UNIVERSIDADE FEDERAL DE ALAGOAS
CAMPUS ARAPIRACA
PROGRAMA DE MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL

AUDENIR NUNES PETUBA

Linguagem Python e sua Aplicação em Sistemas de Amortização na Educação Básica

Arapiraca

2024

AUDENIR NUNES PETUBA

Linguagem Python e sua Aplicação em Sistemas de Amortização na Educação Básica

Dissertação apresentada ao Programa de Mestrado Profissional em Matemática em Rede Nacional do *Campus* Arapiraca da Universidade Federal de Alagoas como parte dos requisitos exigidos para a obtenção do grau de Mestre em Matemática.

Orientador: Prof. Dr. Rinaldo Vieira da Silva Júnior

Arapiraca

2024

Universidade Federal de Alagoas – UFAL
Campus Arapiraca
Biblioteca Setorial *Campus* Arapiraca - BSCA

P512l Petuba, Audenir Nunes
Linguagem Python e sua aplicação em sistemas de amortização na educação básica [recurso eletrônico] / Audenir Nunes Petuba. – Arapiraca, 2024.
44 f.: il.

Orientador: Prof. Dr. Rinaldo Vieira da Silva Júnior.
Dissertação (Mestrado Profissional em Matemática em Rede Nacional) - Universidade Federal de Alagoas, *Campus* Arapiraca, Arapiraca, 2024.
Disponível em: Universidade Digital (UD) – UFAL (*Campus* Arapiraca).
Referências: f. 44.

1. Matemática. 2. Educação financeira. 3. Linguagem Python. I. Silva Júnior, Rinaldo Vieira da. II. Título.

CDU 51

Bibliotecário responsável: Márcio Thiago dos Santos Albuquerque
CRB - 4 / 2052

AUDENIR NUNES PETUBA

Linguagem Python e sua Aplicação em Sistemas de Amortização na Educação Básica

Dissertação apresentada à banca examinadora como requisito parcial para a obtenção do Título de Mestre em Matemática, pelo Programa de Pós-Graduação Mestrado Profissional em Matemática do *Campus* Arapiraca da Universidade Federal de Alagoas, aprovada em 13 de setembro de 2024.

BANCA EXAMINADORA

Documento assinado digitalmente
RINALDO VIEIRA DA SILVA JUNIOR
Data: 13/09/2024 14:57:36-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Rinaldo Vieira da Silva Júnior
Orientador (PROFMAT-Arapiraca/UFAL)

Documento assinado digitalmente
ELTHON ALLEX DA SILVA OLIVEIRA
Data: 16/09/2024 14:28:56-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Elthon Allex da Silva Oliveira
Membro interno (PROFMAT-Arapiraca/UFAL)

Documento assinado digitalmente
VANESSA LUCIA DA SILVA
Data: 14/09/2024 09:42:08-0300
Verifique em https://validar.iti.gov.br

Prof. Dra. Vanessa Lúcia da Silva
Membro externo (IFAL/*Campus* Maceió)

Dedico à minha esposa, Janayna Karlla, e à minha filha, Ayla Clarice.

Agradecimentos

Agradeço a Deus por me dar sabedoria e capacidade de superar os desafios e concluir este trabalho.

À minha amada esposa, Janayna Karlla, por todo o apoio durante essa jornada. Por ter me dado o dom de ser pai da nossa filha Ayla Clarice. Amo vocês!

Ao meu orientador, Prof. Dr. Rinaldo Vieira da Silva Júnior, por todo o suporte oferecido a mim, sendo meu orientador desde a época da graduação.

A todos os professores do PROFMAT - UFAL *campus* Arapiraca pelos ensinamentos durante o curso.

Ao Prof. Dr. Elthon Allex da Silva Oliveira e à Profa. Dra. Vanessa Lúcia da Silva por aceitarem compor a banca avaliadora.

Aos meus colegas de curso pela parceria durante essa trajetória.

Resumo

O uso da linguagem de programação Python tem se mostrado cada vez mais relevante, inclusive na Educação Básica, devido à sua facilidade de aprendizado e aplicação em diversas áreas. Este trabalho propõe explorar a aplicação da linguagem Python na compreensão e implementação de sistemas de amortização, um conceito financeiro importante, com o intuito de introduzir noções de matemática financeira de forma prática e intuitiva para estudantes do ensino fundamental e médio. Através de exemplos e exercícios práticos, os alunos poderão compreender os fundamentos dos sistemas de amortização e desenvolver habilidades de programação ao mesmo tempo, preparando-se melhor para tomar decisões financeiras no futuro. Espera-se que essa abordagem interdisciplinar torne o aprendizado mais engajante e significativo.

Palavras-chaves: ensino; educação financeira; sistemas de amortização; linguagem Python.

Abstract

The use of the Python programming language has become increasingly relevant, even in Basic Education, due to its ease of learning and application in various fields. This work proposes to explore the application of Python in the understanding and implementation of amortization systems, an important financial concept, with the aim of introducing notions of financial mathematics in a practical and intuitive way to elementary and high school students. Through practical examples and exercises, students will be able to understand the fundamentals of amortization systems and develop programming skills simultaneously, better preparing themselves to make financial decisions in the future. This interdisciplinary approach is expected to make learning more engaging and meaningful.

Keywords: teaching; financial education; amortization systems; Python.

Resumen

El uso del lenguaje de programación Python se ha vuelto cada vez más relevante, incluso en la Educación Básica, debido a su facilidad de aprendizaje y aplicación en diversas áreas. Este trabajo propone explorar la aplicación del lenguaje Python en la comprensión e implementación de sistemas de amortización, un concepto financiero importante, con el objetivo de introducir nociones de matemáticas financieras de manera práctica e intuitiva para estudiantes de primaria y secundaria. A través de ejemplos y ejercicios prácticos, los estudiantes podrán comprender los fundamentos de los sistemas de amortización y desarrollar habilidades de programación al mismo tiempo, preparándose mejor para tomar decisiones financieras en el futuro. Se espera que este enfoque interdisciplinario haga que el aprendizaje sea más atractivo y significativo.

Palabras clave: enseñanza; educación financiera; sistemas de amortización; lenguaje Python.

Lista de figuras

Lista de tabelas

Sumário

1 Introdução

O presente trabalho foca na análise e comparação de três sistemas de amortização de empréstimos: o Sistema de Amortização Constante (SAC), o Sistema Price e o Sistema de Amortização Crescente (SACRE), além do uso da linguagem de programação Python para simulações e visualizações. Esses sistemas, amplamente utilizados no mercado financeiro, possuem características distintas que influenciam diretamente as finanças dos tomadores de empréstimos e financiamentos de bens.

Este estudo tem como objetivo principal capacitar os estudantes do Ensino Médio a compreender os sistemas de amortização e seus impactos, além de desenvolver habilidades tecnológicas e analíticas por meio do uso de Python. Especificamente, busca-se:

- Ensinar conceitos de matemática financeira relacionados a sistemas de amortização de empréstimos e financiamentos;
- Promover a educação financeira e o pensamento crítico, essenciais para a tomada de decisões conscientes;
- Incentivar o uso de ferramentas tecnológicas, como Python, para criar tabelas e gráficos, auxiliando na visualização e análise de dados.

A inclusão da educação financeira no Ensino Médio justifica-se por sua relevância na formação de jovens para lidar com situações econômicas reais e no fortalecimento de sua autonomia e responsabilidade financeira. Além disso, o uso de Python permite alinhar o ensino de matemática às competências da Base Nacional Comum Curricular (BNCC), que incentiva a aplicação de tecnologias digitais na educação.

O uso de linguagens de programação é tema de competências e habilidades da área de Matemática e suas Tecnologias na Base Nacional Comum Curricular, tais como a Competência Específica 1:

> Utilizar estratégias, conceitos e procedimentos matemáticos para interpretar situações em diversos contextos, sejam atividades cotidianas, sejam fatos das Ciências da Natureza e Humanas, das questões socioeconômicas ou tecnológicas, divulgados por diferentes meios, de modo a contribuir para uma formação geral. (Brasil, 2018, p. 524)

Nesta competência, temos a habilidade "(EM13MAT406) Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática." (Brasil, 2018, p. 531) A escolha pela linguagem Python se justifica pelo fato de se tratar de uma linguagem de programação de sintaxe simples, de compreensão mais acessível para o entendimento humano, sendo frequentemente escolhida como o ponto de partida para quem está tendo os primeiros contatos com programação. Segundo (Costa *et al.*, 2017, p. 258), “o python é indicado para disciplinas introdutórias por ser uma linguagem intuitiva e de fácil uso”.

A questão central investigada neste trabalho é como os diferentes sistemas de amortização impactam o valor das parcelas e o montante total pago pelos tomadores ao longo do tempo, auxiliando na tomada de decisões financeiras mais conscientes. Adicionalmente, busca-se analisar como a utilização de Python pode facilitar o aprendizado desses sistemas, tornando o ensino de matemática financeira mais dinâmico, acessível e interativo.

Analisando alguns Trabalhos de Conclusão de Curso do PROFMAT, pudemos ver que já existem abordagens acerca do ensino de sistemas de amortização a estudantes do Ensino Médio. Em (Andrade, 2015), o autor apresenta a parte teórica dos sistemas de amortização SAC e Price, apresentando em seguida sugere situações-problema a serem resolvidos por estudantes do Ensino Médio, com aplicação das fórmulas e criação de tabelas. O autor termina o trabalho com uma análise do conteúdo de Matemática Financeira em livros didáticos de Matemática do Ensino Médio, elencando pontos positivos e negativos dessas obras.

Em (Batista Júnior, 2014), abordam-se os sistemas de amortização SAC e Price por meio da resolução de situações-problema e da criação de tabelas em planilha eletrônica Excel. Em (Santos, 2015), os sistemas de amortização SAC e Price, entre outros, são discutidos de forma teórica e prática. Há simulações de empréstimos e financiamentos usando planilha eletrônica Excel. O autor termina apresentando propostas de atividades a serem aplicadas no Ensino Médio, com a resolução de problemas e construção de tabelas com o Excel. Em (Sousa, 2016), os sistemas de amortização SAC e Price são espostos por meio da resolução de exemplos. Em seguida, aplica esses sistemas para descrever situações financeiras no cotidiano, analisando contratos de financiamento do ponto de vista do sistema de amortização utilizado, e simulando as parcelas por meio de tabelas construídas em ferramenta de planilha eletrônica.

Diante da análise desses trabalhos pré-existentes, vimos que a linguagem de programação é um campo ainda pouco explorado quando se trata de trabalhos que abordam os sistemas de amortização. Diante dessa lacuna, neste trabalho usamos a linguagem de programação Python para realizar as simulações de empréstimos e financiamentos, construindo tabelas e gráficos usando o espaço do laboratório de informática da escola. Segundo (Abramovay; Castro *et al.*, 2003), “Os laboratórios são espaços onde a teoria se transfigura em realidade, o ensinamento abstrato se concretiza em ação, som ou imagem. Os diversos experimentos desenvolvidos nos laboratórios permitem aos alunos verificar empiricamente os conteúdos ensinados na escola".

Python é uma das linguagens de programação mais acessíveis e versáteis, podendo ser utilizada na educação por sua simplicidade e eficiência. Ao terem contato com a linguagem Python, os estudantes desenvolvem o pensamento lógico e habilidades de resolução de problemas, além de ganharem confiança em suas capacidades tecnológicas. Isso aprimora a compreensão dos sistemas de amortização e promove uma base sólida em programação que pode ser aplicada em diversos contextos acadêmicos e pessoais.

Este estudo é particularmente relevante para a educação financeira de estudantes do Ensino Médio, preparando-os para um mundo financeiramente complexo e interligado. Compreender os sistemas de amortização, como o Sistema de Amortização Constante (SAC) e o Sistema Price, capacita os alunos a tomarem decisões financeiras mais informadas no futuro e promove habilidades críticas e analíticas essenciais para diversos contextos da vida cotidiana. Além disso, a aplicação prática dos conceitos de amortização em simulações de financiamentos e empréstimos, utilizando Python para criar tabelas e gráficos, é realizada em aulas no laboratório de informática. Essa abordagem facilita a assimilação dos conceitos teóricos e torna o aprendizado mais dinâmico e interativo, permitindo aos alunos observar diretamente os impactos dos diferentes sistemas de amortização.

Outros trabalhos demonstram que houve sucesso ao se fazer uso de linguagens de programação como o Python em aulas de Matemática na Educação Básica. Nesse sentido, podemos citar que "no desenvolvimento das atividades usando o Google Colab, foi percebido um interesse por parte dos alunos em conhecer mais sobre a linguagem de programação Python. Os alunos realizaram tarefas além do que foi solicitado, demonstrando autonomia" (Grave, 2021, p. 54) e

> [...] foi possível notar que a utilização da linguagem de programação Python, alinhado ao uso de teorias que propiciem a melhor compreensão dos estudantes, teve impactos relevantes, ocorrendo uma melhoria no processo de compreensão dos estudantes em relação ao período que estes iniciaram o projeto. (Pesente; Matos; Avelino, 2023, p. 1252)

Assim, este trabalho contribui para o fortalecimento da educação financeira no Ensino Médio, preparando os estudantes para enfrentar desafios econômicos com maior segurança e competência, enquanto desenvolvem habilidades de programação que são essenciais no mundo contemporâneo.

No Capítulo 2, apresentamos conceitos elementares de Matemática Financeira, isto é, juros simples e juros compostos, com resoluções de exemplos e representação gráfica. No

Capítulo 3, definem-se os sistemas de amortização SAC, Price e SACRE, realizando simulações de parcelamentos e criação de tabelas na planilha eletrônica Calc. No Capítulo 4, continuamos abordando os sistemas de amortização, mas realizando as simulações por meio de criação de tabelas com uso da linguagem de programação Python. No Capítulo 5 sugerimos sequências didáticas por meio das quais o professor de Matemática do Ensino Médio pode, em aulas no laboratório de informática, levar os estudantes a realizarem simulações de financiamentos e empréstimos com os sistemas de amortização SAC e Price usando a linguagem Python.

2 Noções Básicas de Matemática Financeira

Neste capítulo trataremos os conceitos básicos da Matemática Financeira. Falaremos sobre o regimes de juros simples e seu crescimento linear, e sobre o regime de juros compostos e seu crescimento exponencial.

2.1 Juros Simples

No regime de juros simples a taxa de juros incide apenas sobre o capital inicial, isto é, o valor principal do financiamento ou investimento. Desta forma, a cada período de capitalização, acrescenta-se ao montante um valor constante.

Sejam C o capital inicial, i a taxa de juros, t o tempo da aplicação, isto é, a quantidade de períodos em que o financiamento ou investimento será capitalizado, e J os juros gerados durante todo o tempo da aplicação. A cada período, será gerado um juro de $C \cdot i$. Para obter os juros gerados durante todo o tempo da aplicação devemos multiplicar $C \cdot i$ por t, obtendo assim a fórmula dos juros gerados no regime de juros simples:

$$J = C \cdot i \cdot t \tag{1}$$

O montante M de uma aplicação é a soma do capital inicial e dos juros, então temos:

$$M = C + J \tag{2}$$

Podemos substituir a Equação 1 na Equação 2 e escrever o montante da seguinte forma:

$$M = C + C \cdot i \cdot t$$

$$M = C \cdot (1 + i \cdot t) \tag{3}$$

Podemos ver que os juros e o montante crescem ou diminuem linearmente em função do tempo, uma vez que tanto na Equação 1 quanto na 3, a variável t tem grau 1.

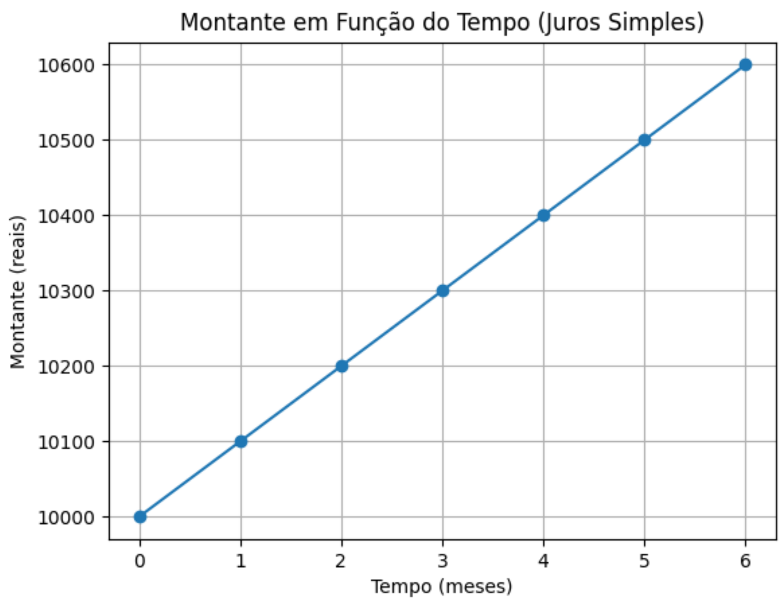
Exemplo 2.1 *Suponha que um investidor aplica R$ 10.000,00, em regime de juros simples, com taxa de 1% ao mês, durante 6 meses. A cada mês, o valor investido será acrescido de R$ 100,00, resultando em juros de R$ 600,00 em 6 meses, tendo-se um montante de R$ 10.600,00 ao final do investimento.*

Esse resultado pode ser obtido usando as Equações 1 e 2, ou, de forma direta, com a Equação 3.

$$M = 10.000 \cdot (1 + 0,01 \cdot 6) = R\$ \ 10.600,00$$

A representação gráfica, gerada em Python, pode ser visualizada na Figura 1.

Figura 1 – Montante em função do tempo para juros simples.



Fonte: o autor (2024)

2.2 *Juros Compostos*

No regime de juros compostos a taxa de juros incide, a cada período, sobre o saldo do investimento no período imediatamente anterior. Desta forma, o montante aumenta exponencialmente em função do tempo.

A cada período, o valor do investimento é calculado multiplicando-se o valor do período anterior por $(1+i)$, então, para o primeiro período atingido pelo investimento, o seu valor estará em $C \cdot (1+i)$. Para um tempo de t períodos, o montante M do investimento será:

$$M = C \cdot (1+i)^t \tag{4}$$

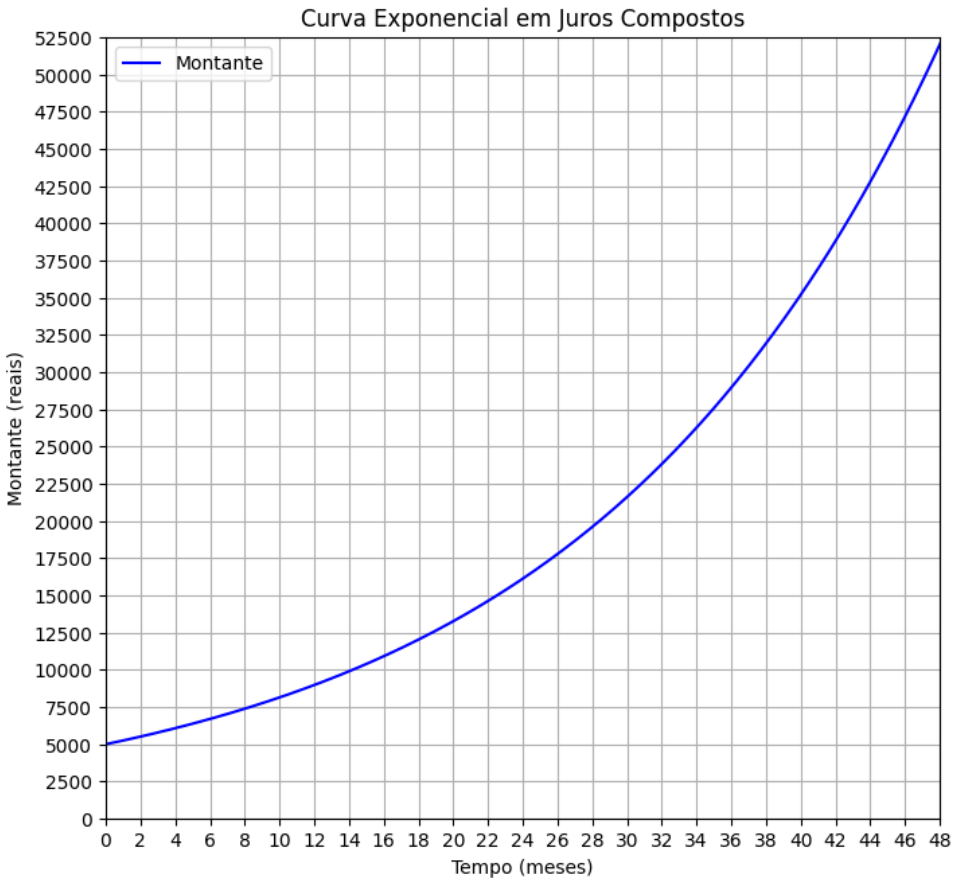
Exemplo 2.2 *Suponha que um investidor aplica R\$ 5.000,00, em regime de juros compostos, com taxa de 5% ao mês, durante 48 meses. No primeiro mês, o valor investido será acrescido de R\$ 250,00, sendo o saldo de R\$ 5.250,00, e nos meses seguintes os juros incidirão sobre o saldo* *do mês imediatamente anterior. Esse investimento renderá juros de R\$ 47.006,35 em 48 meses, tendo-se um montante de R\$ 52.006,35 ao final do investimento.*

Esse resultado pode ser obtido usando a Equação 4.

$$M = 5.000 \cdot (1,05)^{48} = R\$\ 52.006,35$$

No gráfico a seguir (Figura 2) podemos ver a curva que representa o crescimento exponencial do montante em função do tempo.

Figura 2 – Montante em função do tempo para juros compostos.



Fonte: o autor (2024)

3 Sistemas de amortização

Quando adquirimos um financiamento de um imóvel ou de um veículo, ou um empréstimo, pagamos a dívida gerada em prestações, cujos valores podem ser iguais, crescentes ou decrescentes. A definição de quais valores terão essas prestações são feitas pelo sistema de amortização escolhido para a operação financeira realizada. Conhecer como funcionam os sistemas de amortização é importante para que o tomador de um empréstimo ou financiamento possa comparar os valores associados a essas operações financeiras e possa fazer a melhor tomada de decisão ao adquiri-las.

No panorama brasileiro, destacam-se o sistema de amortização constante (SAC), o sistema de amortização Price (ou sistema de amortização francês) e o sistema de amortização crescente (SACRE).

Neste capítulo veremos como funciona cada um desses sistemas de amortização, de qual forma cada um deles determina o valor das prestações a serem pagas por um empréstimo ou financiamento.

3.1 Sistema de Amortização Constante (SAC)

O Sistema de Amortização Constante (SAC) se caracteriza por ter a amortização de valor constante. O valor da amortização é obtido dividindo-se o valor principal do empréstimo ou financiamento pela quantidade de parcelas da operação financeira. Assim, sendo A o valor da amortização no sistema SAC, V o valor principal e n a quantidade de parcelas, o valor de A é calculado fazendo-se:

$$A = \frac{V}{n}. \tag{5}$$

Seja S_k, com $1 \leq k \leq n$, o saldo devedor após o pagamento da k-ésima parcela. Temos que S_k é igual à diferença do saldo devedor no período anterior e da amortização. Assim:

$$S_k = S_{k-1} - A. \tag{6}$$

Sejam J_k os juros na k-ésima parcela e i a taxa de juros por período. O valor de J_k é o produto de S_{k-1} e i, isto é:

$$J_k = S_{k-1} \cdot i. \tag{7}$$

Vale dizer que os saldos devedores, desde o valor principal até a liquidação da dívida, são:

Tabela 1 – Evolução do saldo devedor no sistema SAC

Número da parcela	Saldo devedor
0	$V = S_0$
1	$S_1 = S_0 - A$
2	$S_2 = S_1 - A$
...	...
$n-1$	$S_{n-1} = S_{n-2} - A$
n	$S_n = S_{n-1} - A = 0$

Fonte: o autor (2024)

Finalmente, seja P_k a k-ésima parcela. O valor de P_k é a soma de A e J_k. Então, pela Equação 7, temos:

$$P_k = A + S_{k-1} \cdot i. \tag{8}$$

Exemplo 3.1 *Suponha que um financiamento de R\$ 15.000,00 foi contratado em 23/01/2024, e a primeira parcela com vencimento em 23/02/2024. Considerando que esse financiamento foi feito com sistema SAC e capitalizado mensalmente com taxa de 1,5% para ser pago em 12 parcelas, temos que* $V = R\$15.000,00$, $i = 0,015$ *e* $n = 12$ *e a cada mês a amortização será:*

$$A = \frac{15.000}{12} = R\$\ 1.250,00.$$

A cada parcela que for paga, o saldo principal é reduzido em R\$ 1.250,00, sendo

$$\begin{aligned}
S_0 &= R\$15.000,00\\
S_1 &= 15.000 - 1.250 = R\$13.750,00,\\
S_2 &= 13.750 - 1.250 = R\$12.500,00,\\
S_3 &= 12.500 - 1.250 = R\$11.250,00,\\
&\vdots\\
S_{11} &= 2.500 - 1.250 = R\$\ 1.250,00\\
S_{12} &= 1.250 - 1.250 = R\$0,00.
\end{aligned}$$

Os juros cobrados em cada parcela também serão decrescentes, sendo

$$\begin{aligned}
J_1 &= 15.000 \cdot 0,015 = R\$\ 225,00,\\
J_2 &= 13.750 \cdot 0,015 = R\$\ 206,25,\\
J_3 &= 11.250 \cdot 0,015 = R\$\ 187,50,\\
&\vdots\\
J_{11} &= 2.500 \cdot 0,015 = R\$\ 37,50,\\
J_{12} &= 1.250 \cdot 0,015 = R\$\ 18,75.
\end{aligned}$$

O valor de cada parcela será dado pela soma da amortização e dos juros:

$$P_1 = 1.250 + 225 = R\$\ 1.475,00,$$
$$P_2 = 1.250 + 206,25 = R\$\ 1.456,25,$$
$$P_3 = 1.250 + 187,50 = R\$\ 1.437,50,$$
$$\vdots$$
$$P_{11} = 1.250 + 37,50 = R\$\ 1.287,50,$$
$$P_{12} = 1.250 + 18,75 = R\$\ 1.268,75.$$

Podemos ver as parcelas detalhadas na Figura 3, que mostra a tabela elaborada na planilha eletrônica Calc.

Figura 3 – Exemplo de sistema da amortização SAC.

Número Da parcela	Amortização	Juros	Parcela	Saldo devedor
0	R$ 0,00	R$ 0,00	R$ 0,00	R$ 15.000,00
1	R$ 1.250,00	R$ 225,00	R$ 1.475,00	R$ 13.750,00
2	R$ 1.250,00	R$ 206,25	R$ 1.456,25	R$ 12.500,00
3	R$ 1.250,00	R$ 187,50	R$ 1.437,50	R$ 11.250,00
4	R$ 1.250,00	R$ 168,75	R$ 1.418,75	R$ 10.000,00
5	R$ 1.250,00	R$ 150,00	R$ 1.400,00	R$ 8.750,00
6	R$ 1.250,00	R$ 131,25	R$ 1.381,25	R$ 7.500,00
7	R$ 1.250,00	R$ 112,50	R$ 1.362,50	R$ 6.250,00
8	R$ 1.250,00	R$ 93,75	R$ 1.343,75	R$ 5.000,00
9	R$ 1.250,00	R$ 75,00	R$ 1.325,00	R$ 3.750,00
10	R$ 1.250,00	R$ 56,25	R$ 1.306,25	R$ 2.500,00
11	R$ 1.250,00	R$ 37,50	R$ 1.287,50	R$ 1.250,00
12	R$ 1.250,00	R$ 18,75	R$ 1.268,75	R$ 0,00

Fonte: o autor (2024)

3.1.1 Sistema SAC pós-fixado

Em operações financeiras de longo prazo, como em financiamentos imobiliários, costuma-se incluir uma correção monetária, a fim de prevenir perdas monetárias em decorrência da degradação do valor do dinheiro com o decorrer do tempo.

Um fator de correção monetária comumente usado no mercado financeiro brasileiro é a taxa referencial (TR).

Se um empréstimo ou financiamento é tomado com sistema SAC e correção monetária, a cada período o saldo devedor será corrigido pelo fator de correção, com a taxa vigente naquele período.

Seja r_k o fator de correção monetária, a cada período k o saldo devedor corrigido C_k será:

$$C_k = S_k \cdot (1 + r_k). \tag{9}$$

A amortização A_k, os juros J_k, a parcela P_k e o saldo devedor S_k, no período k, são dados, respectivamente, por:

$$A_k = \frac{C_{k-1}}{-k+n+1}, \tag{10}$$

$$J_k = C_{k-1} \cdot i, \tag{11}$$

$$P_k = A_k + J_k \tag{12}$$

e

$$S_k = C_{k-1} - A_k. \tag{13}$$

Exemplo 3.2 *Suponha que o financiamento do exemplo 3.1 tenha sido feito com correção monetária de* 1% *a cada mês. Assim, temos,* $C_0 = 15.000 \cdot 1,01 = R\$\ 15.150,00$ *e, para o primeiro mês:*

$$A_1 = \frac{15.150}{-1+12+1} = \frac{15.150}{12} = R\$\ 1.262,50,$$

$$J_1 = 15.150 \cdot 0,015 = R\$\ 227,25,$$

$$P_1 = 1.262,50 + 227,25 = R\$\ 1.489,75$$

e

$$S_1 = 15.150 - 1.262,50 = R\$\ 13.887,50.$$

$$C_1 = 13.887,50 \cdot 0,01 = R\$\ 14.026,38$$

$$\vdots$$

Nos meses seguintes esse processo se repete de forma análoga, até o 12º mês, em que a dívida é liquidada.

Podemos ver as parcelas detalhadas na Figura 4, que mostra a tabela elaborada na planilha eletrônica Calc.

3.2 Sistema de Amortização Price

No sistema de amortização Price, ou tabela Price, ou sistema de amortização Francês, temos parcelas de valor constante. O valor da parcela é calculado de modo que a soma de todos os pagamentos seja equivalente ao pagamento integral da dívida.

Figura 4 – Exemplo de sistema da amortização SAC pós-fixado.

Número Da parcela	Amortização	Juros	Parcela	Saldo devedor	Saldo corrigido	TR
0	R$ 0,00	R$ 0,00	R$ 0,00	R$ 15.000,00	R$ 15.150,00	1,00%
1	R$ 1.262,50	R$ 227,25	R$ 1.489,75	R$ 13.887,50	R$ 14.026,38	1,00%
2	R$ 1.275,13	R$ 210,40	R$ 1.485,52	R$ 12.751,25	R$ 12.878,76	1,00%
3	R$ 1.287,88	R$ 193,18	R$ 1.481,06	R$ 11.590,89	R$ 11.706,80	1,00%
4	R$ 1.300,76	R$ 175,60	R$ 1.476,36	R$ 10.406,04	R$ 10.510,10	1,00%
5	R$ 1.313,76	R$ 157,65	R$ 1.471,41	R$ 9.196,34	R$ 9.288,30	1,00%
6	R$ 1.326,90	R$ 139,32	R$ 1.466,22	R$ 7.961,40	R$ 8.041,02	1,00%
7	R$ 1.340,17	R$ 120,62	R$ 1.460,78	R$ 6.700,85	R$ 6.767,85	1,00%
8	R$ 1.353,57	R$ 101,52	R$ 1.455,09	R$ 5.414,28	R$ 5.468,43	1,00%
9	R$ 1.367,11	R$ 82,03	R$ 1.449,13	R$ 4.101,32	R$ 4.142,33	1,00%
10	R$ 1.380,78	R$ 62,13	R$ 1.442,91	R$ 2.761,56	R$ 2.789,17	1,00%
11	R$ 1.394,59	R$ 41,84	R$ 1.436,42	R$ 1.394,59	R$ 1.408,53	1,00%
12	R$ 1.408,53	R$ 21,13	R$ 1.429,66	R$ 0,00	R$ 0,00	1,00%

Fonte: o autor (2024)

Assim, sejam P o valor da parcela de um empréstimo ou financiamento com sistema Price, V o valor principal, i a taxa de juros e n a quantidade de parcelas a serem pagas. Transferindo os pagamentos para a data focal da última parcela, temos:

$$P\cdot(1+i)^{n-1}+P\cdot(1+i)^{n-2}+...+P\cdot(1+i)^{2}+P\cdot(1+i)+P=V\cdot(1+i)^{n} \tag{14}$$

Note que o primeiro membro da Equação 14 é a soma dos n primeiros termos de uma progressão geométrica de razão $1+i$. Considerando P o primeiro termo dessa P.G., segue que:

$$P\cdot\frac{(1+i)^{n}-1}{(1+i)-1}=V\cdot(1+i)^{n}$$

$$P\cdot\frac{(1+i)^{n}-1}{i}=V\cdot(1+i)^{n}$$

$$P=\frac{V\cdot(1+i)^{n}\cdot i}{(1+i)^{n}-1} \tag{15}$$

Portanto, o valor da parcela a ser paga no sistema Price é dado pela Equação 15.

Sejam J_k os juros na k-ésima parcela, i a taxa de juros por período e S_{k-1} o saldo devedor após o pagamento da (k-1)-ésima parcela. O valor dos juros é dado por:

$$J_k=S_{k-1}\cdot i. \tag{16}$$

Seja A_k a amortização na k-ésima parcela. O valor de A_k é dado pela diferença entre os valores da parcela e dos juros, isto é:

$$A_k=P-J_k$$

$$A_k=P-S_{k-1}\cdot i. \tag{17}$$

Note que, como o saldo devedor é decrescente, e os juros incidem diretamente sobre o saldo devedor, então os juros também são decrescentes, enquanto a amortização é crescente ao longo do tempo.

Exemplo 3.3 *Suponha que um financiamento de R\$ 20.000,00 foi contratado em 23/01/2024, e a primeira parcela com vencimento em 23/02/2024. Considerando que esse financiamento foi feito com sistema Price e capitalizado mensalmente com taxa de 1,7% para ser pago em 12 parcelas, temos que* $V = R\$20.000,00$*,* $i = 0,017$ *e* $n = 12$*. O valor de cada parcela desse financiamento será:*

$$P = \frac{V \cdot (1+i)^n \cdot i}{(1+i)^n - 1}$$

$$P = \frac{20.000 \cdot (1,017)^{12} \cdot 0,017}{(1,017)^{12} - 1}$$

$$P = R\$\ 1.856,52.$$

Os juros cobrados na primeira parcela são dados por:

$$J_1 = 20.000 \cdot 0,017 = R\$\ 340,00,$$

e a respectiva amortização será:

$$A_1 = 1.856,52 - 340 = 1.516,52.$$

Podemos ver as parcelas detalhadas na Figura 5, que mostra a tabela elaborada na planilha eletrônica Calc.

Figura 5 – Exemplo de sistema da amortização Price.

Número Da parcela	Parcela	Juros	Amortização	Saldo devedor
0	R$ 0,00	R$ 0,00	R$ 0,00	R$ 20.000,00
1	R$ 1.856,52	R$ 340,00	R$ 1.516,52	R$ 18.483,48
2	R$ 1.856,52	R$ 314,22	R$ 1.542,30	R$ 16.941,18
3	R$ 1.856,52	R$ 288,00	R$ 1.568,52	R$ 15.372,66
4	R$ 1.856,52	R$ 261,34	R$ 1.595,19	R$ 13.777,47
5	R$ 1.856,52	R$ 234,22	R$ 1.622,30	R$ 12.155,17
6	R$ 1.856,52	R$ 206,64	R$ 1.649,88	R$ 10.505,28
7	R$ 1.856,52	R$ 178,59	R$ 1.677,93	R$ 8.827,35
8	R$ 1.856,52	R$ 150,06	R$ 1.706,46	R$ 7.120,90
9	R$ 1.856,52	R$ 121,06	R$ 1.735,47	R$ 5.385,43
10	R$ 1.856,52	R$ 91,55	R$ 1.764,97	R$ 3.620,46
11	R$ 1.856,52	R$ 61,55	R$ 1.794,97	R$ 1.825,49
12	R$ 1.856,52	R$ 31,03	R$ 1.825,49	-R$ 0,00

Fonte: o autor (2024)

3.2.1 Sistema Price pós-fixado

Se um empréstimo ou financiamento é tomado com sistema Price e correção monetária, a cada período o saldo devedor será corrigido pelo fator de correção, com a taxa vigente naquele período.

No contexto brasileiro, quando se faz um financiamento imobiliário pela Caixa Econômica Federal, os valores das parcelas são calculados com o Sistema Price, e o saldo devedor é corrigido pela taxa referencial (TR).

Seja r_k o fator de correção monetária, a cada período k o saldo devedor corrigido C_k será:

$$C_k = S_k \cdot (1 + r_k). \tag{18}$$

A parcela P_k, os juros J_k, a amortização A_k e o saldo devedor S_k, no período k, são dados, respectivamente, por:

$$P_k = \frac{C_{k-1} \cdot (1+i)^{-k+n+1} \cdot i}{(1+i)^{-k+n+1} - 1}, \tag{19}$$

$$J_k = C_{k-1} \cdot i, \tag{20}$$

$$A_k = P_k - J_k \tag{21}$$

e

$$S_k = C_{k-1} - A_k. \tag{22}$$

Exemplo 3.4 *Suponha que o financiamento do exemplo 3.1 tenha sido feito com correção monetária de* 1% *a cada mês. Assim, temos,* $C_0 = 20.000 \cdot 1,01 = R\$\ 20.200,00$ *e, para o primeiro mês:*

$$P_1 = \frac{20.200 \cdot (1,017)^{-1+12+1} \cdot 0,017}{(1,017)^{-1+12+1} - 1} = R\$\ 1.875,09,$$

$$J_1 = 20.200 \cdot 0,017 = R\$\ 343,40,$$

$$A_1 = 1.875,09 - 343,40 = R\$\ 1.531,69$$

e

$$S_1 = 20.200 - 1.531,69 = R\$\ 18.668,31.$$

Nos meses seguintes esse processo se repete de forma análoga, até o 12º mês, em que a dívida é liquidada.

Podemos ver as parcelas detalhadas na Figura 6, que mostra a tabela elaborada na planilha eletrônica Calc.

Figura 6 – Exemplo de sistema da amortização Price pós-fixado.

Número Da Parcela	Parcela	Juros	Amortização	Saldo Devedor	Saldo Corrigido	TR
0	R$ 0,00	R$ 0,00	R$ 0,00	R$ 20.000,00	R$ 20.200,00	1,00%
1	R$ 1.875,09	R$ 343,40	R$ 1.531,69	R$ 18.668,31	R$ 18.855,00	1,00%
2	R$ 1.893,84	R$ 320,53	R$ 1.573,30	R$ 17.281,69	R$ 17.454,51	1,00%
3	R$ 1.912,78	R$ 296,73	R$ 1.616,05	R$ 15.838,46	R$ 15.996,85	1,00%
4	R$ 1.931,90	R$ 271,95	R$ 1.659,96	R$ 14.336,89	R$ 14.480,26	1,00%
5	R$ 1.951,22	R$ 246,16	R$ 1.705,06	R$ 12.775,20	R$ 12.902,95	1,00%
6	R$ 1.970,73	R$ 219,35	R$ 1.751,38	R$ 11.151,57	R$ 11.263,09	1,00%
7	R$ 1.990,44	R$ 191,47	R$ 1.798,97	R$ 9.464,12	R$ 9.558,76	1,00%
8	R$ 2.010,35	R$ 162,50	R$ 1.847,85	R$ 7.710,91	R$ 7.788,02	1,00%
9	R$ 2.030,45	R$ 132,40	R$ 1.898,05	R$ 5.889,97	R$ 5.948,86	1,00%
10	R$ 2.050,75	R$ 101,13	R$ 1.949,62	R$ 3.999,24	R$ 4.039,23	1,00%
11	R$ 2.071,26	R$ 68,67	R$ 2.002,59	R$ 2.036,64	R$ 2.057,01	1,00%
12	R$ 2.091,97	R$ 34,97	R$ 2.057,01	-R$ 0,00	-R$ 0,00	1,00%

Fonte: o autor (2024)

3.3 Sistema de Amortização Crescente (SACRE)

No sistema de amortização crescente (SACRE), a parcela no primeiro ano é constante, cujo valor é calculado pela divisão do valor principal pela quantidade de parcelas a serem pagas e somando-se este resultado aos juros do primeiro período. A cada ano seguinte, a parcela permanece constante, porém com um valor menor em relação ao ano anterior.

A amortização é a diferença entre o valor da parcela e os juros de cada período. Assim, os juros são decrescentes, e as amortizações possuem, a cada um ano de pagamentos, valores crescentes.

A cada ano, o valor da parcela é recalculado usando-se o saldo devedor após o último pagamento no ano ano anterior. Este processo se repete a cada ano de pagamento das parcelas.

Exemplo 3.5 *Suponha que um financiamento de R$ 18.000,00 foi contratado em 23/01/2024, e a primeira parcela com vencimento em 23/02/2024. Considerando que esse financiamento foi feito com sistema SACRE e capitalizado mensalmente com taxa de 1,25% para ser pago em 36 parcelas, temos que* $V = R\$18.000,00$, $i = 0,0125$ *e* $n = 36$. *Os juros no primeiro mês serão:*

$$J_1 = 18.000 \cdot 0,0125 = R\$\ 225,00.$$

O valor da parcela desse financiamento no primeiro ano será:

$$P_1 = P_2 = ... = P_{12} = 18.000/36 + 225 = 500 + 225 = R\$\ 725,00.$$

O valor da amortização no primeiro mês será:

$$A_1 = 725 - 225 = R\$\ 500,00,$$

e o saldo devedor passa a ser

$$S_1 = 18.000 - 500 = R\$\ 17.500,00.$$

No segundo mês, teremos:

$$J_2 = 17.500 \cdot 0,0125 = R\$\ 218,75;$$

$$A_2 = 725 - 218,75 = R\$\ 506,25;$$

$$S_2 = 17.500 - 506,25 = R\$\ 16.993,75.$$

Após os doze primeiros pagamentos, recalcula-se o valor da parcela sobre o saldo devedor, que será $S_{12} = R\$\ 11.569,82$*. Daí, teremos:*

$$J_{13} = 11.569,82 \cdot 0,0125 = R\$\ 144,62;$$

$$P_{13} = P_{14} = ... = P_{24} = 11.569,82/24 + 144,62 = R\$\ 626,70;$$

$$A_{13} = 626,70 - 144,62 = R\$\ 482,08.$$

Após os 24 primeiros pagamentos, o saldo devedor será $S_{24} = R\$\ 5.370,15$*. Daí, teremos:*

$$J_{25} = 5.370,15 \cdot 0,0125 = R\$\ 67,13;$$

$$P_{25} = P_{26} = ... = P_{36} = 5.370,15/12 + 67,13 = R\$\ 514,64;$$

$$A_{25} = 514,64 - 67,13 = R\$\ 477,51.$$

Podemos ver as parcelas detalhadas na Figura 7, que mostra a tabela elaborada na planilha eletrônica Calc.

Note que a última amortização é maior que o saldo devedor anterior, o que faz com que o último saldo devedor seja $-R\$\ 385,02$*, ao invés de* $R\$\ 0,00$*. Isto significa que a instituição financeira deverá devolver* $R\$\ 385,02$ *ao tomador do financiamento, e este valor chamamos de* ***resíduo****.*

Figura 7 – Exemplo de sistema da amortização SACRE.

Número Da parcela	Parcela	Juros	Amortização	Saldo devedor
0	R$ 0,00	R$ 0,00	R$ 0,00	R$ 18.000,00
1	R$ 725,00	R$ 225,00	R$ 500,00	R$ 17.500,00
2	R$ 725,00	R$ 218,75	R$ 506,25	R$ 16.993,75
3	R$ 725,00	R$ 212,42	R$ 512,58	R$ 16.481,17
4	R$ 725,00	R$ 206,01	R$ 518,99	R$ 15.962,19
5	R$ 725,00	R$ 199,53	R$ 525,47	R$ 15.436,71
6	R$ 725,00	R$ 192,96	R$ 532,04	R$ 14.904,67
7	R$ 725,00	R$ 186,31	R$ 538,69	R$ 14.365,98
8	R$ 725,00	R$ 179,57	R$ 545,43	R$ 13.820,56
9	R$ 725,00	R$ 172,76	R$ 552,24	R$ 13.268,31
10	R$ 725,00	R$ 165,85	R$ 559,15	R$ 12.709,17
11	R$ 725,00	R$ 158,86	R$ 566,14	R$ 12.143,03
12	R$ 725,00	R$ 151,79	R$ 573,21	R$ 11.569,82
13	R$ 626,70	R$ 144,62	R$ 482,08	R$ 11.087,74
14	R$ 626,70	R$ 138,60	R$ 488,10	R$ 10.599,64
15	R$ 626,70	R$ 132,50	R$ 494,20	R$ 10.105,44
16	R$ 626,70	R$ 126,32	R$ 500,38	R$ 9.605,06
17	R$ 626,70	R$ 120,06	R$ 506,64	R$ 9.098,42
18	R$ 626,70	R$ 113,73	R$ 512,97	R$ 8.585,45
19	R$ 626,70	R$ 107,32	R$ 519,38	R$ 8.066,07
20	R$ 626,70	R$ 100,83	R$ 525,87	R$ 7.540,20
21	R$ 626,70	R$ 94,25	R$ 532,45	R$ 7.007,76
22	R$ 626,70	R$ 87,60	R$ 539,10	R$ 6.468,65
23	R$ 626,70	R$ 80,86	R$ 545,84	R$ 5.922,81
24	R$ 626,70	R$ 74,04	R$ 552,66	R$ 5.370,15
25	R$ 514,64	R$ 67,13	R$ 447,51	R$ 4.922,64
26	R$ 514,64	R$ 61,53	R$ 453,11	R$ 4.469,53
27	R$ 514,64	R$ 55,87	R$ 458,77	R$ 4.010,76
28	R$ 514,64	R$ 50,13	R$ 464,50	R$ 3.546,26
29	R$ 514,64	R$ 44,33	R$ 470,31	R$ 3.075,94
30	R$ 514,64	R$ 38,45	R$ 476,19	R$ 2.599,75
31	R$ 514,64	R$ 32,50	R$ 482,14	R$ 2.117,61
32	R$ 514,64	R$ 26,47	R$ 488,17	R$ 1.629,44
33	R$ 514,64	R$ 20,37	R$ 494,27	R$ 1.135,17
34	R$ 514,64	R$ 14,19	R$ 500,45	R$ 634,72
35	R$ 514,64	R$ 7,93	R$ 506,71	R$ 128,02
36	R$ 514,64	R$ 1,60	R$ 513,04	-R$ 385,02

Fonte: o autor (2024)

4 Linguagem Python e Sistemas de Amortização

Neste capítulo, continuaremos explorando os Sistemas de Amortização, destacando sua importância na análise financeira e apresentando uma abordagem prática usando a linguagem de programação Python. A escolha do Python como ferramenta de implementação não apenas simplifica a aplicação dos conceitos teóricos, mas também proporciona flexibilidade e facilidade na manipulação de dados.

Ao longo das próximas seções, abordaremos diferentes métodos de amortização e demonstraremos como implementá-los eficientemente em Python. Além disso, abordaremos questões práticas, como a visualização dos resultados e a análise de cenários, utilizando recursos disponíveis nessa linguagem.

4.1 Sistema SAC em Linguagem Python

Vamos utilizar um código em Python (Figura 8), no qual criamos uma classe chamada ”Tabela_SAC”. A classe inclui um método para gerar e imprimir uma tabela de amortização com informações sobre cada parcela. A definição da classe está na *linha 1*.

O método de inicialização ”__init__” é o construtor da classe, usado para inicializar os atributos quando um objeto da classe é criado. Neste caso, ele recebe três parâmetros: *principal* (valor principal do empréstimo ou financiamento), *taxa* (taxa de juros por período, a qual é convertida para decimal ao ser dividida por 100) e *parcelas* (a quantidade de parcelas, sendo uma a cada período), nas linhas de 2 a 5.

O método “gerar_tabela” gera e imprime a tabela de amortização, usando o loop *for* para iterar sobre cada período (parcela), calculando os juros, a amortização (de acordo com a Equação 5), a parcela total e o saldo devedor restante, nas linhas de 7 a 26.

Nas linhas de 29 e 30, um objeto ”sac_tabela” é criado usando a classe ”Tabela_SAC” com valores específicos para o principal, taxa de juros e número de parcelas. Em seguida, o método ”gerar_tabela” é chamado para imprimir a tabela de amortização. O resultado final será uma tabela de amortização formatada e impressa no console com informações sobre cada parcela ao longo do tempo, seguindo o método SAC.

Os valores usados são os mesmos do exemplo 3.1. Para compilar o código usamos o Google Colab. O resultado é a tabela mostrada na Figura 9.

Figura 8 – Código para criar tabela de amortização SAC em Python.

```python
class Tabela_SAC:
    def __init__(self, principal, taxa, parcelas):
        self.principal = principal
        self.taxa = taxa / 100.0  # Convertendo a taxa para decimal
        self.parcelas = parcelas

    def gerar_tabela(self):
        dash = '-' * 68
        amortizacao = self.principal / self.parcelas
        saldo_devedor = self.principal

        print(dash)
        print('{:<10}{:<15}{:<15}{:<15}{:<20}'.format('Parcela', 'Juros', 'Amortização', 'Parcela', 'Saldo Devedor'))
        print(dash)

        # Adiciona a linha inicial com valores zero
        print('{:<10}{:<15.2f}{:<15.2f}{:<15.2f}{:<20.2f}'.format(0, 0.0, 0.0, 0.0, self.principal))

        for periodo in range(1, self.parcelas + 1):
            juros = saldo_devedor * self.taxa
            parcela = juros + amortizacao
            saldo_devedor -= amortizacao

            print('{:<10}{:<15.2f}{:<15.2f}{:<15.2f}{:<20.2f}'.format(periodo, juros, amortizacao, parcela, saldo_devedor))

        print(dash)

# Exemplo de uso da classe
sac_tabela = Tabela_SAC(principal=15000, taxa=1.5, parcelas=12)
sac_tabela.gerar_tabela()
```

Fonte: o autor (2024)

Figura 9 – Tabela de amortização SAC em Python.

Parcela	Juros	Amortização	Parcela	Saldo Devedor
0	0.00	0.00	0.00	15000.00
1	225.00	1250.00	1475.00	13750.00
2	206.25	1250.00	1456.25	12500.00
3	187.50	1250.00	1437.50	11250.00
4	168.75	1250.00	1418.75	10000.00
5	150.00	1250.00	1400.00	8750.00
6	131.25	1250.00	1381.25	7500.00
7	112.50	1250.00	1362.50	6250.00
8	93.75	1250.00	1343.75	5000.00
9	75.00	1250.00	1325.00	3750.00
10	56.25	1250.00	1306.25	2500.00
11	37.50	1250.00	1287.50	1250.00
12	18.75	1250.00	1268.75	0.00

Fonte: o autor (2024)

4.2 *Sistema SAC pós-fixado em Linguagem Python*

No código para imprimir uma tabela de amortização SAC pós-fixado, criamos a classe “Tabela_SAC_pos” (Figura 10), na qual utilizamos os comandos da linha 10, que recebe e armazena a taxa de correção por período do empréstimo ou financiamento.

No loop *for*, nas linhas de 19 a 24, definimos o saldo corrigido, calculado sobre a taxa de correção, os juros, também calculado sobre a taxa de correção, e o valor da amortização é calculado conforme a Equação 10.

Figura 10 – Código para criar tabela de amortização SAC pós-fixado em Python.

```
1 class Tabela_SAC_pos:
2     def __init__(self, principal, taxa, parcelas):
3         self.principal = principal
4         self.taxa = taxa / 100.0  # Convertendo a taxa para decimal
5         self.parcelas = parcelas
6
7     def gerar_tabela(self):
8         dash = '-' * 90
9         saldo_devedor = self.principal
10        tr = float(input("Qual é a taxa de correção?     "))
11
12        print(dash)
13        print('{:<10}{:<15}{:<15}{:<15}{:<20}{:<20}'.format('Parcela', 'Juros', 'Amortização', 'Parcela', 'Saldo Devedor', 'Saldo Corrigido'))
14        print(dash)
15
16        # Adiciona a linha inicial com valores zero
17        print('{:<10}{:<15.2f}{:<15.2f}{:<15.2f}{:<20.2f}{:<20.2f}'.format(0, 0.0, 0.0, 0.0, self.principal, self.principal * (1 + tr / 100)))
18
19        for periodo in range(1, self.parcelas + 1):
20            saldo_corrigido = saldo_devedor * (1 + tr / 100)
21            juros = (saldo_corrigido * self.taxa)
22            amortizacao = saldo_corrigido / (-periodo + self.parcelas + 1)
23            parcela = juros + amortizacao
24            saldo_devedor = saldo_corrigido - amortizacao
25
26            print('{:<10}{:<15.2f}{:<15.2f}{:<15.2f}{:<20.2f}{:<20.2f}'.format(periodo, juros, amortizacao,\
27                                                                               parcela, saldo_devedor, saldo_devedor * (1 + tr / 100)))
28
29        print(dash)
30
31 # Exemplo de uso da classe
32 sac_tabela_pos = Tabela_SAC_pos(principal=15000, taxa=1.5, parcelas=12)
33 sac_tabela_pos.gerar_tabela()
```

Fonte: o autor (2024)

Esse código gera e imprime uma tabela que descreve as parcelas de um empréstimo ou financiamento com sistema SAC pós-fixado. Os valores são os mesmos do exemplo 3.2. A Figura 11 mostra essa tabela, gerada com o Google Colab.

Figura 11 – Tabela de amortização SAC pós-fixado em Python.

Parcela	Juros	Amortização	Parcela	Saldo Devedor	Saldo Corrigido
0	0.00	0.00	0.00	15000.00	15150.00
1	227.25	1262.50	1489.75	13887.50	14026.38
2	210.40	1275.12	1485.52	12751.25	12878.76
3	193.18	1287.88	1481.06	11590.89	11706.80
4	175.60	1300.76	1476.36	10406.04	10510.10
5	157.65	1313.76	1471.41	9196.34	9288.30
6	139.32	1326.90	1466.22	7961.40	8041.02
7	120.62	1340.17	1460.78	6700.85	6767.85
8	101.52	1353.57	1455.09	5414.28	5468.43
9	82.03	1367.11	1449.13	4101.32	4142.33
10	62.13	1380.78	1442.91	2761.56	2789.17
11	41.84	1394.59	1436.42	1394.59	1408.53
12	21.13	1408.53	1429.66	0.00	0.00

Fonte: o autor (2024)

4.3 Sistema Price em Linguagem Python

Vamos utilizar um código em Python, no qual criamos uma classe chamada “Tabela_Price” (Figura 12). Inicialmente, definimos a classe e a construímos com o método __init__ de forma análoga ao que fizemos para a tabela do sistema SAC, inserindo os parâmetros *principal*, *taxa* e *parcelas*.

No método “gerar_tabela”, que gera e imprime a tabela de amortização, estabelecemos o cálculo da parcela de acordo com a Equação 15. No loop *for*, calculam-se os juros, a amortização e o saldo devedor restante.

Nas linhas de 1 a 5, um objeto “price_tabela” é criado usando a classe “Tabela_Price” com valores específicos para o valor principal, a taxa de juros e o número de parcelas. Em seguida, o método “gerar_tabela” é chamado para imprimir a tabela de amortização. O resultado final será uma tabela de amortização formatada e impressa no console com informações sobre cada parcela ao longo do tempo, seguindo o método Price.

Figura 12 – Código para criar tabela de amortização Price em Python.

```
1 class Tabela_Price:
2     def __init__(self, principal, taxa, parcelas):
3         self.principal = principal
4         self.taxa = taxa / 100.0  # Convertendo a taxa para decimal
5         self.parcelas = parcelas
6
7     def gerar_tabela(self):
8         dash = '-' * 68
9         parcela = (self.principal * (1 + self.taxa) ** (self.parcelas) * self.taxa) / ((1 + self.taxa) ** (self.parcelas) - 1)
10        saldo_devedor = self.principal
11
12        print(dash)
13        print('{:<10}{:<15}{:<15}{:<15}{:<20}'.format('Parcela', 'Juros', 'Amortização', 'Parcela', 'Saldo Devedor'))
14        print(dash)
15
16        # Adiciona a linha inicial com valores zero
17        print('{:<10}{:<15.2f}{:<15.2f}{:<15.2f}{:<20.2f}'.format(0, 0.0, 0.0, 0.0, self.principal))
18
19        for periodo in range(1, self.parcelas + 1):
20            juros = saldo_devedor * self.taxa
21            amortizacao = parcela - juros
22            saldo_devedor -= amortizacao
23
24            print('{:<10}{:<15.2f}{:<15.2f}{:<15.2f}{:<20.2f}'.format(periodo, juros, amortizacao, parcela, saldo_devedor))
25
26        print(dash)
27
28 # Exemplo de uso da classe
29 price_tabela = Tabela_Price(principal=20000, taxa=1.7, parcelas=12)
30 price_tabela.gerar_tabela()
```

Fonte: o autor (2024)

Os valores usados são os mesmos do exemplo 3.3. Para compilar o código usamos o Google Colab. A Figura 13 mostra essa tabela, gerada com o Google Colab.

Figura 13 – Tabela de amortização Price em Python.

Parcela	Juros	Amortização	Parcela	Saldo Devedor
0	0.00	0.00	0.00	20000.00
1	340.00	1516.52	1856.52	18483.48
2	314.22	1542.30	1856.52	16941.18
3	288.00	1568.52	1856.52	15372.66
4	261.34	1595.19	1856.52	13777.47
5	234.22	1622.30	1856.52	12155.17
6	206.64	1649.88	1856.52	10505.28
7	178.59	1677.93	1856.52	8827.35
8	150.06	1706.46	1856.52	7120.90
9	121.06	1735.47	1856.52	5385.43
10	91.55	1764.97	1856.52	3620.46
11	61.55	1794.97	1856.52	1825.49
12	31.03	1825.49	1856.52	-0.00

Fonte: o autor (2024)

4.4 *Sistema Price pós-fixado em Linguagem Python*

Na classe "Tabela_Price_pos" (Figura 14), criada para imprimir uma tabela do sistema Price pós-fixado, utilizamos o comando da linha 10 para receber e armazenar a taxa de correção.

Dentro do loop *for*, calculam-se o saldo corrigido e os juros usando-se a taxa de correção, e o valor da parcela é calculado de acordo com a Equação 19.

Figura 14 – Código para criar tabela de amortização Price pós-fixado em Python.

```
1 class Tabela_Price_pos:
2     def __init__(self, principal, taxa, parcelas):
3         self.principal = principal
4         self.taxa = taxa / 100.0  # Convertendo a taxa para decimal
5         self.parcelas = parcelas
6
7     def gerar_tabela(self):
8         dash = '-' * 90
9         saldo_devedor = self.principal
10        tr = float(input("Qual é a taxa de correção?     "))
11        saldo_corrigido = saldo_devedor * (1 + tr / 100)
12
13        print(dash)
14        print('{:<10}{:<15}{:<15}{:<15}{:<20}{:<20}'.format('Parcela', 'Juros', 'Amortização', 'Parcela', 'Saldo Devedor', 'Saldo Corrigido'))
15        print(dash)
16
17        # Adiciona a linha inicial com valores zero
18        print('{:<10}{:<15.2f}{:<15.2f}{:<15.2f}{:<20.2f}{:<20.2f}'.format(0, 0.0, 0.0, 0.0, self.principal, self.principal * (1 + tr / 100)))
19
20        for periodo in range(1, self.parcelas + 1):
21            saldo_corrigido = saldo_devedor * (1 + tr / 100)
22            juros = ((saldo_corrigido * self.taxa))
23            parcela = (saldo_corrigido * self.taxa * (1 + self.taxa) ** (-periodo + self.parcelas + 1)) / (
24                        (1 + self.taxa) ** (-periodo + self.parcelas + 1) - 1)
25            amortizacao = parcela - juros
26            saldo_devedor = saldo_corrigido - amortizacao
27
28            print('{:<10}{:<15.2f}{:<15.2f}{:<15.2f}{:<20.2f}{:<20.2f}'.format(periodo, juros, amortizacao, parcela, \
29                                                                                saldo_devedor, saldo_devedor * (1 + tr / 100)))
30
31        print(dash)
32
33 # Exemplo de uso da classe
34 price_tabela_pos = Tabela_Price_pos(principal=20000, taxa=1.7, parcelas=12)
35 price_tabela_pos.gerar_tabela()
```

Fonte: o autor (2024)

Esse código gera e imprime a tabela que pode ser vista na Figura 15, que contém informações a respeito das parcelas de um empréstimo ou financiamento com sistema Price pós-fixado. Os valores são os mesmos do exemplo 3.4. Utilizamos, para isso, o Google Colab.

Figura 15 – Tabela de amortização Price pós-fixado em Python.

Parcela	Juros	Amortização	Parcela	Saldo Devedor	Saldo Corrigido
0	0.00	0.00	0.00	20000.00	20200.00
1	343.40	1531.69	1875.09	18668.31	18855.00
2	320.53	1573.30	1893.84	17281.69	17454.51
3	296.73	1616.05	1912.78	15838.46	15996.85
4	271.95	1659.96	1931.90	14336.89	14480.26
5	246.16	1705.06	1951.22	12775.20	12902.95
6	219.35	1751.38	1970.73	11151.57	11263.09
7	191.47	1798.97	1990.44	9464.12	9558.76
8	162.50	1847.85	2010.35	7710.91	7788.02
9	132.40	1898.05	2030.45	5889.97	5948.86
10	101.13	1949.62	2050.75	3999.24	4039.23
11	68.67	2002.59	2071.26	2036.64	2057.01
12	34.97	2057.01	2091.97	-0.00	-0.00

Fonte: o autor (2024)

4.5 Sistema SACRE em Linguagem Python

Para criar uma tabela do sistema de amortização SACRE em Python, criamos a classe "Tabela_SACRE" (Figura 16), na qual usamos o loop *for* três vezes, visto que, no sistema SACRE, o valor da parcela é recalculado a cada ano.

Figura 16 – Tabela de amortização SACRE em Python.

```python
class Tabela_SACRE:
    def __init__(self, principal, taxa, parcelas):
        self.principal = principal
        self.taxa = taxa / 100.0  # Convertendo a taxa para decimal
        self.parcelas = parcelas

    def gerar_tabela(self):
        dash = '-' * 42
        saldo_devedor = self.principal

        print(dash)
        print('{:<10}{:<15}{:<15}{:<15}{:<20}'.format('Período', 'Juros', 'Amortização', 'Parcela', 'Saldo Devedor'))
        print(dash)

        for periodo in range(1, 13):
            juros = (saldo_devedor * self.taxa)
            parcela = (self.principal / self.parcelas) + (self.principal * self.taxa)
            amortizacao = parcela - juros
            saldo_devedor -= amortizacao
            print('{:<2}{:>15.2f}{:>15.2f}{:>15.2f}{:>20.2f}'.format(periodo, juros, amortizacao, parcela, saldo_devedor))

        for periodo in range(13, 25):
            juros = (saldo_devedor * self.taxa)
            parcela = (11569.82 / (self.parcelas - 12)) + (11569.82 * self.taxa)
            amortizacao = parcela - juros
            saldo_devedor -= amortizacao
            print('{:<2}{:>15.2f}{:>15.2f}{:>15.2f}{:>20.2f}'.format(periodo, juros, amortizacao, parcela, saldo_devedor))

        for periodo in range(25, self.parcelas + 1):
            juros = (saldo_devedor * self.taxa)
            parcela = (5370.15 / (self.parcelas - 24)) + (5370.15 * self.taxa)
            amortizacao = parcela - juros
            saldo_devedor -= amortizacao
            print('{:<2}{:>15.2f}{:>15.2f}{:>15.2f}{:>20.2f}'.format(periodo, juros, amortizacao, parcela, saldo_devedor))

        print(dash)

# Exemplo de uso da classe
sacre_tabela = Tabela_SACRE(principal=18000, taxa=1.25, parcelas=36)
sacre_tabela.gerar_tabela()
```

Fonte: o autor (2024)

Esse código gera e imprime a tabela que vemos na Figura 17, criada no Google Colab, contendo informações sobre as parcelas do financiamento, o mesmo do exemplo 3.5.

Figura 17 – Tabela de amortização SACRE em Python.

Período	Juros	Amortização	Parcela	Saldo Devedor
1	225.00	500.00	725.00	17500.00
2	218.75	506.25	725.00	16993.75
3	212.42	512.58	725.00	16481.17
4	206.01	518.99	725.00	15962.19
5	199.53	525.47	725.00	15436.71
6	192.96	532.04	725.00	14904.67
7	186.31	538.69	725.00	14365.98
8	179.57	545.43	725.00	13820.56
9	172.76	552.24	725.00	13268.31
10	165.85	559.15	725.00	12709.17
11	158.86	566.14	725.00	12143.03
12	151.79	573.21	725.00	11569.82
13	144.62	482.08	626.70	11087.74
14	138.60	488.10	626.70	10599.64
15	132.50	494.20	626.70	10105.44
16	126.32	500.38	626.70	9605.06
17	120.06	506.64	626.70	9098.42
18	113.73	512.97	626.70	8585.45
19	107.32	519.38	626.70	8066.07
20	100.83	525.87	626.70	7540.20
21	94.25	532.45	626.70	7007.76
22	87.60	539.10	626.70	6468.65
23	80.86	545.84	626.70	5922.81
24	74.04	552.66	626.70	5370.15
25	67.13	447.51	514.64	4922.64
26	61.53	453.11	514.64	4469.53
27	55.87	458.77	514.64	4010.76
28	50.13	464.50	514.64	3546.26
29	44.33	470.31	514.64	3075.94
30	38.45	476.19	514.64	2599.75
31	32.50	482.14	514.64	2117.61
32	26.47	488.17	514.64	1629.44
33	20.37	494.27	514.64	1135.17
34	14.19	500.45	514.64	634.72
35	7.93	506.71	514.64	128.02
36	1.60	513.04	514.64	-385.02

Fonte: o autor (2024)

5 Propostas de Atividades: Simulações com Linguagem Python

Neste capítulo apresentaremos duas sequências didáticas, por meio de propostas de atividades que podem ser usadas por professores de Matemática após a explanação em sala de aula dos conteúdos de Matemática Financeira e Sistemas de Amortização.

Levando em conta que estudantes do Ensino Médio estão prestes a ingressar na vida adulta, é importante que possam conhecer situações financeiras com as quais se depararão. Dentre essas operações estão empréstimos e financiamentos, cujos valores dos pagamentos são determinados pelo sistema de amortização utilizado.

> Nessa perspectiva, após se formarem no Ensino Médio, não poucos discentes lidarão com a aquisição de bens e serviços, tais como veículos e imóveis. Alguns se depararão com situações de financiamentos de cursos de graduação e outros cursos de formação profissional, ou ainda, empréstimos para a quitação de faturas/boletos. Diante disso, esse público carece de estratégias que o auxiliem na decisão sobre a aquisição desses e outros itens, de forma consciente. (Freitas; Ferreira; Moreira, 2021, p. 152)

As sequências didáticas priorizam análises das características dos sistemas SAC e Price, por serem os dois sistemas de amortização mais usados no Brasil. A depender da instituição financeira, algumas operações financeiras podem ter determinado qual sistema de amortização será adotado, enquanto outras podem oferecer a possibilidade de escolha por parte do cliente. Aqui, estamos considerando essa possibilidade de escolha.

5.1 Sequências didáticas

A primeira proposta é que, após a explanação do conteúdo de Matemática Financeira e Sistemas de Amortização em sala de aula, o professor leve os estudantes ao laboratório de informática, a fim de que usem linguagem de programação Python para construir tabelas e gráficos que simulam um financiamento de uma motocicleta usando os sistemas de amortização SAC e Price. O estudante deve ser instruído a efetuar cálculos adicionais e escolher a melhor opção diante das situações apresentadas.

A escolha por simular o financiamento de uma motocicleta se deve ao fato de que os estudantes, principalmente os oriundos de escola pública, adquirem a motocicleta como primeiro veículo. Assim, temos um tema significativo para os estudantes, estando alinhado aos interesses de muitos deles, o que é muito importante no processo de ensino-aprendizagem. Como dito por Paulo Freire, “quanto mais assumam os homens uma postura ativa na investigação de sua temática, tanto mais aprofundam a sua tomada de consciência em torno da realidade e, explicitando sua temática significativa, se apropriam dela” (Freire, 1987, p. 56).

Proposta 5.1.1 *Suponha que uma motocicleta custa R\$* $18.000,00$*. Uma pessoa está disposta a pagar uma entrada de R\$* $3.000,00$ *e financiar o restante em* 18 *parcelas, com taxa de* $2,5\%$ *ao mês, podendo escolher entre o sistema SAC ou o sistema Price. Qual sistema de amortização é mais vantajoso?*

Por meio de uma classe criada em linguagem Python, podemos realizar uma simulação que exibe, por meio de uma tabela, o descrição das parcelas a serem pagas por esse financiamento. A Figura 18 mostra um trecho da classe em linguagem Python que gera a tabela que descreve as parcelas do financiamento no sistema SAC. Na *linha 9* vemos a definição do valor da amortização como a razão entre o valor principal e a quantidade de parcelas, conforme a Equação 5.

Figura 18 – Trecho da classe em Python para gerar a tabela que descreve as parcelas do financiamento no sistema SAC.

```
1 class Tabela_SAC:
2     def __init__(self, principal, taxa, parcelas):
3         self.principal = principal
4         self.taxa = taxa / 100.0  # Convertendo a taxa para decimal
5         self.parcelas = parcelas
6
7     def gerar_tabela(self):
8         dash = '-' * 68
9         amortizacao = self.principal / self.parcelas
10        saldo_devedor = self.principal
```

Fonte: o autor (2024)

A Figura 19 mostra a simulação das parcelas com sistema SAC.

Ao observar a evolução das parcelas com sistema SAC, pode-se notar que estas decrescem em progressão aritmética. Daí, o professor pode instruir os estudantes a calcularem a soma de todas as parcelas, resultando em R\$ $18.562,50$. Adicionando os R\$ $3.000,00$ pagos na entrada, conclui-se que o total a ser pago por esse financiamento é de R\$ $21.562,50$

Figura 19 – Simulação com sistema SAC gerada com linguagem Python.

Parcela	Juros	Amortização	Parcela	Saldo Devedor
0	0.00	0.00	0.00	15000.00
1	375.00	833.33	1208.33	14166.67
2	354.17	833.33	1187.50	13333.33
3	333.33	833.33	1166.67	12500.00
4	312.50	833.33	1145.83	11666.67
5	291.67	833.33	1125.00	10833.33
6	270.83	833.33	1104.17	10000.00
7	250.00	833.33	1083.33	9166.67
8	229.17	833.33	1062.50	8333.33
9	208.33	833.33	1041.67	7500.00
10	187.50	833.33	1020.83	6666.67
11	166.67	833.33	1000.00	5833.33
12	145.83	833.33	979.17	5000.00
13	125.00	833.33	958.33	4166.67
14	104.17	833.33	937.50	3333.33
15	83.33	833.33	916.67	2500.00
16	62.50	833.33	895.83	1666.67
17	41.67	833.33	875.00	833.33
18	20.83	833.33	854.17	-0.00

Fonte: o autor (2024)

A Figura 20 mostra um trecho da classe em linguagem Python que gera a tabela que descreve as parcelas do financiamento no sistema Price. Na *linha 9* vemos a definição do valor da parcela sendo calculado conforme a Equação 15.

Figura 20 – Trecho da classe em Python para gerar a tabela que descreve as parcelas do financiamento no sistema Price.

```
7   def gerar_tabela(self):
8       dash = '-' * 68
9       parcela = (self.principal * (1 + self.taxa) ** (self.parcelas) * self.taxa) / ((1 + self.taxa) ** (self.parcelas) - 1)
10      saldo_devedor = self.principal
```

Fonte: o autor (2024)

A Figura 21 mostra a simulação das parcelas com sistema Price.

Vê-se que todas as parcelas são iguais a R\$ $1.045,05$, de modo que a soma de todas as parcelas é igual a R\$ $18.810,90$. Adicionando os R\$ $3.000,00$ pagos na entrada, conclui-se que o total a ser pago por esse financiamento é de R\$ $21.810,90$.

Ao realizar essas simulações, o estudante poderá fazer a tomada de decisão que mais o favorece financeiramente. O professor pode questionar qual a melhor opção para quem deseja escolher o menor valor total dos pagamentos, devendo ter como resposta a opção por sistema SAC. Por outro lado, se o questionamento for qual a melhor opção para quem deseja começar os pagamentos pagando menos por parcela, deve-se ter como resposta a opção por sistema Price.

O professor pode, ainda, apresentar a ferramenta gráfica *matplotlib* do Python, para que o estudante possa plotar gráficos que ilustram a evolução dos juros, das amortizações e das parcelas

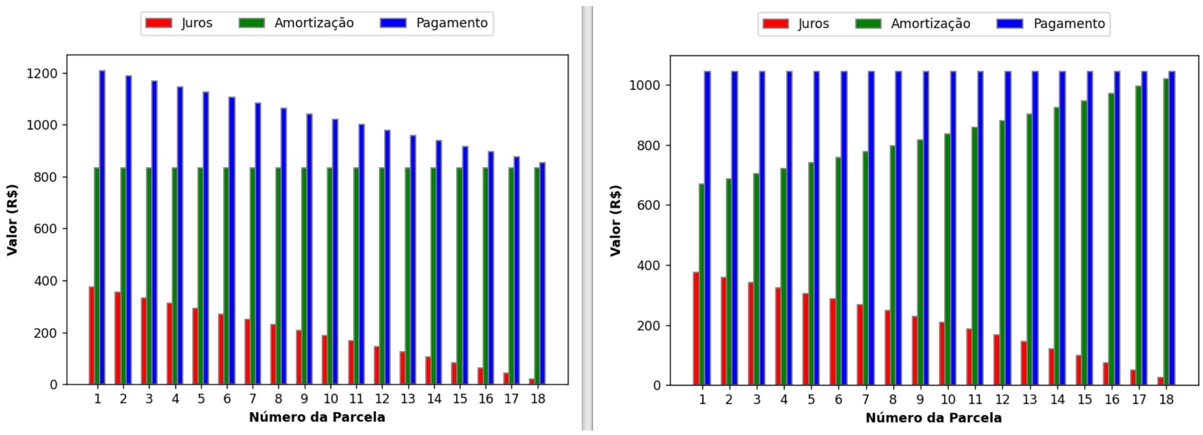
Figura 21 – Simulação com sistema Price gerada com linguagem Python.

Parcela	Juros	Amortização	Parcela	Saldo Devedor
0	0.00	0.00	0.00	15000.00
1	375.00	670.05	1045.05	14329.95
2	358.25	686.80	1045.05	13643.15
3	341.08	703.97	1045.05	12939.17
4	323.48	721.57	1045.05	12217.60
5	305.44	739.61	1045.05	11477.99
6	286.95	758.10	1045.05	10719.89
7	268.00	777.05	1045.05	9942.84
8	248.57	796.48	1045.05	9146.35
9	228.66	816.39	1045.05	8329.96
10	208.25	836.80	1045.05	7493.16
11	187.33	857.72	1045.05	6635.44
12	165.89	879.17	1045.05	5756.27
13	143.91	901.14	1045.05	4855.13
14	121.38	923.67	1045.05	3931.46
15	98.29	946.76	1045.05	2984.69
16	74.62	970.43	1045.05	2014.26
17	50.36	994.69	1045.05	1019.56
18	25.49	1019.56	1045.05	-0.00

Fonte: o autor (2024)

ao longo do tempo, como na Figura 22, verificando, no sistema SAC, o decrescimento das parcelas (azul), as amortizações constantes (verde) e o decrescimento dos juros (vermelho). Já no sistema Price, verificar as parcelas de valor constante (azul), o crescimento das amortizações (verde) e o decrescimento dos juros (vermelho). Com isso, os estudantes podem visualizar e comparar os dois sistemas de amortização graficamente.

Figura 22 – Representações gráficas em Python. SAC (esquerda) e Price (direita).



Fonte: o autor (2024)

A Figura 23 organiza as abordagens metodológicas que podem ser exploradas pelo professor de Matemática na Proposta 5.1.1.

A segunda proposta é fazer com que o estudante decida pela melhor opção numa contratação de empréstimo. O professor pode apresentar um problema que consista em dois

Figura 23 – Abordagens metodológicas da Proposta 5.1.1

Abordagem Metodológica	**Descrição**	**Resultado Esperado**
Simulação das Parcelas com SAC e Price	Simular a evolução das parcelas utilizando SAC e Price, observando as características de cada sistema.	Obter parcelas decrescentes com SAC e parcelas constantes de R$ $1.045,05$ em Price (Figuras 19 e 21).
Cálculo da Soma das Parcelas com SAC e Price	Calcular a soma de todas as parcelas no sistema SAC (usando progressão aritmética) e no sistema Price (multiplicando o valor da parcela pela quantidade de períodos) e adicionar o valor pago na entrada para obter o total do financiamento.	Com SAC: total das parcelas de R$ $18.562,50$; com a entrada: R$ $21.562,50$. Com Price: total das parcelas de R$ $18.810,90$; com a entrada: R$ $21.810,90$.
Análise Comparativa das Parcelas e Tomada de Decisão	Comparar qual sistema de amortização resulta no menor valor total e em parcelas iniciais menores e decidir qual sistema é mais vantajoso financeiramente.	Escolher SAC para ter o menor valor total dos pagamentos ou escolher o sistema Price para ter parcelas iniciais de menor valor.
Uso do Matplotlib, uma biblioteca do Python com vários recursos gráficos, para Visualização Gráfica das Características de cada Sistema de Amortização	Plotar gráficos que permitem observar graficamente a evolução dos juros, amortizações e parcelas em cada sistema de amortização ao longo do tempo.	Perceber, conforme a Figura 22, no sistema SAC, o decrescimento das parcelas (azul), as amortizações constantes (verde) e o decrescimento dos juros (vermelho). Já no sistema Price, verificar as parcelas de valor constante (azul), o crescimento das amortizações (verde) e o decrescimento dos juros (vermelho).

Fonte: o autor (2024)

bancos oferecendo empréstimos, um com sistema SAC e o outro com sistema Price, instruindo os estudantes a fazerem a melhor escolha diante das opções apresentadas. Aqui, o professor também pode instruir os estudantes a fazerem alterações nos parâmetros, verificando com quais valores uma proposta, antes menos vantajosa, passaria a ser equivalente ou mais vantajosa do que a outra.

Proposta 5.1.2 *Suponha que uma pessoa precise contratar um empréstimo de R$* $10.000,00$*, pretendendo pagar em 12 parcelas, e avalia duas propostas, uma do Banco A e outra do Banco B. Considerando que o Banco A oferece o empréstimo com sistema SAC e o Banco B oferece* *com sistema Price, ambos com taxa de juros de* $5,5\%$ *ao mês. Em qual banco essa pessoa deve tomar o empréstimo?*

O professor pode questionar aos estudantes em qual banco essa pessoa pagaria o menor valor total pelo empréstimo. Ao comparar os resultados obtidos nas tabelas geradas em Python (Figura 24), o estudante deve responder que a melhor opção é ofertada pelo Banco A, pois as parcelas resultam no total de R\$ $13.575,00$, enquanto no Banco B o total é de R\$ $13.923,48$.

Figura 24 – Simulação das parcelas do empréstimo com sistema SAC (esquerda) e Price (direita) gerada com linguagem Python.

Parcela	Juros	Amortização	Parcela	Saldo Devedor
0	0.00	0.00	0.00	10000.00
1	550.00	833.33	1383.33	9166.67
2	504.17	833.33	1337.50	8333.33
3	458.33	833.33	1291.67	7500.00
4	412.50	833.33	1245.83	6666.67
5	366.67	833.33	1200.00	5833.33
6	320.83	833.33	1154.17	5000.00
7	275.00	833.33	1108.33	4166.67
8	229.17	833.33	1062.50	3333.33
9	183.33	833.33	1016.67	2500.00
10	137.50	833.33	970.83	1666.67
11	91.67	833.33	925.00	833.33
12	45.83	833.33	879.17	-0.00

Parcela	Juros	Amortização	Parcela	Saldo Devedor
0	0.00	0.00	0.00	10000.00
1	550.00	610.29	1160.29	9389.71
2	516.43	643.86	1160.29	8745.85
3	481.02	679.27	1160.29	8066.58
4	443.66	716.63	1160.29	7349.95
5	404.25	756.05	1160.29	6593.90
6	362.66	797.63	1160.29	5796.28
7	318.80	841.50	1160.29	4954.78
8	272.51	887.78	1160.29	4067.00
9	223.68	936.61	1160.29	3130.39
10	172.17	988.12	1160.29	2142.27
11	117.82	1042.47	1160.29	1099.80
12	60.49	1099.80	1160.29	-0.00

Fonte: o autor (2024)

O professor pode questionar qual deveria ser a taxa mensal cobrada pelo Banco B para que as duas propostas gerassem o mesmo total na soma das parcelas. Possivelmente o estudante descobrirá isso alterando o parâmetro da taxa no código Python até verificar que a taxa mensal deveria ser de $5,047\%$, sendo a parcela de R\$ $1.131,25$.

Ao realizar essas simulações, o estudante deverá ser capaz de compreender que o conhecimento de Matemática Financeira, aliado ao pensamento computacional, por meio da linguagem de programação, podem torná-lo capaz de usar recursos tecnológicos para resolver problemas de natureza financeira com os quais se depararão durante a vida adulta. A importância dessa abordagem está expressa na BNCC, em competências que versam sobre a implementação da computação na Educação Básica, tais como “Compreender as possibilidades e os limites da Computação para resolver problemas, tanto em termos de viabilidade quanto de eficiência, propondo e analisando soluções computacionais para diversos domínios do conhecimento, considerando diferentes aspectos.” (Brasil, 2022, p. 61).

6 Considerações Finais

Este trabalho apresentou uma análise detalhada dos sistemas de amortização de empréstimos: Sistema de Amortização Constante (SAC), Sistema Price e Sistema de Amortização Crescente (SACRE), com ênfase na aplicação da linguagem de programação Python para realizar, com estudantes do Ensino Médio, simulações e visualizações de dois desses sistemas (SAC e Price) no laboratório de informática da escola.

A utilização de Python é fundamental para a realização das simulações, pois permite uma visualização clara e dinâmica dos impactos de cada sistema de amortização. Com Python, os estudantes podem criar tabelas que ilustraram a variação das parcelas ao longo do tempo, bem como o valor total pago ao final do período de amortização. Isso facilita a compreensão das diferenças entre os sistemas, destacando as vantagens e desvantagens de cada um e munindo os estudantes de informações úteis para boas tomadas de decisões financeiras no futuro.

Através das simulações, os estudantes podem observar que o sistema SAC, apesar de iniciar com parcelas mais altas, resulta em um valor total pago menor em comparação ao sistema Price, que mantém parcelas constantes, mas com um custo total mais elevado.

O uso de Python não só auxilia na análise dos sistemas de amortização, mas também demonstra a importância de integrar conhecimentos de programação com matemática financeira. A capacidade de automatizar cálculos complexos e gerar visualizações intuitivas torna o aprendizado mais eficiente e aplicável a situações reais.

Portanto, este trabalho contribui para a educação financeira, mostrando que a aplicação de ferramentas tecnológicas, como a programação em Python, é essencial para a formação de estudantes preparados para enfrentar desafios financeiros no futuro. A integração de Python no estudo de matemática financeira permite um entendimento mais profundo e uma capacidade analítica aprimorada, proporcionando aos estudantes uma ferramenta poderosa para a tomada de decisões financeiras informadas.

Usamos neste trabalho o paradigma de programação POO (Programação Orientada a Objetos), pois o foco foi o estudante manipular os parâmetros dados nas situações-problema para observar o comportamento dos valores nas tabelas geradas. Como trabalho futuro, o código poderá ser refatorado para o paradigma imperativo para facilitar o entendimento do estudante e permitir que ele dê instruções diretas à máquina.

A implementação de Python no ensino médio pode enfrentar alguns desafios, tais como infraestrutura inadequada em muitas escolas públicas, falta de formação específica dos professores e a pouco conhecimento dos alunos com manuseio de computadores e conceitos de programação.

Existem outras possibilidades de continuação e estudos a partir das equações que descrevem os sistemas de financiamento, podendo até levar ao desenvolvimento de um aplicativo gratuito nas áreas STEM (Science, Technology, Engineering and Mathematics), como educação e ciência. (Unbehaum; Gava; Artes, 2023)

Referências

ABRAMOVAY, M. *et al.*. *Ensino médio*: múltiplas vozes. Brasília: UNESCO, 2003.

ANDRADE, P. R. d. S. *Matemática financeira*: trabalhando sistema de amortização no ensino médio. 2015. Dissertação (Mestrado Profissional em Matemática) — Universidade Federal do Tocantins, Palmas, 2015.

BATISTA JÚNIOR, R. I. *Matemática financeira contextualizada em sistemas de amortização e impostos de renda.* 2014. Dissertação (Mestrado Profissional em Matemática) — Universidade Estadual Paulista (Unesp), Rio Claro, 2014.

BRASIL. *Base Nacional Comum Curricular*. Brasília: Ministério da Educação, 2018.

BRASIL. *Computação na Educação Básica - Complemento à BNCC*. Brasília: Ministério da Educação, 2022.

COSTA, A. C. *et al.*. Python: Será que é possível numa escola pública de Ensino Médio? *In:* WORKSHOP DE INFORMÁTICA NA ESCOLA, 23., 2017, Porto Alegre. **Anais** [...], Porto Alegre: Sociedade Brasileira de Computação, 2017, p. 255–264.

FREIRE, P. *Pedagogia do oprimido*. 17. ed. Rio de Janeiro: Paz e Terra, 1987.

FREITAS, B.; FERREIRA, F. A.; MOREIRA, V. G. Empréstimos & financiamentos: uma revisão sistemática sobre o ensino de sistemas de amortização. *Revista Internacional de Pesquisa em Educação Matemática*, Sociedade Brasileira de Educação Matemática, v. 11, n. 3, p. 151–172, 2021.

GRAVE, L. A. S. *O pensamento computacional na prática*: uma experiência usando python em aulas de matemática básica. 2021. Dissertação (Mestrado Profissional em Matemática) — Universidade Federal de Santa Maria, Santa Maria, 2021.

PESENTE, G. M.; MATOS, E. A. S. Á. de; AVELINO, A. O ensino de matemática por meio da linguagem de programação python. *Revista Contemporânea*, v. 3, n. 3, p. 1239–1256, 2023.

SANTOS, M. J. F. *Sistemas de amortização na Educação básica.* 2015. Dissertação (Mestrado Profissional em Matemática) — Universidade Federal de Alagoas, Maceió, 2015.

SOUSA, B. O. d. *Matemática financeira e os sistemas de financiamentos no cotidiano. 2016*. Dissertação (Mestrado Profissional em Matemática) — Universidade Federal do Piauí, Teresina, 2016.

UNBEHAUM, S.; GAVA, T. M.; ARTES, A. *Panorama de educação STEM no Brasil [livreto eletrônico]*: reflexões sobre a análise de dados e documentação bibliográfica. São Paulo: British Council Brasil, 2023.

VENDITE, L. L. *Matemática Financeira e a Utilização de Planilhas Eletrônicas*. Rio de Janeiro: Ciência Moderna, 2017.