the analysis, introductory programming subjects use Java and C as languages and have a failure rate of 57.52% (data from 2011 to 2019). Therefore, we proceeded with the study Python and JavaScript basic elements, as well as the consequent analysis of advantages and disadvantages for the introductory teaching of programming. Mini-courses of these languages were developed for practical evaluation with students. The offering of short courses (even online) was harmed by the pandemic period, however subsequent attempts will be made. The results of this research were useful for the process of updating the curricular matrix of the Internet Systems Technology course, in choosing JavaScript as the first programming language. For engineering courses, Python was considered the most suitable alternative.

KEYWORDS: Programming Language. Higher Education. Computing. Engineering.

Alan Michael de Oliveira
alanoliveiramichael@gmail.com
Universidade Tecnológica Federal do Paraná, Guarapuava, Paraná, Brasil

Eleandro Maschio
eleandrom@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Guarapuava, Paraná, Brasil

Recebido: 04 set. 2020.

Aprovado: 01 out. 2020.

Direito autorial: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



INTRODUÇÃO

Independentemente do curso escolhido, programar é uma atividade indispensável no ensino superior em Ciência da Computação (MASCHIO, 2015). Além disso, é uma atividade importante e presente nos cursos de Engenharia. Portanto, potencializar o ensino de programação mostra-se interessante no contexto institucional da UTFPR.

No cenário do Câmpus Guarapuava (UTFPR-GP), três dos quatro cursos existentes ofertam disciplinas de programação: Tecnologia em Sistemas para Internet (TSI), Engenharia Mecânica (EM) e Engenharia Civil (EC). Em levantamento realizado, considerando as ofertas de disciplinas introdutórias de programação, desde a fundação do câmpus em 2011, tem-se uma **taxa de reprovação de 57,52%** (que já chegou a 63,46% em 2012). Atualmente, o curso de TSI tem Java como primeira linguagem, enquanto as engenharias usam C.

Alguns autores da área defendem a escolha da primeira linguagem de programação como um dos fatores determinantes na motivação do aluno. Em 2017, a Universidade de Stanford, após utilizar Java por 15 anos como primeira linguagem, optou por trocá-la por JavaScript, pela simplicidade. Tal mudança teve como objetivo introduzir os fundamentos de programação por meio de uma linguagem menos complexa para os iniciantes (ROBERTS, 2019; CLABURN, 2017).

Nessa mesma perspectiva, a linguagem Python também tem se tornado uma boa opção didática (DOWNEY, 2016; RADENSKI, 2006). Além disso, Python vem crescendo progressivamente entre a comunidade de programadores nos últimos anos, como observado no *rankings* recentes da TIOBE (2019). A principal característica reside na sintaxe ser relativamente simples, assemelhando-se muito ao pseudocódigo. Esse diferencial da linguagem mostra-se ainda mais marcante no ensino de introdução da lógica de programação.

Entretanto, a problemática pode ser apresentada em termos práticos, apenas por meio da codificação de um mesmo exemplo nas quatro linguagens. Os códigos-fonte 1 e 2 mostram a implementação do exemplo *Hello World* (Olá, Mundo), respectivamente, em Java e C.

Código-fonte 1 – *Hello World* em Java

```
public class Main
{
    public static void main(String[] args)
    {
        System.out.println("Hello World");
    }
}
```

Fonte: Autoria própria (2020).

Código-fonte 2 – *Hello World* em C

```
#include<stdio.h>

main()
{
    printf("Hello World\n");
}
```

Fonte: Autoria própria (2020).

Pode-se observar que o simples *Hello World*, em Java, já faz uso de conceitos como: classes, métodos (*main*), modificadores de acesso (*static*), tipos de retorno (*void*), passagem de parâmetros (*String[] args*), matriz unidimensional (para passagem dos argumentos em *String[]*). Na linguagem C, são usados conceitos como importação de bibliotecas (*#include*), funções (*main*) e quebra de linha (*\n*).

Em contrapartida, nas linguagens JavaScript e Python, a implementação consiste em uma única linha, responsável por exibir a mensagem na tela. Isso pode ser observado nos códigos-fonte 3 e 4.

Código-fonte 3 – *Hello World* em JavaScript

```
console.log("Hello World");
```

Fonte: Autoria própria (2020).

Código-fonte 4 – *Hello World* em Python

```
print("Hello World")
```

Fonte: Autoria própria (2020).

Trata-se de uma simplificação grosseira afirmar que substituir a linguagem resolve o problema das reprovações nas disciplinas citadas. Contudo, diante dos exemplos, é evidente que a linguagem pode trazer uma complexidade desnecessária para o primeiro contato com a programação.

Nesse sentido, a pesquisa teve como objetivo realizar um levantamento sobre vantagens, desvantagens e considerações do uso de Python e Javascript como linguagens introdutórias no processo de ensino-aprendizagem dos fundamentos de programação, considerado o contexto dos cursos da UTFPR-GP.

MATERIAL E MÉTODOS

Revisão bibliográfica: Foram buscados, na literatura, artigos científicos que tratassem do uso de Python e JavaScript como linguagens introdutórias nas disciplinas de programação. Essa busca enfocou artigos que trouxessem vantagens e desvantagens, além do relato de experiência sobre a possível utilização dessas linguagens no contexto da presente pesquisa.

Levantamento realizado no Câmpus Guarapuava: Foi realizado um levantamento, junto ao Departamento de Registros Acadêmicos (DERAC) sobre as reprovações, nas disciplinas introdutórias de programação dos cursos de Tecnologia em Sistemas para Internet (TSI), Engenharia Mecânica (EM) e Engenharia Civil (EC) no Câmpus Guarapuava. O curso de Tecnologia em Manutenção Industrial foi desconsiderado porque não possui disciplina introdutória de programação na matriz curricular. O levantamento abrangeu os dados desde a fundação do câmpus, no primeiro semestre de 2011, até o segundo semestre de 2019 (último semestre encerrado no momento).

Estudo sobre as linguagens Python e JavaScript: Ambas as linguagens tiveram seus elementos básicos estudados, partindo dos fundamentos de programação do pesquisador na linguagem Java (aprendida na disciplina introdutória de programação no curso de TSI). Foram consultados materiais didáticos (livros,

apostilas e videoaulas) com a intenção de conhecer as peculiaridades de Python e JavaScript. Como comparativo, um mesmo conjunto de 77 problemas, referentes ao conteúdo introdutório de programação, foi resolvido nas 3 linguagens.

Análise de vantagens e desvantagens para o ensino introdutório de programação: Diante da experiência, foram levantadas vantagens, desvantagens e características de Python e JavaScript em comparação com Java e C (linguagens atualmente ensinadas no câmpus). Foram considerados não apenas aspectos sintáticos, como também, por exemplo: o uso no contexto dos cursos do câmpus, no exercício profissional do egresso e a aceitação comercial; adoção da linguagem em juízes on-line; e qualidade de material didático on-line e gratuito.

Avaliação prática de Python e JavaScript com alunos: Pretendia-se realizar a avaliação prática de, pelo menos, uma das linguagens com alunos do Câmpus Guarapuava. Para isso, foram idealizados minicursos, que seriam ministrados presencialmente. Devido ao período de pandemia mundial (COVID-19), infelizmente, a oferta presencial dos minicursos foi impossibilitada. Houve tentativa de oferta on-line na qual não existiram interessados que correspondessem aos requisitos. Esperava-se utilizar uma abordagem quali-quantitativa, com a aplicação de questionários (perguntas abertas e fechadas) e também comparando número de submissões e de acertos dos exercícios em cada linguagem (propostos em um juiz on-line).

RESULTADOS E DISCUSSÃO

LEVANTAMENTO REALIZADO NO CÂMPUS GUARAPUAVA

O Quadro 1 reúne todas as disciplinas introdutórias de programação já oferecidas no câmpus e as respectivas linguagens utilizadas, bem como o período que cada uma delas foi ofertada. Os semestres são indicados como 1 ou 2, sucedendo o ano após a barra (2011/1 indica o primeiro semestre de 2011).

Quadro 1 – Disciplinas introdutórias de programação oferecidas no Câmpus Guarapuava

Curso	Disciplina	Linguagem	Início	Fim
Tecnologia em Sistemas para Internet	Algoritmos	Portugol	2011/1	2015/2
	Linguagem de Programação	C	2011/1	2015/2
	Pensamento Computacional e Fundamentos de Programação	Java	2016/1	2019/2
	Introdução à Orientação a Objetos	Java	2016/1	2019/2
Engenharia Mecânica	Computação 1	C	2011/2	2019/2
Engenharia Civil	Fundamentos de Programação	C	2014/1	2019/2

Fonte: Autoria própria (2020).

A Tabela 1 resume a relação de alunos matriculados e aprovados, por ano, nas disciplinas introdutórias de programação do câmpus. Convém destacar que, em 2011/1 havia apenas disciplinas de TSI, uma vez que se oferece Computação 1 apenas no 2º Período de EM. Além disso, Fundamentos de Programação, disciplina pertencente ao 2º Período de EC, teve a primeira oferta em 2014/2.

Em suma, desde a fundação do câmpus, ocorreram **2.818 matrículas** em disciplinas introdutórias de programação. Dentro disso, foram **1.197 aprovações** e **1.621 reprovações**. Portanto, tem-se uma **taxa de aprovação de 42,48%** e uma

taxa de reprovação de 57,52%. A maior taxa de aprovação foi de 53,79% em 2015 e a menor taxa foi de 36,54% em 2012.

Tabela 1 – Relação de alunos matriculados e aprovados em disciplinas introdutórias de programação por ano

Ano	Matriculados	Aprovados	Percentual(%)
2011	140	54	38,57%
2012	208	76	36,54%
2013	250	116	46,40%
2014	331	149	45,02%
2015	290	156	53,79%
2016	402	156	38,81%
2017	400	167	41,75%
2018	419	176	42,00%
2019	378	147	38,89%
TOTAL	2.818	1.197	42,48%

Fonte: Autoria própria (2020).

COMPARATIVO ENTRE AS LINGUAGENS CONSIDERADAS

Diante da revisão bibliográfica e do estudo das linguagens Python e JavaScript, foi estabelecido um comparativo entre as linguagens Python, JavaScript, Java e C. Além das diferenças sintáticas, vários outros aspectos foram contemplados, como: vantagens, desvantagens, considerações relacionadas aos cursos aos cursos, disponibilidade de materiais, adoção por juízes on-line, aceitação no mercado, entre outros. O comparativo está disponível em: cutt.ly/uf3Um9A.

ESTRUTURAÇÃO E PRODUÇÃO DE MATERIAL DIDÁTICO PARA MINICURSOS DE PYTHON E JAVASCRIPT

Com a intenção de avaliar, pelo menos, uma das linguagens, entre Python e JavaScript, na prática com aprendizes, foram estruturados dois minicursos, sendo um para cada linguagem. Basicamente, ambos os minicursos possuem o mesmo conteúdo e variam apenas em questões específicas das linguagens. Cada minicurso consiste em 6 aulas e enfoca aspectos introdutórios de programação na linguagem, por meio de exemplos e exercícios. Como exemplos e exercícios, utilizaram-se enunciados da plataforma URI Online Judge, que provê correção automática, por meio de um juiz on-line, mediante a submissão da resposta pelo aluno. As aulas obedecem a seguinte estrutura:

- Aula 1 – Configuração do ambiente para as aulas: Apresentação da proposta do minicurso e respectivo funcionamento; Configuração do ambiente computacional para as aulas; e Cadastro na Plataforma URI Online Judge;
- Aula 2 – Introdução: Conceitos básicos de programação (algoritmo, programa, linguagem de programação, linguagens de alto e de baixo

nível); Sintaxe básica da linguagem e palavras-reservadas; Instruções de entrada, de saída e de atribuição; Variáveis, tipos de dados (inclusive *string*) e conversão entre tipos de dados; Operadores e expressões aritméticas; e Comentários;

- Aula 3 – Estruturas Condicionais: Operadores relacionais e relacionais; Expressões relacionais, booleanas e compostas; Estruturas condicionais da linguagem; e Composições envolvendo estruturas condicionais;
- Aula 4 – Estruturas de Repetição: Estruturas de repetição condicionais; Estrutura de repetição com variável de controle (laços contados e iteração em listas); Composições envolvendo estruturas condicionais e de repetição; e Instrução *break*;
- Aula 5 – Listas: Conceito de listas como estrutura de dados unidimensional, índices e tamanho; Declaração e acesso a elementos isolados; Inserção, remoção, alteração e busca; Percorrimento com estruturas de repetição; e Entendimento de *strings* como estruturas de dados e métodos específicos;
- Aula 6 – Matrizes Bidimensionais: Conceito de estrutura de dados bidimensional (implementada como listas de listas), índices, linhas, colunas e tamanho; Declaração e acesso a elementos isolados; Inserção, remoção, alteração e busca; e Percorrimento com estruturas de repetição.

OFERTA DE MINICURSOS

Cada minicurso é composto por 6 encontros com duração de 90 minutos. As aulas presenciais totalizaram 9 horas. Os encontros poderiam ser compreendidos em 3 semanas (sendo 2 encontros por semana) ou 6 semanas (no caso de 1 encontro por semana). A carga-horária total (máxima) estimada foi de 20 horas.

Devido ao período de pandemia (COVID-19), a oferta presencial dos minicursos foi impossibilitada. Como alternativa, proposta dos minicursos foi reorientada para o formato remoto. O minicurso on-line foi ofertado aos alunos do câmpus. Como requisito, foi pedido que os participantes não tivessem experiência prévia com programação e que possuísem disponibilidade de tempo necessária. Os alunos estavam com aulas suspensas na época. Esperava-se um mínimo de 5 e um máximo de 15 alunos na oferta. Mesmo com a extensão dos prazos para inscrição, houve apenas 4 inscritos e nenhum atendia os requisitos.

VALIDAÇÃO DO MATERIAL DO MINICURSO

Apesar de não ter ocorrido a avaliação prática com aprendizes, o minicurso de Python teve oferta validada com um voluntário. A versão era composta por 5 encontros com duração de 90 minutos, totalizando 7 horas e 30 minutos de aulas remotas. Os encontros correspondiam às aulas de 2 a 6, conforme supracitado. O participante não tinha conhecimento prévio de programação. Como formação, era graduado em Administração e doutorando no Programa de Pós-Graduação em Tecnologia e Sociedade (PPGTE) do Câmpus Curitiba.

Na oferta, o cronograma do minicurso precisou ser estendido de 2,5 para 4 semanas, diante da justificativa de questões pessoais do participante (falecimento na família). Apenas alterações pontuais foram indicadas, e realizadas, para aperfeiçoamento do material. A principal necessidade foi a inclusão de uma primeira aula para configuração do ambiente computacional.

AUXÍLIO NO PROCESSO DE ATUALIZAÇÃO DA MATRIZ CURRICULAR DE TSI

Atualmente, a matriz curricular do curso de TSI passa por um processo de atualização. Houve duas matrizes anteriores, sendo a primeira praticada desde a implantação, em 2011/1 até 2015/2 (dez semestres) e a outra desde 2016/1 até o presente (nove semestres).

A proposta da próxima matriz curricular do curso baseia-se em competências. Os estudos realizados pela pesquisa foram levados às reuniões do Núcleo Docente Estruturante (NDE) de TSI, para que auxiliasse na escolha da linguagem introdutória de programação no curso. O corpo docente escolheu por JavaScript, considerando as particularidades e os seguintes aspectos da linguagem: **(1)** Sintaticamente simples, com instruções de controle não muito diferentes de Java, C e PHP; **(2)** Entre as linguagens consideradas, mostra-se mais aderente ao perfil do egresso (cenário de oportunidades nacionais e mundiais), pois dificilmente pode ser evitada na programação web; **(3)** Não se pode, no atual contexto, omiti-la da matriz curricular do curso; **(4)** Antecipa a formação para as oportunidades de estágio em programação web; e **(5)** Crescente quantidade de materiais tendo JavaScript como primeira linguagem.

SUGESTÃO DE PYTHON COMO PRIMEIRA LINGUAGEM PARA AS ENGENHARIAS

No contexto das engenharias, Python destaca-se por enfatizar a linguagem como ferramenta diferencial no exercício da profissão (ou mesmo na trajetória acadêmica). Existem várias bibliotecas com funcionalidades úteis para engenheiros, tais como: NumPy (matrizes e funções matemáticas complexas), SciPy (matemática e engenharia), Matplotlib e Seaborn (visualização de dados), além de Plotly (visualização interativa de dados), para citar algumas.

Além disso, a sintaxe básica da linguagem é tão simples quanto o próprio pseudocódigo. Como referência de estudos, há o livro *Pense Python*, oferecido sob licença Creative Commons (Downey, 2016). Ainda, a plataforma URI Online Judge permite que os exercícios iniciais sejam resolvidos por programação em blocos, com conversão automática para o código Python. Trata-se de um amparo útil para os primeiros passos daqueles alunos que apresentam dificuldades.

A linguagem C continua se mostrando pertinente às engenharias, por conta da programação de Arduino, essencialmente feita na linguagem. Entretanto, deve-se considerar que a proporção de alunos que nunca programarão para Arduino (e que terão a linguagem C como empecilho) predomina no curso de engenharia. Nada impede, também, que aqueles alunos interessados em programação para Arduino aprendam a linguagem C, partindo dos fundamentos de programação em outra linguagem mais didática.

CONCLUSÃO E TRABALHOS FUTUROS

A pesquisa buscou considerar Python e JavaScript como primeiras linguagens nas disciplinas introdutórias de programação do Câmpus Guarapuava. Diante do levantamento realizado, tem-se um histórico de reprovação de 57,52%. Obviamente, a mera substituição da linguagem não deve resolver o problema das reprovações por si. Entretanto, observa-se que as linguagens Java e C desfavorecem o primeiro contato com a programação.

Os resultados dessa pesquisa auxiliaram na escolha da primeira linguagem de programação, para a nova matriz curricular do curso de TSI. A contribuição se estende às engenharias, no sentido de recomendar Python como alternativa mais adequada. Citam-se como trabalhos futuros imediatos: **(1)** Avaliação das linguagens Python e JavaScript na prática com aprendizes em duas ofertas de cada minicurso; **(2)** Disponibilização na Internet do material didático produzido, para que a comunidade possa ter acesso tanto às videoaulas quanto aos demais conteúdos dos minicursos; e **(3)** Disseminação dos resultados completos da pesquisa em publicações da área, tais como: Journal on Computational Thinking (JCThink), Simpósio Brasileiro de Educação em Computação (EduComp), Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WAlgProg) e Workshop de Educação em Computação (WEI).

AGRADECIMENTOS

Agradecemos aos servidores do Departamento de Registros Acadêmicos (DERAC) do Câmpus Guarapuava por auxiliarem com o levantamento realizado e à UTFPR pela bolsa de Iniciação Científica (Edital 02/2019 – PROPPG).

REFERÊNCIAS

CLABURN, Thomas. Stanford **Uni's intro to CompSci course adopts JavaScript, bins Java: Java's days are numbered – but it's a very large number.** 2017. The Register. Disponível em: https://www.theregister.com/2017/04/24/stanford_tests_javascript_in_place_of_java/. Acesso em: 20 ago. 2020.

DOWNEY, Allen B.. **Pense em Python: pense como um cientista da Computação.** 312: Novatec, 2016. 312 p. (Ampliado para Python 3).

MASCHIO, Eleandro; DIRENE, Alexandre. Múltiplas Representações Externas no Suporte à Aquisição de Conhecimento em Programação de Computadores. **Revista Brasileira de Informática na Educação**, v. 23, p. 81-96, 2015

RADENSKI, Atanas. "Python First": a lab-base digital introduction to computer science. **Proceedings 11th annual SIGCSE conference on Innovation and technology in computer science education.** New York: Association For Computing Machinery, 2006. p. 197-201. Disponível em: <https://dl.acm.org/doi/10.1145/1140124.1140177>. Acesso em: 12 jan. 2020.

ROBERTS, Eric. **Introduction to JavaScript Programming**: the “nothing but a browser” approach. New York: Pearson, 2019. 480 p.

TIOBE Software BV. **TIOBE Index for June 2019**: Python continues to soar in the TIOBE Index. Disponível em: <https://www.tiobe.com/tiobe-index>. Acesso em: 2 jul. 2019.