

TABELAS DE AMORTIZAÇÃO DE FINANCIAMENTOS EM PYTHON

DELGADO, D. M.

*Fatec Mococa - Coordenadoria de Gestão Empresarial
darlan.delgado@fatec.sp.gov.br*

Loan Amortization Tables in Python

Eixo Tecnológico: Gestão e Negócios.

Resumo

O artigo aborda a elaboração de um código em linguagem de programação Python para a geração de tabelas do Sistema Francês de Amortização (SFA). O objetivo é apresentar o resultado parcial de pesquisa sobre as aplicações de Python no ensino de graduação tecnológica na Educação Profissional, na área de Economia e Finanças. O código gera como saída o resumo da operação financeira do financiamento, a sua respectiva tabela de amortização e a criação do arquivo de planilha eletrônica Microsoft Excel. O estudo foi desenhado metodologicamente em função de sólida bibliografia em Matemática Financeira para a elaboração das tabelas de amortização de empréstimos. De modo análogo, a elaboração do código foi baseada em pesquisa bibliográfica e documental sobre a linguagem de programação Python e das bibliotecas necessárias. Há resultados computacionais e educacionais da pesquisa. Na dimensão computacional, obteve-se um código apto a gerar tabelas financeiras e os respectivos resumos das operações de crédito, bem como criar o arquivo no formato Microsoft Excel. Na dimensão educacional, verifica-se que as competências a serem desenvolvidas nos alunos da Educação Profissional são principalmente as inerentes aos conhecimentos e à parte técnica.

Palavras-chave: *Matemática Financeira, Sistema de Amortização Francês, Python, Ensino de graduação, Educação Profissional.*

Abstract

The article discusses the development of a code in the Python programming language for generating tables of the French System (FS). The aim is to present the partial result of a research into the applications of Python in undergraduate technological teaching in Vocational Education, in the field of Economics and Finance. The code generates as output the summary of the financial operation of the loan, its respective amortization table and the creation of a Microsoft Excel spreadsheet file. The study was methodologically designed based on a solid bibliography in Financial Mathematics for the preparation of loan amortization tables. Similarly, the development of the code was based on bibliographic and documentary research into the Python programming language and the necessary libraries. There are computational and educational results from the research. On the computational side, we obtained a code capable of generating financial tables and the respective summaries of credit operations, as well as creating the file in Microsoft Excel format. In the educational dimension, the competencies to be developed in vocational education students are mainly those inherent to knowledge and the technical side.

Key-words: *Financial Mathematics, French System, Python, Undergraduate teaching, Vocational Education.*

1. Introdução

O artigo tem como temática a aplicação de linguagem de programação Python à elaboração de tabelas de amortização de operações de crédito, campo da Matemática Financeira e da perícia econômico-financeira [1]. De modo particular, interessa abordar a utilização de Python e de bibliotecas necessárias para elaborar as mencionadas tabelas de amortização, o resumo da operação de modo análogo ao realizado por bancos e instituições financeiras e criar o arquivo de extensão “xlsx”, o qual pode ser posteriormente manipulado no Microsoft Excel, caso desejado pelo usuário, para fins de alterações de apresentação. Trata-se de recorte de uma

Anais da VII Mostra de Docentes em RJJ

pesquisa que aborda as aplicações de Python em disciplinas do eixo gestão e negócios do ensino de graduação tecnológica da Educação Profissional.

A Educação Profissional e Tecnológica (EPT) tem como marca emblemática ser uma modalidade educacional direcionada ao atendimento de demandas e necessidades do setor produtivo, de todas as áreas e setores econômicos e de organizações privadas, públicas e do terceiro setor. A ênfase da EPT está na agilidade, na flexibilidade e no foco de desenvolvimento de competências específicas para atender aos setores econômicos, às organizações e às ocupações que passam a surgir a partir de mudanças tecnológicas. As demandas disruptivas, emergentes e inovativas exigem desta modalidade educacional um olhar atento por parte de *policy makers* e gestores públicos [2] e [3].

Os acelerados processos de digitalização da economia e de informatização da gestão empresarial têm promovido mudanças disruptivas em ao menos três dimensões: na velocidade exigida para se adaptar e agir diante de mudanças tecnológicas, econômicas e culturais, nas competências das pessoas para trabalhar nos novos contextos e situações laborais, e nas estratégias de formação educacional [4]. Dado este contexto, o desenvolvimento de competências no ambiente educacional se torna pauta estratégica como objeto ou diretriz de política educacional. Isto é ainda mais significativo na EPT, contemplando todos os eixos tecnológicos.

Atualmente a modelagem computacional é o “terceiro pilar” da pesquisa científica [5], ao lado da teoria e da experimentação. A computação se tornou um instrumento central na ciência econômica [6] e o aprendizado e uso de linguagens de programação têm se apresentado aceleradamente como tarefas dos economistas [7]. Mais de 70% das universidades *top 100* nos Estados Unidos têm oferecido cursos de programação em Python [8] e muitas empresas no setor financeiro e bancário passaram a adotar Python no lugar do Microsoft Excel [9]. Python ocupa o primeiro lugar entre as linguagens mais usadas no mundo em agosto de 2023 pelo TIOBE Index (<https://www.tiobe.com/tiobe-index/>).

2. Materiais e métodos

2.1. Materiais

O código apresentado foi elaborado no *Google Colaboratory (Colab)*, um Ambiente de Desenvolvimento Integrado (IDE) executado diretamente na página da internet (<https://colab.research.google.com>), no qual se tem um *notebook* como ambiente amigável para executar códigos na linguagem de programação Python. Foram empregadas as bibliotecas *NumPy* (<https://numpy.org>), *NumPy Financial* (<https://numpy.org/numpy-financial/latest/>) e *Pandas* (<https://pandas.pydata.org>) para elaborar o código discutido neste artigo.

2.2. Metodologia

Para a elaboração do código foi realizada pesquisa bibliográfica, referenciada ao longo do artigo, e documental sobre as bibliotecas necessárias. Destacam-se, em termos de orientação metodológica em Matemática Financeira, as obras *Prova Pericial Contábil: teoria e prática* [1], *Matemática Financeira e suas Aplicações* [10], *Tabela Price: mitos e paradigmas* [11] e *Capitalização de Juros em Renegociação de Dívidas: sistemas de amortização* [12]. As diretrizes metodológicas específicas à codificação em Python se devem à *Introduction to Computation and Programming Using Python* [13], *Python for Economists* [14], *Using Python*

Anais da VII Mostra de Docentes em RJI and Google Colab to Teach Undergraduate Microeconomic Theory [9] e *Matemática com Python* [15]. Foram empregadas funções das bibliotecas *NumPy*, para geração de um vetor (*array*) dos períodos; *NumPy Financial*, para o cálculo da taxa de Custo Efetivo Total (CET) por meio da Taxa Interna de Retorno (TIR) e, dessa forma, a aplicação da função “*irr*” e a biblioteca *Pandas*, para a elaboração da tabela em si e a criação do arquivo no formato Microsoft Excel (*xlsx*), utilizando diversas funções.

3. Resultados e Discussão

A linguagem de programação Python apresenta-se como alternativa às planilhas eletrônicas, como o Microsoft Excel, na elaboração de tabelas de sistemas de amortização, frequentemente utilizados nos casos de financiamentos via crédito direto ao consumidor e outras modalidades de operações de crédito. Objetiva-se aqui apresentar um código elaborado em Python que executa três tarefas: i) apresenta um resumo da operação de crédito, semelhante ao apresentado em contratos de instituições financeiras, ii) apresenta a tabela (na tela) demonstrando a evolução período a período da operação de crédito, iii) gera um arquivo “*xlsx*” (planilha Excel).

Aborda-se, de modo particular, o sistema de amortização chamado Sistema Francês, mais conhecido no Brasil como Tabela Price. O Sistema de Amortização Francês (SAF) estabelece prestações iguais (fixas), periódicas e sucessivas compostas de duas partes, sendo uma de juros e outra de capital (amortização) [1]. Esse sistema favorece a administração do fluxo de caixa tanto da perspectiva do credor quanto do devedor, sendo largamente utilizado no mercado financeiro e de capitais no Brasil [12].

A denominação Tabela Price se deve ao britânico Richard Price (1723-1791), que publicou a obra *Observations on Revisionary Payments*, a qual contém as tabelas com os coeficientes de 1 a 100 períodos, com taxas variando de 2% a 10% ao ano, para o cálculo de séries uniformes de pagamentos [11] e [12].

A distinção entre o SAF e a Tabela Price, como empregados na prática financeira brasileira, está no fato de que a última adota o uso da taxa de juro proporcional (linear) simples no lugar da taxa equivalente composta [10]. Quanto a esta peculiaridade, verifica-se na abordagem histórica da obra *Tabela Price: mitos e paradigmas* [11] que Richard Price subdividiu as tabelas de juro composto das anuidades em semestralidades, trimestralidades, mensalidades e renda perpétua. Isso implica que se em certa operação de crédito conste no contrato uma taxa anual de 30% e as capitalizações (prestações) forem mensais, então a taxa mensal será de 2,5% ao mês ($30\% \div 12$ meses), não a taxa equivalente mensal em capitalização composta de 2,2104% a.m. [10]. Observe que o custo efetivo anual dessa operação é equivalente a 34,4889% (ou seja, 2,5% capitalizados de modo exponencial por 12 meses).

As tabelas que demonstram a evolução dos valores monetários no Sistema de Amortização Francês (SAF)/Price têm os seguintes elementos constituintes, em colunas: i) Períodos, ii) Saldo Devedor, iii) Amortização, iv) Juros e v) Prestação (igual para todos os períodos). São características determinantes deste sistema de amortização ter prestações iguais e sucessivas, amortizações do principal crescentes e juros decrescentes ao longo dos períodos e saldo devedor decrescente à razão da amortização em cada período [1], [10], [11] e [12].

A elaboração da tabela segue um algoritmo lógico e matemático, o qual pode ser descrito em etapas. Dado um valor a ser financiado, o Valor Presente (VP ou PV em inglês), uma taxa de juros periódica (*i*) e um período ou prazo de financiamento (*n*), na primeira etapa (Etapa 1) deve-se calcular o valor da prestação/pagamento (PGTO ou PMT, em inglês) periódica/o. A

Anais da VII Mostra de Docentes em RJJ

equação para o cálculo da prestação (PMT) pode ser apresentada alternativamente de duas formas, conforme manipulação algébrica [1] e [12]:

$$PMT = PV \times \frac{i \times (1+i)^n}{(1+i)^n - 1} \quad \text{ou} \quad PMT = PV \times \frac{i}{1 - (1+i)^{-n}} \quad (1)$$

Tendo-se calculado o valor do pagamento periódico (PGTO ou PMT), a Etapa 2 é o cálculo do valor referente aos juros devidos para o primeiro período. De partida, o Saldo Devedor (SD) no momento inicial (período “zero”, momento da contratação) é idêntico ao valor financiado (VP ou PV). Matematicamente, tem-se:

$$J_1 = SD_0 \times i \quad (2)$$

Nos períodos posteriores ao primeiro, a equação de cálculo do Juro para o período “t” pode ser generalizada para:

$$J_t = SD_{t-1} \times i \quad (3)$$

A equação 3 implica que a taxa de juros (i) sempre irá incidir sobre o Saldo Devedor do período imediatamente anterior (t – 1).

A Etapa 3 consiste no cálculo do valor referente à amortização (A) de parte do Saldo Devedor (SD), ou seja, o valor monetário que efetivamente quita parte do principal. A equação para o cálculo da Amortização em um período “t” é oriunda do fato de que o valor da prestação (PGTO ou PMT) é composto de duas partes, uma referente aos juros (J) remuneratórios da concessão do capital financiado e outra inerente à parte que amortiza/quita parcela do capital tomado a crédito. Dessa forma, a equação de cálculo da Amortização (A) no período “t” é dada por:

$$A_t = PMT - J_t \quad (4)$$

A Etapa 4 é o cálculo do Saldo Devedor no período “t”, o qual é o Saldo Devedor do período imediatamente anterior (t – 1) menos a Amortização (A) no período considerado (t). Tem-se:

$$SD_t = SD_{t-1} - A_t \quad (5)$$

Calculado o Saldo Devedor (na Etapa 4) está terminada a rodada de cálculos para um dado período “t”. O período seguinte tem início com o cálculo do valor referente aos juros (J), depois a amortização (A) e, por conseguinte, o novo saldo devedor (SD). Este processo se repete até o último período, quando necessariamente deve ocorrer a quitação integral do saldo devedor.

Em termos de recursos tecnológicos para a realização dos cálculos financeiros inerentes a esta metodologia do SAF/Price, tradicionalmente se tem utilizado a calculadora HP 12C e as planilhas eletrônicas, como o Microsoft Excel.

Vale destacar que as calculadoras financeiras da marca Hewlett-Packard (HP) possuem funções para os cálculos inerentes ao SAF/Price, o que implica dizer que os modelos HP 10B, HP 12C, HP 17BII e HP 19BII estão aptos a executar os referidos cálculos. O modelo HP 12C, mais adotado por profissionais no mercado financeiro, apresenta ao menos três desvantagens, a saber, a necessidade de realizar os cálculos passo a passo (período a período), não haver uma visualização tabular e não ser possível ter uma visualização gráfica, dada a sua própria estrutura

Anais da VII Mostra de Docentes em RJI

física, de tela simples. Não é objetivo deste artigo abordar procedimentos de cálculo nas calculadoras mencionadas. O leitor interessado pode consultar a obra *Matemática Financeira: com o uso das calculadoras HP 12C, HP 19BII, HP 17BII e HP 10B* [16].

Quanto às planilhas eletrônicas, em particular o Microsoft Excel, poder-se-á apresentar algumas considerações após a formalização da apresentação do código elaborado em Python.

O código foi concebido para ser executado no *Google Colaboratory (Colab)*, pois este é um Ambiente de Desenvolvimento Integrado (IDE) executado diretamente na *web*, sendo uma ferramenta gratuita, sem necessidade de instalação [9]. Para fazer uso da função que calcula a Taxa Interna de Retorno (TIR), empregada no cálculo do Custo Efetivo (CET) da operação, torna-se necessário instalar no *Colab* a biblioteca *NumPy Financial* antes de importar a função “*irr*” (*internal rate of return*). A biblioteca *Pandas* é importada para se utilizar diversas funções e elaborar o *DataFrame* (a tabela Price propriamente dita). Da biblioteca *NumPy* é importada a função “*array*” para elaborar o vetor de períodos.

As entradas de dados (*inputs*) são as seguintes: 1) Valor a ser financiado (variável “*valor*”); 2) Valor de uma plausível Tarifa de Abertura de Cadastro (TAC), conforme Resolução n. 3.919/2010 do Banco Central do Brasil (variável “*TAC*”); 3) Prazo da operação em meses (variável “*meses*”) e 4) Taxa anual nominal de juros (variável “*taxaAnualNominal*”). Vale lembrar que a taxa mensal é calculada de modo linear, dividindo-se a taxa anual nominal por 12 (meses) [10].

Os resultados da execução do código, após a inserção dos valores das variáveis, são: 1) Impressão na tela do *Colab Notebook* das principais informações da operação de crédito (Valor total financiado, Valor líquido entregue, Taxa de juros mensal nominal, CET mensal, CET anual, Valor das parcelas mensais, Quantidade de parcelas e Valor total da dívida); 2) A apresentação na tela do *Colab Notebook* da Tabela Price contendo as colunas Períodos, Saldo, Amort, Juro e Prestação, por período; 3) Geração do arquivo na extensão “*xlsx*”, o qual possibilita posteriores manuseios no Microsoft Excel.

A título de demonstração foi tomado o exemplo contido nas páginas 799-800 da obra *Prova Pericial Contábil: teoria e prática* [1], a saber, o financiamento de R\$ 120.000,00 pelo prazo de 12 meses, à taxa nominal de 10,5% ao ano, proporcional a 0,8750% ao mês, sem incidência de tarifas e correção monetária. O resultado gera uma tabela idêntica à exibida na obra, além do resumo da operação e da criação do arquivo de Excel. O código pode ser observado na Figura 1. A saída da execução do código pode ser visualizada na Figura 2. Vale observar que, dada a natureza da aplicação da taxa de juros nominal mensal, o CET anual da operação é igual a 11,0203% e não 10,5% a.a. Se for aplicada uma TAC, o código exibirá os valores percentuais das CET mensal e anual.

É significativo afirmar que a elaboração de uma planilha no Microsoft Excel para produzir os mesmos resultados é consideravelmente complexa, em particular para gerar a flexibilidade de simular prazos distintos e exibir as tabelas. Se for considerada a exibição de um gráfico, não contemplada neste artigo, o nível de complexidade de automatização no Excel aumenta consideravelmente em comparação ao que é possível em Python empregando a biblioteca *Matplotlib* (<https://matplotlib.org>), sendo possível apenas com o emprego de macros/*Visual Basic for Applications* (VBA).

Python requer um aprendizado do tipo *learning by doing* [5] e tem uma curva de aprendizado exponencial, como indicado na literatura. No entanto, exige dos alunos competências relacionadas a conhecimentos prévios necessários às aplicações. Dito de outra forma, é necessário que o aluno já domine consideravelmente bem matemática, além dos tópicos de matemática financeira, para que possa aplicá-los à ferramenta de TI que é o Python.

Fig. 1 – Código para elaboração de Tabela de Amortização Price.

```

1 # Instala a Biblioteca NumPy Financial
2 !pip install numpy-financial

1 # Importa funções e Bibliotecas
2 import pandas as pd
3 from numpy import array
4 from numpy_financial import irr
5 # Insere os dados principais do Financiamento
6 valor = 120_000
7 TAC = 0
8 meses = 12
9 taxaAnualNominal = 10.5
10 # Calcula Capital financiado e percentual
11 capital = valor + TAC
12 valorPerc = (valor / capital) * 100
13 # Gera o vetor de prazo
14 n = array([x for x in range (0, meses + 1)])
15 # Calcula a taxa mensal proporcional
16 taxaMensal = taxaAnualNominal / 12
17 # Calcula a prestação fixa mensal e valor total da dívida
18 prest = capital * taxaMensal/100/(1-(1 + taxaMensal/100)**(-meses))
19 divida = prest * meses
20 # Calcula elementos para a elaboração da Tabela de Amortização
21 saldo = capital
22 saldoList = [capital]
23 juroList = [0]
24 amortList = [0]
25 prestList = [0]
26 for i in range(meses):
27     juro = (saldo * taxaMensal/100)
28     juroList.append(juro)
29     prest = prest
30     prestList.append(prest)
31     amort = prest - juro
32     amortList.append(amort)
33     saldo -= amort
34     saldoList.append(saldo)
35 # Calcula as taxas efetivas mensal e anual pela TIR
36 listaTIR = [-valor] + [prest]*meses
37 tirMensal = irr(listaTIR)
38 tirMensalPerc = tirMensal*100
39 tirAnual = (1+tirMensal)**12-1
40 tirAnualPerc = tirAnual*100
41 # Cria o DataFrame (Tabela) com Pandas
42 price = pd.DataFrame({'Periodos':n, 'Saldo':saldoList, 'Amort':amortList,
43                     'Juro':juroList, 'Prestação':prestList})
44 # Torna a coluna 'Periodos' o index do DataFrame
45 price = price.set_index('Periodos')
46 # Acrescenta a última linha, com os Totais
47 price.loc['Totais'] = price.sum(axis=0)
48 # Aplica o padrão brasileiro de moeda pontuação nas colunas
49 def formato_brasileiro(valor):
50     return f'R$ {valor:,.2f}'.replace('.', '_').replace(',', '.').replace('_', ',')
51 colunas_formatar = ['Saldo', 'Amort', 'Juro', 'Prestação']
52 for col in colunas_formatar:
53     price[col] = price[col].apply(formato_brasileiro)
54 # Retira o termo totais de "Saldo" por não fazer sentido o somatório
55 price.loc['Totais', 'Saldo'] = 'xxx'
56 # Imprime na tela o resumo da operação
57 print(f'Valor total financiado = R$ {capital:,.2f} (100%)')
58 print(f'Valor líquido entregue = R$ {valor:,.2f} ({valorPerc:.2f}% valor total)')
59 print(f'Taxa de juros mensal nominal = {taxaMensal:.4f}%')
60 print(f'CET mensal = {tirMensalPerc:,.4f}%')
61 print(f'CET anual = {tirAnualPerc:,.4f}%')
62 print(f'Valor das parcelas mensais = R$ {prest:,.2f}')
63 print(f'Quantidade de parcelas = {meses}')
64 print(f'Valor total da dívida = R$ {divida:,.2f}')
65 # Visualiza a Tabela pronta
66 display(price)
67 # Gera o arquivo em Microsoft Excel com os índices
68 price.to_excel("tabela_price_MostraRJI2023.xlsx")

```

Fonte: Elaboração própria do autor utilizando *Python* no *Google Colab*.

Fig. 2 – Saída do código para elaboração de Tabela de Amortização Price.

Valor total financiado = R\$ 120,000.00 (100%)
 Valor líquido entregue = R\$ 120,000.00 (100.00% valor total)
 Taxa de juros mensal nominal = 0.8750%
 CET mensal = 0.8750%
 CET anual = 11.0203%
 Valor das parcelas mensais = R\$ 10,577.83
 Quantidade de parcelas = 12
 Valor total da dívida = R\$ 126,933.99

	Saldo	Amort	Juro	Prestação
Periodos				
0	R\$ 120.000,00	R\$ 0,00	R\$ 0,00	R\$ 0,00
1	R\$ 110.472,17	R\$ 9.527,83	R\$ 1.050,00	R\$ 10.577,83
2	R\$ 100.860,97	R\$ 9.611,20	R\$ 966,63	R\$ 10.577,83
3	R\$ 91.165,67	R\$ 9.695,30	R\$ 882,53	R\$ 10.577,83
4	R\$ 81.385,54	R\$ 9.780,13	R\$ 797,70	R\$ 10.577,83
5	R\$ 71.519,83	R\$ 9.865,71	R\$ 712,12	R\$ 10.577,83
6	R\$ 61.567,79	R\$ 9.952,03	R\$ 625,80	R\$ 10.577,83
7	R\$ 51.528,68	R\$ 10.039,11	R\$ 538,72	R\$ 10.577,83
8	R\$ 41.401,72	R\$ 10.126,96	R\$ 450,88	R\$ 10.577,83
9	R\$ 31.186,15	R\$ 10.215,57	R\$ 362,27	R\$ 10.577,83
10	R\$ 20.881,20	R\$ 10.304,95	R\$ 272,88	R\$ 10.577,83
11	R\$ 10.486,08	R\$ 10.395,12	R\$ 182,71	R\$ 10.577,83
12	R\$ 0,00	R\$ 10.486,08	R\$ 91,75	R\$ 10.577,83
Totais	xxx	R\$ 120.000,00	R\$ 6.933,99	R\$ 126.933,99

Fonte: Elaboração própria do autor utilizando *Python* no *Google Colab*.

Do ponto de vista da gestão educacional, os argumentos anteriores implicam que ao considerar uma disciplina de duas horas-aula semanais, como Matemática Financeira, se torna desafiador conciliar o conteúdo curricular original e o treinamento para aplicar a linguagem Python. Como as competências – de conhecimento e técnicas (*hard skills*) – se sobrepõem, uma possibilidade de ação de gestão é ofertar cursos extracurriculares e de extensão de Python e de Matemática e Estatística aplicadas em Python em momentos de aprendizagem distintos aos de disciplinas como Matemática Financeira. Isso é válido para disciplinas como Economia e Análise de Investimentos, entre outras do eixo gestão e negócios.

4. Considerações finais

O intuito deste artigo foi apresentar resultados parciais de uma pesquisa que tem foco na análise de aplicações de Python em disciplinas do eixo gestão e negócios na Educação Profissional de graduação, na área de Economia e Finanças.

A pesquisa possibilitou criar um código versátil, apto a gerar simulações de operações de crédito no Sistema Francês de Amortização/Tabela Price, considerando as variáveis de entrada inerentes. O código possibilita a geração de tabelas e resumos das operações simuladas, criando os respectivos arquivos de Excel. Foram elaborados códigos com correção monetária e no Sistema de Amortização Constante (SAC) [1], não abordados aqui.

O conteúdo reportado apresenta-se como exemplificação das potencialidades que a linguagem de programação Python disponibiliza a profissionais da área de gestão. Por outro

Anais da VII Mostra de Docentes em RJJ

lado, exige ações de gestão educacional para ofertar condições e momentos de aprendizagem e desenvolvimento de competências profissionais aos alunos da Educação Profissional, os quais podem ser na forma de cursos extracurriculares e cursos de extensão.

Referências

- [1] HOOG, W. A. Z. **Prova pericial contábil: teoria e prática**. 17 ed. Curitiba: Juruá, 2022.
- [2] DELGADO, D. M. A educação profissional e tecnológica e a C&T no Brasil: cultura científica e o tripé ensino, pesquisa e extensão. In: FREIRE, E.; VERONA, J. A.; BATISTA, S. S. dos S. (orgs). **Educação profissional e tecnológica: extensão e cultura**. Jundiaí-SP: Paco, 2018.
- [3] DELGADO, D. M; GOMES, L. R. **A educação profissional ao longo do processo de industrialização no Brasil**. São Carlos: EdUFSCar, 2019.
- [4] DELGADO, D. M. **Inovação em política e gestão da educação profissional e tecnológica**. São Paulo: Centro Paula Souza, 2015.
- [5] PUGH, D. R. **Python for research and teaching economics**. Proceedings of the 13yh Python in Science Conference. Available at. <https://conference.scipy.org/proceedings/scipy2014/pdfs/pugh.pdf>. Acesso em: jul. 2023.
- [6] AROUBA, S. B.; FERNÁNDEZ-VILLAVERDE, J. F. **A comparison of programming languages in economics**. NBER Working Paper Series, June 2014. Available at. <https://www.nber.org/papers/w20263>. Acesso em: jul. 2023.
- [7] ROMERO-AGUILLAR, R. **Python for economists**. Available at. <http://randall-romero.com/wp-content/uploads/Notes/Using-Python-CompEcon-English.pdf>. Zietz, J., 2007. Dynamic programming: an introduction by example. J. Econ. Educ. 38 (2), 165–186. Acesso em: jul. 2023.
- [8] SHI, X.; CHEN, Y. New teaching method of Python programming for liberal arts students. **International Journal of Innovation and Research in Educational Sciences**, vol. 7, issue 3, p. 261-271, 2020. Disponível em: <https://www.ijires.org/index.php/issues?view=publication&task=show&id=584>. Acesso em: ago. 2023.
- [9] KUROKI, M. Using Python and Google Colab to teach undergraduate microeconomic theory. **International Review of Economics Education**, 38, 2021. <https://doi.org/10.1016/j.iree.2021.100225>. Acesso em: ago 2023.
- [10] ASSAF NETO, A. **Matemática financeira e suas aplicações**. 15 ed. São Paulo; Atlas, 2022.
- [11] NOGUEIRA, J. M. **Tabela Price: mitos e paradigmas**. 3 ed. Campinas-SP: Millenium, 2013.
- [12] SANDRINI, J. C.; CHEROBIM, A. P. M. S. **Capitalização de juros em renegociação de dívidas: sistemas de amortização**. Curitiba: Juruá, 2016.
- [13] GUTTAG, J. V. **Introduction to computation and programming using python**. 2nd ed. Cambridge-MA: The MIT Press, 2017.
- [14] BELL, A. **Python for economists**. Harvard, 2016. Available at: https://scholar.harvard.edu/files/ambell/files/python_for_economists.pdf. Acesso em: jul. 2023.
- [15] MARCONDES, G. A. B. **Matemática com Python: um guia prático**. São Paulo: Novatec, 2018.
- [16] CAMPOS FILHO, A. **Matemática financeira: com o uso das calculadoras HP 12C, HP 19BII, HP 17BII e HP 10B**. 2 ed. São Paulo: Atlas, 2001.