



**UNIVERSIDADE FEDERAL DO RECÔNCAVO DA BAHIA - UFRB
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS - CETEC
MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL
– PROFMAT**

FERNANDO LUIZ DE ALMEIDA ARAÚJO

Explorando a Matemática Financeira e o Uso do Python no Ensino Médio

CRUZ DAS ALMAS - BA

2023

FERNANDO LUIZ DE ALMEIDA ARAÚJO

Explorando a Matemática Financeira e o Uso do Python no Ensino Médio

Dissertação de Mestrado apresentada à
Comissão Acadêmica Institucional do
PROFMAT-UFRB como requisito parcial
para obtenção do título de Mestre em
Matemática.

Orientador: Prof. Dr. Anderson Reis da
Cruz.

CRUZ DAS ALMAS - BA

2023

FERNANDO LUIZ DE ALMEIDA ARAÚJO

Explorando a Matemática Financeira e o Uso do Python no Ensino
Médio

Dissertação apresentada ao Programa de Mestrado Profissional em Matemática em Rede Nacional — PROFMAT, do Centro de Ciências Exatas e Tecnológicas da Universidade Federal do Recôncavo da Bahia — UFRB, como requisito parcial para obtenção do grau de Mestre em Matemática.

Aprovado em: 20 / 07 / 2023

BANCA EXAMINADORA

Prof.Dr. Anderson Reis da Cruz (Orientador)
UFRB - CETEC.

Prof. Dr. Luiz Alberto de Oliveira Silva
UFRB - CETEC.

Prof. Dr. Marcus Vinicius da Conceição Morro
IFRO.

FICHA CATALOGRÁFICA

A663e

Araújo, Fernando Luiz de Almeida.

Explorando a matemática financeira e o uso do Python no Ensino Médio / Fernando Luiz de Almeida Araújo._ Cruz das Almas, BA, 2023.

113f.; il.

Dissertação (Mestrado) – Universidade Federal do Recôncavo da Bahia, Centro de Ciências Exatas e Tecnológicas, Mestrado Profissional em Matemática – PROFMAT.

Orientador: Prof. Dr. Anderson Reis da Cruz.

1. Matemática financeira – Estudo e ensino. 2. Matemática financeira – Python (Linguagem de programação de computador). 3. Ensino médio – Análise. I. Universidade Federal do Recôncavo da Bahia, Ciências Exatas e Tecnológicas. II. Título.

CDD: 513.93

DEDICATÓRIA

Dedico esta conquista aos meus pais, aos meus irmãos, a minha esposa e aos meus filhos

“Astronarta libertado
Minha vida me urtrapassa
Em quarqué rota que eu faça.”

(Dois mil e um - Tom Zé)

AGRADECIMENTOS

Agradeço primeiramente a Deus que a cada dia tem se revelado a mim de forma incrível e me mantêm sob seus cuidados a todo instante. Aos meus pais Osvaldo Cardoso de Araújo (In Memoriam) e Arlinda de Almeida Araújo e ao meu segundo pai Antonio de Lima Motta , que moldaram meu caráter e sempre me apoiaram nas decisões importantes da minha caminhada, aconselhando-me e acompanhando meus passos. À minha esposa Bartira e aos meus filhos Fernando, Pedro e Maria que me apoiaram e souberam compreender a minha ausência, até mesmo em momentos especiais.

A Universidade Federal do Recôncavo da Bahia (UFRB), ao Centro de Ciências Exatas e Tecnológicas (CETEC), todos os servidores, professores e alunos, por possibilitara realização do curso;

Aos meus amigos do PROFMAT da turma de 2021, em especial, ao Prof. Me. Paulo Cesar, Prof. Ricardo Pinto e ao Prof. Sidney dos Anjos, pelo apoio e incentivo nos momentos difíceis e pelo tempo que passamos juntos estudando durante todo o curso e sempre com o mesmo lema: “estudar, aprender e ensinar ao próximo o que aprendeu”.

Ao meu amigo Prof. Dr. Reginaldo Pereira dos Santos Junior, que sempre acreditou em mim e que se disponibilizou para fazer leitura e contribuições desta dissertação e terá sempre a minha gratidão.

Agradeço aos professores da banca pela aceitação em fazer parte da mesma e participar dessa etapa tão importante para mim.

E de forma singular, agradeço ao meu orientador Prof. Dr. Anderson Reis da Cruz pelo acolhimento em me orientar, pelas sugestões ao longo do texto, pelos direcionamentos nos passos a seguir a cada etapa, pelas revisões e pela compreensão durante a pesquisa e escrita, pela boa convivência ao longo dos meses que estivemos juntos e por sua simpatia. Muito obrigado professor.

Por fim, a todos que, direta ou indiretamente, contribuíram para a realização e a conclusão deste sonho, o meu MUITO OBRIGADO!!!

A felicidade de mais esta conquista é um reflexo de toda a dedicação, esforço, incentivo e amor que vocês sempre manifestaram por mim.

RESUMO

Este trabalho tem como objetivo principal conscientizar os professores sobre a importância de ensinar matemática financeira de forma relevante para a realidade dos alunos. Além disso, busca incentivar a adoção de novas tecnologias no ensino dessa disciplina e fornecer material teórico e prático para auxiliar os professores nessa tarefa.

Para atingir esses objetivos, o trabalho propõe uma abordagem inovadora para o ensino da matemática financeira no Ensino Médio, utilizando a integração do Python e do ambiente online Google Colab. Através dessa abordagem, busca-se estimular o interesse dos alunos, promover uma compreensão mais profunda dos conceitos financeiros e desenvolver habilidades de resolução de problemas. Dessa forma, os estudantes estarão preparados para tomar decisões financeiras informadas em suas vidas pessoais e profissionais.

Ao explorar essa abordagem inovadora, o trabalho visa preencher uma lacuna no ensino da matemática financeira, proporcionando aos professores uma estratégia eficaz para engajar os alunos e tornar o conteúdo mais relevante para suas experiências e necessidades. A integração do Python e do ambiente Google Colab permite a aplicação prática dos conceitos financeiros, utilizando tecnologias modernas que são cada vez mais relevantes no mundo atual.

Palavras-chave: Linguagem de Programação Python; Matemática Financeira; Pensamento Computacional; Aprendizagem Significativa; Ensino Médio.

ABSTRACT

This work aims to raise awareness among teachers about the importance of teaching financial mathematics in a relevant manner, considering the reality of the students. Furthermore, it encourages the adoption of new technologies in the teaching of this subject and provides theoretical and practical materials to assist teachers in this task.

To achieve these objectives, the work proposes an innovative approach to teaching financial mathematics in high school, utilizing the integration of Python and the online platform Google Colab. Through this approach, the goal is to stimulate students' interest, promote a deeper understanding of financial concepts, and develop problem-solving skills. In this way, students will be prepared to make informed financial decisions in their personal and professional lives.

By exploring this innovative approach, the work aims to fill a gap in the teaching of financial mathematics, providing teachers with an effective strategy to engage students and make the content more relevant to their experiences and needs. The integration of Python and Google Colab enables the practical application of financial concepts, using modern technologies that are increasingly relevant in today's world.

Keywords: Python Programming Language; Financial Mathematics; Computational Thinking; Meaningful Learning; High School.

LISTA DE ILUSTRAÇÕES

LISTA DE FIGURAS

Figura 1	– Montante: juros simples.	18
Figura 2	– Comparação Juros simples x Juro composta	21
Figura 3	– Representação gráfica Fluxo de caixa	22
Figura 4	– Fluxo Caixa	39
Figura 5	– Fluxo Caixa	40
Figura 6	– Questão ENEM 2012	44
Figura 7	– Questão ENEM 2012	44
Figura 8	– Planejar para o Futuro	48
Figura 9	– Pilares do Pensamento Computacional	54
Figura 10	– Ambiente Google Colab	58
Figura 11	– Operadores Aritméticos	59
Figura 12	– Operadores Relacionais	60
Figura 13	– Cotação índice bovespa	71
Figura 14	– Diagrama de Fluxo de Caixa	90
Figura 15	– Diagrama de Fluxo de Caixa	91
Figura 16	– IBOV vs CDI	110

LISTA DE CÓDIGOS

LISTA DE TABELAS

Tabela 5	CALCULO DO I.R. NOS CDBs	38
Tabela 6	Resolução Questão ENEM/2012	44

SUMÁRIO

INTRODUÇÃO	12
1	MATEMÁTICA FINANCEIRA	15
1.1	Conceitos Básicos da Matemática Financeira	15
1.2	Sistemas de Capitalização	17
1.2.1	<i>Juros Simples</i>	17
1.2.2	<i>Juros Compostos</i>	19
1.3	Fluxo de Caixa	21
1.4	Séries de Capitais	25
1.5	Sistemas de Amortização de Empréstimos e Financiamentos	27
1.6	Sistema Financeiro	30
1.7	Investimentos	33
1.7.1	<i>Renda Fixa</i>	34
1.8	Renda Variável	41
2	O USO DE RECURSOS COMPUTACIONAIS PARA O ENSINO DE MATEMÁTICA	49
2.1	O Pensamento Computacional.	53
3	INTRODUÇÃO À PROGRAMAÇÃO COM PYTHON	57
3.1	Noções Básicas de Python	57
3.1.1	<i>Bibliotecas Científicas do Python</i>	68
4	PROPOSTAS DIDÁTICAS	72
4.1	Sequência Didática: Introdução ao Python, Matemática Financeira com Python e Uso das Bibliotecas NumPy, Pandas e Matplotlib	74
5	CONSIDERAÇÕES FINAIS	78
ANEXO A	Exemplos e Exercícios Sugeridos para os Planos de Aula	79
ANEXO A.1	Primeiros cálculos financeiros com Python	79
ANEXO A.2	Cálculos financeiros com Numpy financial	88
ANEXO A.3	Fontes de Dados Financeiros: Explorando yfinance e Pandas para Análise e Dados	92
ANEXO A.4	Visualização de dados financeiros com Matplotlib	105

ANEXO A.4.1 Comparando o Desempenho: IBOV vs CDI 106

REFERÊNCIAS **113**

INTRODUÇÃO

A matemática financeira desempenha um papel fundamental no contexto das finanças pessoais e empresariais, fornecendo as ferramentas necessárias para tomar decisões informadas sobre investimentos, empréstimos e planejamento financeiro. No entanto, muitos estudantes enfrentam dificuldades em compreender e aplicar os conceitos e cálculos associados a essa área do conhecimento. Além disso, a falta de contextualização e a abordagem tradicional da matemática financeira podem limitar o engajamento e a compreensão dos alunos. Nesse contexto, o professor deve propor problemas que façam parte da realidade do aluno e incentivá-lo a “pensar, refletir, formar e expressar sua própria opinião, sem precisar abandonar os conhecimentos particulares da disciplina” (PAIVA, 2016)

Diante desse cenário, o presente trabalho propõe uma abordagem diferenciada para o ensino da matemática financeira no Ensino Médio, integrando o uso da linguagem de programação Python e o ambiente online Google Colab. O objetivo geral deste trabalho é explorar a aplicação da matemática financeira no ensino médio, utilizando a linguagem de programação Python como ferramenta para a compreensão dos conceitos financeiros e o desenvolvimento de habilidades de resolução de problemas.

Os objetivos específicos deste trabalho são os seguintes:

1. Conscientizar os professores da necessidade de ensinar a matemática financeira de forma contextualizada e relevante para a realidade dos alunos.
2. Incentivar a inserção de novas tecnologias, como o Python, no ensino da matemática financeira, visando aprimorar o processo de ensino e aprendizagem.
3. Fornecer material teórico e prático que auxilie os professores no ensino da matemática financeira, de modo que faça sentido no universo do estudante.

Essa escolha de tema se baseia na experiência adquirida ao longo de 25 anos em sala de aula e nos relatos de colegas de trabalho, que indicam que a matemática financeira é abordada de forma superficial e descontextualizada nas escolas, focando apenas na memorização e aplicação de fórmulas. Essa abordagem limitada não desperta o interesse dos alunos e não promove uma compreensão profunda dos conceitos matemáticos e financeiros. Portanto, é necessário buscar uma

nova abordagem que proporcione uma experiência mais significativa e envolvente para os estudantes.

A metodologia utilizada nesta dissertação de mestrado é baseada em pesquisa teórica, que envolveu um levantamento bibliográfico em livros, artigos científicos, revistas, dissertações e teses sobre matemática financeira, educação financeira, ensino de matemática e o uso de tecnologias aplicadas à área. Também foram consultadas as diretrizes da Base Nacional Comum Curricular (BNCC). A diversidade de referências teóricas disponíveis permitiu a comparação, revisão e síntese do que foi considerado mais relevante.

A escolha deste tema se justifica pela necessidade de promover uma abordagem inovadora e contextualizada para o ensino da matemática financeira no Ensino Médio. A integração do Python e da programação como ferramentas de ensino possibilita uma experiência prática e envolvente, estimulando o desenvolvimento de habilidades de resolução de problemas e a compreensão dos conceitos financeiros de forma mais significativa. Através de aulas mais atraentes e dinâmicas, é possível despertar o interesse dos alunos, tornando o aprendizado da matemática financeira mais acessível e relevante para suas vidas. A utilização do Python como linguagem de programação proporciona uma abordagem prática, permitindo que os estudantes apliquem os conceitos financeiros em situações reais, resolvendo problemas e realizando análises utilizando dados reais.

A motivação para essa abordagem surge da necessidade de superar as limitações da abordagem tradicional da matemática financeira, que muitas vezes não desperta o interesse dos alunos e não promove uma compreensão profunda dos conceitos. Ao integrar o Python ao ensino da matemática financeira, busca-se tornar o processo de aprendizagem mais significativo e engajador, conectando os conceitos financeiros com situações do cotidiano dos estudantes.

A escolha do Python como linguagem de programação se justifica pela sua popularidade, facilidade de aprendizado e ampla gama de bibliotecas especializadas em análise de dados e finanças. O ambiente online Google Colab oferece uma plataforma acessível e gratuita para os estudantes utilizarem o Python sem a necessidade de instalação de softwares adicionais.

Ao conscientizar os professores sobre a importância de ensinar a matemática financeira de forma contextualizada e relevante, este trabalho visa contribuir para a melhoria do processo de ensino e aprendizagem, capacitando os alunos com habilidades práticas e relevantes para sua vida financeira. O fornecimento de material teórico e prático auxiliará os professores no planejamento e na condução das aulas, permitindo que eles adaptem o currículo de matemática financeira de acordo

com as necessidades e interesses dos estudantes.

Neste contexto, a apresentação deste trabalho será feita da seguinte forma:

- Capítulo 1: Apresenta os principais conceitos da matemática financeira, são eles: juros simples, juros compostos, taxas de juros, séries uniformes e sistema de amortização, sistema financeiro e investimentos.
- Capítulo 2: Apresenta a integração de tecnologias digitais, computação e Matemática como uma poderosa ferramenta para o aprimoramento dos alunos no contexto do ensino médio. Neste capítulo, exploraremos de que maneira essas áreas se entrelaçam e como sua combinação pode enriquecer o processo de aprendizagem dos estudantes. Serão apresentados os pilares do pensamento computacional e como eles podem contribuir para o desenvolvimento de habilidades cognitivas essenciais na resolução de desafios complexos.
- Capítulo 3: Apresenta-se uma abrangente introdução à programação utilizando a linguagem Python. Os leitores serão guiados através dos conceitos básicos de programação, como variáveis, estruturas de controle e funções, com foco na sua aplicação no contexto da matemática financeira. Serão fornecidos exemplos práticos e exercícios para que os leitores possam desenvolver habilidades de programação e aplicá-las na resolução de problemas financeiros.
- Capítulo 4: Apresenta propostas didáticas que visam integrar os conceitos de matemática financeira e o uso do Python no ensino médio. Serão exploradas estratégias pedagógicas para envolver os alunos de forma prática e interativa, promovendo o aprendizado ativo e a aplicação dos conhecimentos adquiridos. São propostos problemas que envolvem conceitos relevantes do Ensino Médio, juntamente com sugestões de codificação e execução de programas para sua resolução. Além disso, é oferecida uma sequência de atividades que explora problemas adicionais de programação relacionados a coleta, análise, manipulação e visualização de dados financeiros.
- Capítulo 5: Veremos as considerações finais deste trabalho.

1 MATEMÁTICA FINANCEIRA

Neste capítulo, abordamos os principais conceitos da matemática financeira que julgamos importantes. Esta é considerada como um importante instrumento para desenvolver a Educação Financeira, uma vez que operar, analisar e interpretar cálculos financeiros e toda a matemática comercial são imprescindíveis à Educação Financeira de um indivíduo.

1.1 Conceitos Básicos da Matemática Financeira

Neste seção, exploramos os conceitos mais utilizados na matemática financeira que devem ser aplicados aos alunos do ensino médio.

Ao longo deste capítulo, apresentaremos conceitos e definições relacionados a matemática financeira, tomando como base as referências (SODRE, 2020), (BOYER; MERZBACH, 2019), (BRUNI, 2000), (NETO, 2000a), (POMPEO; HAZZAN, 2007), (FARIA, 1983) e (SAMANEZ, 2002).

Na abordagem escolhida neste trabalho, para apresentar a Matemática Financeira, será necessário antes explorar conceitos fundamentais, como a noção de porcentagem e taxa percentuais que são a base da Matemática Financeira.

É muito comum nos depararmos diariamente com situações que envolvam porcentagem, como no comércio, em bancos e em pesquisas. De maneira geral, podemos dizer que a porcentagem é o conceito mais básico que dá origem a questões de Matemática Financeira.

A porcentagem é uma maneira de representar uma proporção como uma fração de 100. Por exemplo, 25% é o mesmo que $25/100$ ou $0,25$.

Para calcularmos de forma mais rápida alguns acréscimos e decréscimos em relação ao valor de algo, podemos utilizar o fator de aumento e o fator de desconto, respectivamente. A seguir apresentaremos algumas situações significativas em que nos deparamos no dia-a-dia, as quais conduzem a importantes habilidades que o estudante no Ensino Médio deve adquirir.

Um valor A aumentado ou diminuído de uma porcentagem $i = k\%$, conforme respectivamente, $i > 0$ e $i < 0$, resulta em $A + iA = (1 + i)A$. O fator $1 + i$ é

dito fator de variação de A com taxa i .

Podemos generalizar situações de n aumentos e m descontos sucessivos. Sendo i_1, i_2, \dots, i_n , as taxas percentuais de aumento e j_1, j_2, \dots, j_m as taxas percentuais de desconto sobre um valor inicial V_0 , o valor final V_n é dado por

$$V_n = V_0 \cdot (1 + i) \cdot \dots \cdot (1 + i_n) \cdot (1 - j_1) \cdot (1 - j_2) \cdot \dots \cdot (1 - j_m)$$

no qual os fatores $(1 + i_q)$, com $1 \leq q \leq n$, representam situações de aumento e os fatores $(1 - j_p)$, com $1 \leq p \leq m$, representam situações de descontos.

Vamos destacar alguns termos presentes no estudo da Matemática Financeira:

O **Capital** é qualquer valor expresso em moeda e disponível em determinada época. Pelo princípio financeiro da evolução do capital no tempo, este envolve uma unidade monetária e uma referência temporal. O capital inicialmente empregado numa operação será denotado por PV (Valor Presente, Valor Atual, Capital Inicial ou Principal).

O **Juro** é a remuneração do capital emprestado, num certo período de tempo. O juro será denotado por J .

O **Montante** é a soma do capital com os juros relativos a um certo período de tempo. O montante será denotado por FV (Valor Futuro).

$$FV = PV + J$$

Taxa de Juros é a taxa de crescimento do capital, ou seja, é a razão entre o juros pagos ou recebidos no fim de um determinado período de tempo, e o capital inicialmente emprestado. A taxa de juros será denotada por i . A taxa de juros do tempo total da operação é dada por:

$$i = \frac{J}{PV}$$

A taxa de juros pode estar na forma **percentual (centesimal)** ou na forma **unitária** e possui sempre uma referência temporal.

Forma percentual: 8% ao ano; 12% ao mês; 143,5% ao semestre.

Forma unitária: 0,08 ao ano; 0,12 ao mês; 1,435 ao semestre.

Numa aplicação financeira podemos considerar dois tipos de taxa de juros: **Taxa bruta:** é obtida considerando o valor da aplicação e o valor do resgate bruto, sem levar em conta o desconto do imposto devido.

Taxa líquida: é obtida considerando o valor da aplicação e o valor do resgate líquido, levando em conta o desconto do imposto devido.

1.2 Sistemas de Capitalização

O regime de capitalização determina o modo pelo qual será calculado o juro de um capital aplicado a uma certa taxa, durante um período de tempo. Trataremos da capitalização simples e da capitalização composta.

1.2.1 Juros Simples

No regime de juros simples, os juros de cada intervalo são calculados sempre sobre o mesmo capital principal, uma vez que não existe capitalização de juros nesse regime. Conseqüentemente, o capital aumentará a uma taxa linear e a taxa de juros terá um comportamento linear em relação ao tempo, podendo ser visto como uma *função afim* ou como uma *PA*. Portanto, no regime de juros simples, o rendimento mensal é sempre calculado pelo produto da taxa pelo capital inicial, da seguinte forma:

Passada uma unidade de tempo $\rightarrow j = c \cdot i$.

Passadas duas unidades de tempo $\rightarrow j = c \cdot i + c \cdot i = 2 \cdot c \cdot i$.

Passadas três unidades de tempo $\rightarrow j = c \cdot i + c \cdot i + c \cdot i = 3 \cdot c \cdot i$.

\vdots

Passadas n unidades de tempo $\rightarrow j = c \cdot i + \dots + c \cdot i = c \cdot i \cdot n$

Como o montante é $M = C + J$, então passada n unidades de tempo, temos

$$M = c + c \cdot i \cdot n \Rightarrow M = C(1 + i \cdot n).$$

Usando um argumento indutivo, verifica-se que essa expressão é válida para qualquer n .

Exemplo 1.1 *Júlia contraiu de uma amiga um empréstimo de R\$ 1.000,00, em regime de juros simples, à taxa de 5% ao mês, sem prazo determinado para pagar. Vamos calcular os montantes mês a mês.*

Como a taxa é de 5% ao mês, temos que o juro mensal será

$$J = C \cdot i \Rightarrow J = 1000 \cdot 0,05 = 50.$$

Logo, após:

1 mês o montante a ser pago será: $M_1 = 1.000 + 50 = 1.050$.

2 meses o montante a ser pago será: $M_2 = 1.100$.

3 meses o montante a ser pago será: $M_3 = 1.150$.

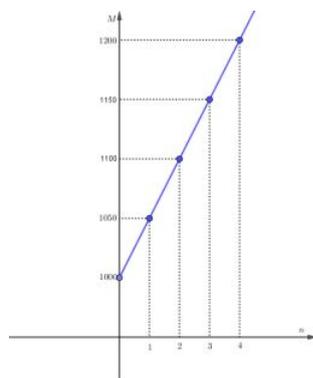
4 meses o montante a ser pago será: $M_4 = 1.200$.

⋮

n meses o montante será: $M_n = 1.000 \cdot (1 + 0,05 \cdot n)$ ou $M_n = 1.000 + 50 \cdot n$.

Podemos perceber que o montante pode ser visto como uma progressão aritmética de primeiro termo 1.000 e razão 50 ou como uma função afim $f(x) = ax + b$, onde $a = 50$ e $b = 1.000$. Podemos ver a representação gráfica do montante na Figura 1.

Figura 1 – Montante:
juros simples.



Fonte: Elaborada pelo autor

Assim, se $M = C(1 + ni)$ então

$$J = M - C = Cni,$$

sendo n o número de períodos de capitalização, tal que esse período se refere à taxa. Portanto, se a taxa é mensal, n deve ser o número de meses do processo de capitalização; se a taxa é anual, n deve ser expresso em anos e assim por diante.

Os juros simples têm aplicações práticas restritas. Seu uso limita-se principalmente às operações de curto prazo, sendo utilizados, basicamente, para o cálculo de valores monetários. Um exemplo onde isso ocorre é na cobrança de juros de mora por atraso de pagamento. Geralmente o valor do juros de mora é cobrado mediante um tempo de capitalização que não é um múltiplo inteiro do período de referência, portanto é necessário introduzir nos cálculos uma taxa equivalente

à taxa fixada que esteja de acordo com o tempo transcorrido. Por isso, antes de darmos exemplos dessa situação, vamos estabelecer o que são taxas equivalentes no regime de juros simples.

Dizemos que duas taxas de juros i_1 e i_2 referidas respectivamente a períodos p_1 e p_2 , são equivalentes se, para um intervalo de tempo $T = n_1 p_1 = n_2 p_2$, as duas taxas produzem os mesmos juros, qualquer que seja o capital inicial C .

Afirmamos que as taxas equivalentes em regime de juros simples são proporcionais.

De fato, se i_1 e i_2 são equivalentes, então $C n_1 i_1 = C n_2 i_2$ e portanto $\frac{i_1}{i_2} = \frac{n_2}{n_1}$. Mas da definição temos que $\frac{n_2}{n_1} = \frac{p_1}{p_2}$, de modo que

$$\frac{i_1}{i_2} = \frac{p_1}{p_2}, \text{ ou } \frac{i_1}{p_1} = \frac{i_2}{p_2}$$

Exemplo 1.2 Qual a taxa mensal equivalente (para juros simples) à taxa de 6% a.a.?

Solução: Seja i_m a taxa mensal pedida

Como o ano tem 12 meses, segue que

$$\frac{i_m}{1} = \frac{0,06}{12} \Rightarrow i_m = 0,005 \text{ ou } 0,5\% \text{ a.m.}$$

1.2.2 Juros Compostos

No regime de Capitalização Composta (juros compostos) o juro de cada período deve ser somado ao capital para o cálculo do juro do período seguinte. Neste caso, o juro de cada período é calculado sobre o montante do período anterior e não apenas sobre o capital inicial, como ocorre na capitalização simples.

Na capitalização em juros compostos os montantes são relativos a cada período de capitalização. Se $M_0 = C$ é o capital inicial e M_n é o montante correspondente após n períodos de capitalização à taxa i , temos que:

N	Período	Montante
0	Data inicial	$M_0 = C$
1	Após 1 período de capitalização	$M_1 = C(1 + i)$
2	Após 2 período de capitalização	$M_2 = C(1 + i)^2$

N	Período	Montante
3	Após 3 período de capitalização	$M_3 = C(1+i)^3$
\vdots	\vdots	\vdots
n	Após n período de capitalização	$M_n = C(1+i)^n$

Portanto, no regime de juros compostos, o montante gerado após n períodos de capitalização à taxa i é dado por

$$M_n = C(1+i)^n$$

A expressão acima pode ser confirmada por indução matemática.

Observação.: Os sucessivos montantes de uma capitalização composta são os termos de uma progressão geométrica de razão $1+i$ e primeiro termo $M_0 = C$.

Podemos também, como no regime de capitalização simples, obter uma expressão para os juros na capitalização composta, fazendo a diferença entre o montante e o capital aplicado em n períodos de capitalização:

$$J_n = M_n - C = C(1+i)^n - C = C((1+i)^n - 1).$$

Dizemos que duas taxas de juros i_n e i_k referidas respectivamente a períodos p_n e p_k , são equivalentes se, a partir do mesmo capital inicial, produzirem o mesmo montante, no mesmo espaço de tempo. Seja C um capital qualquer, T um espaço de tempo e i_n e i_k duas taxas. Seja i_n referida a um período p_n , de forma que $T = np_n$ e i_k referida a um período p_k , de modo que $T = kp_k$, com n e k inteiros. Portanto, se i_n e i_k são equivalentes, geram o mesmo montante, logo:

$$\begin{aligned} C(1+i_n)^n &= C(1+i_k)^k \Leftrightarrow (1+i_n)^n = (1+i_k)^k \\ &\Leftrightarrow 1+i_n = (1+i_k)^{\frac{k}{n}} \\ &\Leftrightarrow i_n = (1+i_k)^{\frac{k}{n}} - 1 \end{aligned}$$

Exemplo 1.3 Qual a taxa anual equivalente à taxa de 1% a.m.?

Solução: Seja i_a a taxa anual pedida. Como a taxa anual é aplicada uma vez, enquanto a mensal é aplicada doze vezes, temos:

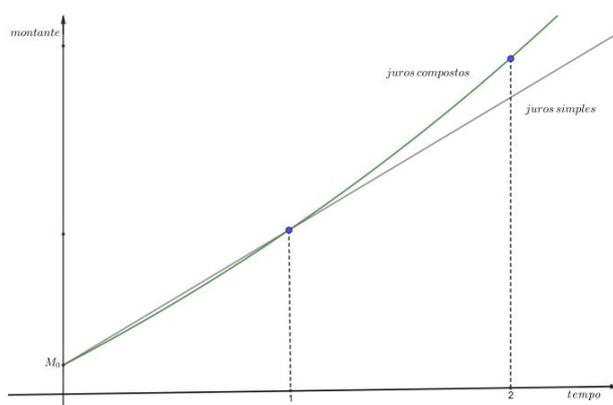
$$\begin{aligned} (1+i_a)^1 &= (1+0,01)^{12} \Rightarrow (1+i_a) \cong 1,12682503 \\ &\Rightarrow i_a \cong 0,12682503 \end{aligned}$$

Portanto, a taxa anual equivalente é 12,68% a.a.

Se um capital C é aplicado à taxa i por n períodos em regime de juros simples temos $M_n = C(1 + ni)$. Agora, se for no regime de juros compostos temos $M_n = C(1+i)^n$. Como $i > 0$, tem-se, para todo $n \geq 1$, $(1+i)^n \geq (1+ni)$, valendo a igualdade se e somente se $n = 0$ ou $n = 1$ (desigualdade de Bernoulli).

A Figura 2 ilustra a distinção geométrica entre os dois regimes:

Figura 2 – Comparação Juros simples x Juro composta



Fonte: Elaborada pelo autor

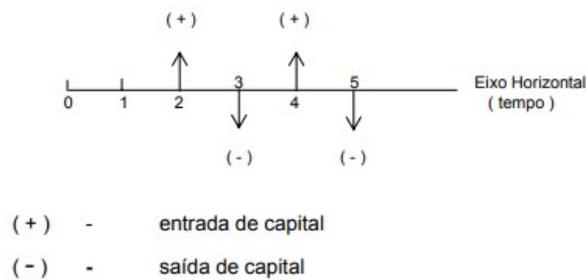
Do gráfico acima concluímos que:

- Nos períodos $n = 0$ e $n = 1$, o montante é o mesmo nos dois regimes.
- Quando o período variar entre 0 e 1 ($0 < n < 1$), o montante na capitalização simples é maior do que na composta. É por esse motivo que geralmente a capitalização simples só é utilizada para operações de curto prazo.
- Quando o período for maior do que 1, o montante na capitalização simples é menor do que na composta.

1.3 Fluxo de Caixa

Denomina-se Fluxo de Caixa a representação do conjunto de entrada e saídas de capital ao longo do tempo. A elaboração do fluxo de caixa é indispensável na análise de rentabilidade e custos de operações financeiras, e no estudo de viabilidade econômica de projetos e investimentos. Tal representação é feita por meio de tabelas, ou esquematicamente, como na figura abaixo:

Figura 3 – Representação gráfica Fluxo de caixa



Fonte: Elaborada pelo autor

Nos próximos exemplos utilizaremos, nos cálculos, duas fórmulas muito estudadas no ensino médio. A fórmula da soma dos termos de uma progressão geométrica, dada por:

$$S_n = \frac{a_1(q^n - 1)}{q - 1}$$

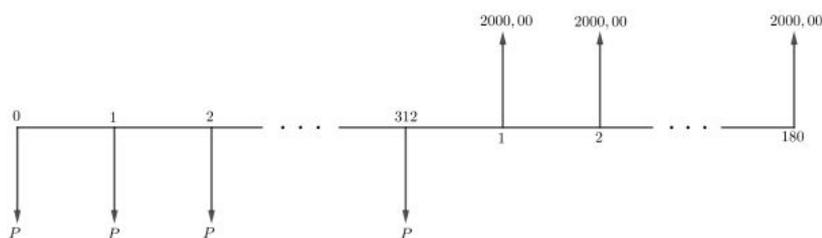
Onde a_1 , q e n são, respectivamente, o primeiro termo, a razão (diferente de 1) e a quantidade de termos da progressão geométrica. E a fórmula da soma de infinitos termos de uma progressão geométrica, dada p

$$S_\infty = \frac{a_1}{q - 1}$$

Onde a_1 e q são, respectivamente, o primeiro termo e a razão da progressão geométrica, com $-1 < q < 1$.

Exemplo 1.4 *Quantos reais deverão ser depositados mensalmente, por 26 anos, em um fundo de investimento que rende 0,5% ao mês, para que se possa resgatar, a partir do primeiro mês após o último depósito, a quantia mensal de R\$ 2000,00, durante 15 anos? E se o resgate mensal de R\$ 2000,00 fosse por tempo indefinido?*

Solução: Seja P o valor do depósito mensal. Um fluxo de caixa da primeira situação pode ser elaborado da seguinte forma (as datas se referem a meses):



Avançando os depósitos para a data 312 e recuando os resgates também para a data 312, obtemos a seguinte equação

$$P(1,005)^{312} + P(1,005)^{311} + \dots + P = \frac{(2000)}{(1,005)} + \dots + \frac{2000}{(1,005)^{180}}.$$

Colocando P e 2000 em evidencia, obtemos:

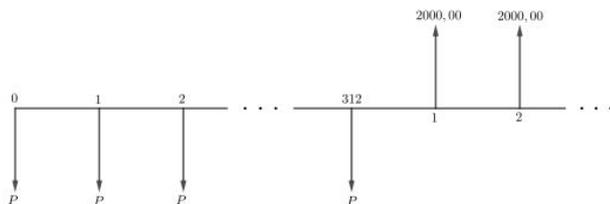
$$P((1,005)^{312} + \dots + 1) = 2000\left(\frac{1}{1,005} + \dots + \frac{1}{1,005^{180}}\right)$$

Usando a fórmula da soma dos termos de uma progressão geométrica, obtemos

$$P\left(1\left(\frac{(1,005)^{313} - 1}{1,005 - 1}\right)\right) = 2000\left(\frac{1}{1,005}\left(\frac{1,005^{180} - 1}{\frac{1}{1,005} - 1}\right)\right)$$

$$\Rightarrow P(752,81) \cong 2000 \cdot 118,50 \Rightarrow P \cong 314,82$$

Portanto, deverão ser depositados R\$ 314,82 mensais durante 26 anos para que se possa resgatar, a partir do primeiro mês após o último depósito, a quantia mensal de R\$ 2000,00, durante 15 anos. E se o resgate mensal de R\$ 2000,00 fosse por tempo indefinido, teríamos o seguinte fluxo de caixa:



Avançando os depósitos para a data 312 e recuando os resgates também para a data 312, obtemos a seguinte equação

$$P(1,005)^{312} + P(1,005)^{311} + \dots + P = \frac{(2000)}{(1,005)} + \dots + \frac{2000}{(1,005)^{180}}$$

Colocando P e 2000 em evidência, obtemos:

$$P\left((1,005)^{312} + \dots + 1\right) = 2000\left(\frac{1}{1,005} + \frac{1}{(1,005)^2} + \dots\right).$$

Usando a fórmula da soma dos termos de uma progressão geométrica no primeiro membro e a fórmula da soma de infinitos termos de uma progressão

geométrica no segundo membro da equação, obtemos

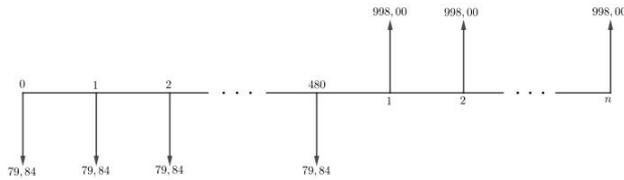
$$P \left(1 \left(\frac{(1,005)^{313} - 1}{1,005 - 1} \right) \right) = 2000 \left(\left(\frac{\frac{1}{1,005}}{\frac{1}{1,005} - 1} \right) \right) \Rightarrow P(752, 81) \cong 2000 \cdot 200$$

$$\Rightarrow P \cong 531, 34.$$

Portanto, deverão ser depositados R\$ 531,34 mensais durante 26 anos para que se possa resgatar, a partir do primeiro mês após o último depósito, a quantia mensal de R\$ 2000,00, por tempo indefinido.

Exemplo 1.5 *Uma pessoa se aposentou aos 65 anos de idade, tendo depositado mensalmente, por 40 anos, em um fundo de previdência privada, a quantia de R\$ 79,84 (8% de R\$ 998,00) a juros de 0,4% a.m. Se o resgate mensal foi estipulado em R\$ 998,00, a partir do primeiro mês após o último depósito, até que idade esta pessoa receberá sua aposentadoria?*

Solução: Seja n a quantidade de meses de resgate. Um fluxo de caixa do aposentado pode ser elaborado da seguinte forma (as datas se referem a meses e os valores em reais):



Avançando os depósitos para a data 480 e recuando os resgates também para a data 480, obtemos a seguinte equação

$$79,84(1,004)^{480} + 79,84(1,004)^{479} + \dots + 79,84 = \frac{998}{(1,004)} + \dots + \frac{998}{(1,004)^n}$$

Colocando 79,84 e 998 em evidência, obtemos

$$79,84 \left((1,004)^{180} + \dots + 1 \right) = 998 \left(\frac{1}{1,004} + \dots + \frac{1}{1,004^n} \right)$$

Usando a fórmula da soma dos termos de uma progressão geométrica, obtemos:

$$79,84 \left(1 \left(\frac{(1,004)^{481} - 1}{1,004 - 1} \right) \right) = 998 \left(\frac{1}{1,004} \left(\frac{1,004^{-n} - 1}{1,004^{-1} - 1} \right) \right)$$

Logo,

$$\left(\frac{1}{1,004}\right)^n = 0,5342336$$

Portanto,

$$n \cong \frac{\log(0,5342336)}{\log(1) - \log(1,004)} \cong 157,04$$

Como 157,04 meses equivalem a 13,09 anos aproximadamente, temos que essa pessoa receberá sua aposentadoria até os 78 anos de idade.

1.4 Séries de Capitais

Nas operações financeiras o capital pode ser pago (ou recebido) de uma só vez ou através de uma sucessão de pagamentos (ou recebimentos) que denominamos **Série de Capitais**. Abordaremos nesta seção as séries cujos termos (pagamentos ou recebimentos) são iguais e consecutivos (série constante ou uniforme) sob o regime de juros compostos. Denotaremos os termos (prestações) por PMT . Os vencimentos dos termos de uma série uniforme podem ocorrer no final de cada período (**Série Postecipada**), no início (**Série Antecipada**) ou após um período de carência (**Série Diferida**).

Afim de determinarmos uma fórmula para o valor das prestações de uma série uniforme, utilizaremos as seguintes notações:

- PMT : valor do pagamento periódico ou prestação;
- PV : Valor presente da série de pagamentos;
- i : a taxa de capitalização utilizada;
- n : o número de prestações da renda;
- FV : valor futuro da renda.

Para auxiliar na visualização das diversas situações apresentadas é utilizado o conceito de Fluxo de Caixa, que se trata de uma representação gráfica utilizada para mostrar as entradas e saídas de capital.

Considerando uma série uniforme postecipada com n termos, o valor presente PV dessa série na data focal zero, calculado a uma taxa de juros compostos i , corresponde à soma dos valores atuais dos seus n termos PMT nessa mesma data.

Desse modo podemos escrever:

$$PV = \frac{PMT}{(1+i)^n} + \frac{PMT}{(1+i)^{n-1}} + \dots + \frac{PMT}{(1+i)^2} + \frac{PMT}{(1+i)^1}$$

Utilizando a soma dos n termos de uma progressão geométrica concluímos que

$$PV = PMT \times \left[\frac{1 - (1+i)^{-n}}{i} \right] \quad (1.1)$$

Obs.: o fator $a_{ni} = \left[1 - \frac{(1+i)^{-n}}{i} \right]$ é chamado de **Fator de Valor Atual**.

Logo o valor futuro FV dessa série na data focal n , à mesma taxa de juros compostos i , é dado por:

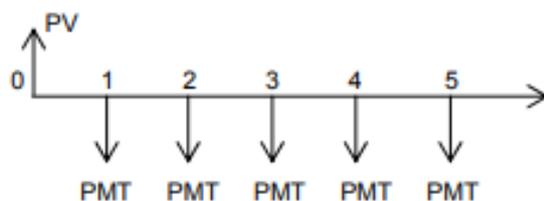
$$FV = PMT \times \left[\frac{-1 + (1+i)^n}{i} \right] \quad (1.2)$$

Obs.: o fator $S_{ni} = \left[\frac{-1 + (1+i)^n}{i} \right]$ é chamado de **Fator de Acumulação de Capital**.

Apresentamos as fórmulas de PV e FV para uma série postecipada. Essas fórmulas, juntamente com o princípio de equivalência de capitais, nos permitirão resolver os mais variados problemas referentes às séries antecipadas e diferidas.

Séries Perpétua

Uma série perpétua é um conjunto de capitais cujo número de termos não é determinado, tendendo ao infinito, como aposentadorias, seguros, impostos, etc. Trataremos da série perpétua uniforme postecipada, representada no fluxo abaixo:



O valor dessa série é o limite de $PV = PMT \times \left[\frac{1 - (1+i)^{-n}}{i} \right]$ quando n tende ao infinito.

Desse modo,

$$\lim_{N \rightarrow \infty} PMT \times \left[\frac{1 - (1+i)^{-n}}{i} \right]$$

Concluímos que

$$PV = \frac{PMT}{i}$$

Portanto, é possível calcular o valor presente de um bem alugado ou de um investimento sem prazo determinado apenas com os valores da parcela e da taxa

de juros. Conseqüentemente, dado o valor presente e a taxa é possível calcular o valor da prestação ou aluguel de um bem.

Exemplo 1.6 *Um imóvel foi alugado, por tempo indeterminado, com prestações mensais de R\$ 1500,00 a uma taxa de 2% ao mês. Qual o valor do imóvel?*

Resolução: Como o prazo do aluguel é indeterminado e o valor das parcelas é constante, podemos utilizar a expressão da série uniforme infinita ou perpetuidade. Desta forma, utilizando a expressão do valor presente PV

$$PV = \frac{PMT}{i} \Rightarrow \frac{1500}{0,02} \Rightarrow PV = 75.000,00$$

Portanto, apenas com o valor das prestações e da taxa de juros utilizada, foi possível calcular o valor presente do imóvel.

1.5 Sistemas de Amortização de Empréstimos e Financiamentos

No processo de reembolso de empréstimos de longo e médio prazo são utilizados diversos sistemas de amortização. Nestes sistemas a dívida é liquidada através de pagamentos periódicos (prestações), que consistem de amortização (devolução do principal emprestado) e encargos (juros, taxas, seguros, impostos, etc). Com relação aos encargos, vamos considerar apenas os juros, já que os demais variam conforme o contrato estabelecido entre as partes.

Denotaremos a k -ésima prestação por PMT_k e podemos escrever:

$$PMT_k = A_k + J_k$$

sendo A_k a parcela da amortização e J_k o juro, referentes à k -ésima prestação.

O juro J_k é calculado da seguinte forma:

$$J_k = i \times SD_{k-1}$$

onde SD_{k-1} é o saldo devedor do período $k - 1$.

Dentre os principais sistemas de amortização de empréstimos destacaremos os seguintes:

- Sistema de Amortização Constante (SAC).
- Sistema Americano de Amortização (SAA).
- Sistema de Amortização Francês (SAF) - Tabela Price.

Qualquer sistema de amortização pode ter ou não carência. Durante esse prazo os juros podem ser pagos (carência parcial) ou capitalizados e incorporados ao principal (carência total).

Em todos os sistemas de amortização, cada pagamento é a soma do valor amortizado com os juros do saldo devedor: $PMT = A + J$, onde PMT é a prestação (pagamento), A é a amortização e J os juros. Nas nossas análises dos diferentes tipos de sistemas de amortização, utilizaremos uma tabela onde terão as informações principais sobre o sistema adotado. Em todos os nossos exemplos, utilizaremos um financiamento hipotético de R\$ 300.000,00 que será pago ao final de 5 meses a uma taxa de 4% a.m.(SODRE, 2020) .

Sistema de Amortização Constante

No SAC as amortizações são constantes e calculadas dividindo-se o valor do principal pelo número de prestações n .

$$A = \frac{PV}{n} \quad (1.3)$$

Nessa perspectiva, o tomador realiza amortizações de mesmo valor no final de cada período. No nosso exemplo, como o valor financiado é R\$ 300.000,00e será amortizado em 5 parcelas, faremos $R\$ 300.000,00 : 5 = R\$ 60.000,00$. Este valor representa a amortização no final de cada período. Este sistema é muito utilizado no financiamento da casa própria.

Sistema de Pagamentos Variáveis

N	Juros	Amortização	Prestação	Saldo devedor
0	0	0	0	300.000,00
1	12.000,00	60.000,00	72.000,00	270.000,00
2	9.600,00	60.000,00	69.600,00	225.000,00
3	7.200,00	60.000,00	67.200,00	165.000,00
4	4.800,00	60.000,00	64.800,00	90.000,00
5	2.400,00	60.000,00	62.400,00	0
Totais	36.000,00	300.000,00	336.000,00	

Sistema de Amortização Francês

No SAF as prestações são iguais e calculadas segundo uma série uniforme postecipada. Esse sistema é o mais utilizado pelo mercado, pois há programas es-

pecíficos nas calculadoras financeiras que facilitam a montagem do plano de amortizações. Um caso particular desse sistema é a Tabela Price, onde a taxa de juros é anual e as prestações são mensais; utiliza-se então a taxa mensal proporcional. Utilizaremos o mesmo exemplo do sistema anterior.

Os valores das prestações são iguais. Ou seja, a soma dos juros com a amortização ao final de cada período é constante. É muito utilizado em financiamentos de bens duráveis como automóveis, eletrodomésticos, etc., e também no comércio em geral, como nas compras a prazo onde se cobram os juros.

Usando a equação (2.7), teremos $PMT = R\$ 67.388,13$

Sistema de Pagamentos Variáveis

N	Juros	Amortização	Prestação	Saldo devedor
0	0	0	0	300.000,00
1	12.000,00	55.388,13	67.388,13	244.611,87
2	9.784,47	57.603,66	67.388,13	187.008,21
3	7.480,32	59.907,81	67.388,13	127.100,40
4	5.084,01	62.304,12	67.388,13	64.796,28
5	2.591,85	64.796,28	67.388,13	0
Totais	36.940,65	300.000,00	336.940,15	

Sistema Americano de Amortização

No SAA o principal é devolvido em uma só parcela, ao final do prazo do empréstimo. Os juros podem ser pagos periodicamente, ou capitalizados e incorporados ao saldo devedor. Continuaremos com o mesmo exemplo anterior.

O devedor paga a dívida no final do último período, porém se compromete a pagar os juros de cada período.

SISTEMA DE PAGAMENTOS VARIÁVEIS

N	Juros	Amortização	Prestação	Saldo devedor
0	0	0	0	300.000,00
1	12.000,00	30.000,00	42.000,00	270.000,00
2	10.800,00	45.000,00	55.800,00	225.000,00
3	9.000,00	60.000,00	69.000,00	165.000,00
4	6.600,00	75.000,00	81.600,00	90.000,00
5	3.600,00	90.000,00	93.600,00	0
Totais	42.000,00	300.000,00	342.000,00	

1.6 Sistema Financeiro

As seções seguintes desta seção, apresentamos os conceitos e definições relacionados ao sistema financeiro, tomando como base as principais referências usadas foram os livros (NETO, 2000b),(VIDOR, 2016),(JARDIM et al., 2016).

Muitos termos usados na atualidade são desconhecidos pela maioria dos alunos e por uma parte dos professores. É positivo que o docente compreenda o funcionamento dos sistemas financeiros, cobranças de impostos, entre outros assuntos, para assim poder inseri-los no seu cotidiano escolar. Nesta seção faremos uma breve apresentação dos principais temas.

O sistema financeiro é um conjunto de instituições, instrumentos e mercados que permitem a circulação do dinheiro na economia de um país. É um componente fundamental da economia moderna, pois permite que os recursos financeiros sejam alocados de forma eficiente e apropriada.

As instituições financeiras incluem bancos comerciais, bancos de investimento, seguradoras, empresas de cartões de crédito, entre outros. Estas instituições oferecem diversos produtos e serviços financeiros, tais como empréstimos, depósitos, investimentos, seguros, entre outros.

Os instrumentos financeiros são títulos que representam um valor monetário e são utilizados para captar recursos financeiros ou transferir riscos. Exemplos de instrumentos financeiros incluem ações, títulos de dívida, derivativos, entre outros.

Já os mercados financeiros são onde são negociados os instrumentos financeiros. Esses mercados podem ser organizados, como a bolsa de valores, ou podem ser de balcão, como o mercado de derivativos.

O sistema financeiro é regulado pelos governos e por entidades reguladoras, que têm como objetivo garantir a estabilidade do sistema financeiro e proteger os consumidores. A regulação inclui regras para a criação e operação das instituições financeiras, para a emissão e negociação de instrumentos financeiros e para a proteção dos consumidores.

Em resumo, o sistema financeiro é um componente importante da economia, pois permite a alocação eficiente dos recursos financeiros e ajuda a promover o crescimento econômico.

Todos os países desenvolvidos ou em desenvolvimento buscam crescimento econômico, estabilidade de preços e distribuição de renda. Para alcançar e/ou manter esses objetivos os governos utilizam-se das políticas econômicas.

Segundo (LIEDTKE, 2019), política econômica é um conjunto de ações

planejadas e executadas pelo governo, com o objetivo de produzir um certo impacto na situação econômica local e também cumprir propósitos sociais. Seus principais objetivos são a estabilização dos preços, o crescimento econômico, a criação de empregos e distribuição de riquezas. As políticas econômicas podem ser divididas em: monetária, fiscal, cambial e de rendas.

O governo precisa estimular a economia. Mas quais são as estratégias? Será que aumentar a quantidade de capital no mercado melhora a situação econômica do país? A resposta é não. Pois uma grande quantidade de moeda faz o mercado inflar, ocasionando a inflação. Sendo assim, o governo através do Banco Central trabalha com mecanismos para controlar a quantidade de moeda na economia.

Na política fiscal o governo deve fazer seu orçamento, isto é, o resultado das receitas menos os gastos públicos. Se os gastos forem maiores do que a receita, o governo vende títulos de crédito para angariar recursos, e a taxa que o governo paga para os investidores que lhe emprestam dinheiro é chamada de taxa Selic. Essa taxa baliza todas as outras transações no mercado financeiro. A taxa Selic é chamada taxa básica de juros do Brasil. Dentro da política monetária, o governo controla essa taxa. O COPOM – Comissão de Política Monetária é órgão responsável em implementar a política monetária; definir a meta da taxa SELIC e seu eventual viés; analisar o Relatório de Inflação; definir a taxa Selic meta; cumprir a meta de inflação definida pelo Conselho Monetário Nacional. O Copom é composto pelos membros da Diretoria Colegiada do Banco Central do Brasil.

Uma das principais funções dos bancos comerciais é emprestar dinheiro. Para isso, eles arrecadam valores das pessoas por meio de poupanças e outras aplicações financeiras. Essa arrecadação tem um custo, conhecido como taxa de captação. Para cobrir suas despesas e obter lucro, os bancos, quando emprestam ou financiam, cobram taxas de juros maiores que essa taxa de captação. A diferença entre as duas taxas é chamada de *spread* bancário. As explicações mais aceitas para grandes *spreads* estão relacionadas ao risco de inadimplência que as instituições têm ao emprestar dinheiro e aos grandes lucros almejados pelas mesmas. Isso se torna mais extremo quando se considera o uso do cheque especial e o cartão de crédito. Os juros chegam a passar, em média, de 300% ao ano no cheque especial e cartão de crédito. Quando se opta em pagar o valor mínimo de uma fatura de cartão de crédito, por exemplo, isso significa pegar um empréstimo com taxas de juros altíssimas.

- **Taxas Básicas de Remuneração do Capital**

As taxas básicas de remuneração do capital se referem às taxas de juros

que são pagas aos investidores pelo dinheiro que eles investem em diversas modalidades de investimentos. Essas taxas são estabelecidas pelos próprios emissores dos títulos, e podem variar de acordo com o risco do investimento e com as condições do mercado financeiro.

SELIC - Calculada e divulgada diariamente pelo Selic (Sistema Especial de Liquidação e Custódia), representa a taxa média das operações de financiamento por um dia, lastreadas em títulos públicos federais na forma de operações compromissadas. Expressa na forma anual, base de 252 dias úteis, é a taxa, efetivamente, praticada no mercado interfinanceiro. No Brasil, a taxa básica do mercado de capital é a taxa Selic, que é a taxa de juros definida pelo Comitê de Política Monetária (COPOM) do Banco Central. A Selic é utilizada como referência para a definição de juros em diversos tipos de investimentos, como títulos públicos, fundos de investimento e poupança.

Exemplo 1.7 *Considere que o COPOM tenha divulgado em uma de suas reuniões que a meta para a taxa SELIC é 9,5% ao ano. Calcule a taxa equivalente por dia útil, sabendo que o Banco Central padronizou o número de dias úteis do ano e do mês como 252 e 21, respectivamente.*

Resolução: Pela equação tem-se $1 + I = (1 + i)^t$, daí

$$1 + 0,095 = (1 + i)^{252}$$

$$\Rightarrow 1,095^{\frac{1}{252}} = 1 + i$$

$$\Rightarrow i = 1,095^{\frac{1}{252}} - 1$$

$$\Rightarrow i \approx 0,0004 = 0,04\% \text{ ao dia útil.}$$

CDI - O CDI é uma espécie de empréstimo que um banco faz a outro por um determinado período de tempo. O banco que empresta o dinheiro é chamado de "tomador" e o banco que recebe o dinheiro é chamado de "depositante". O tomador paga uma taxa de juros ao depositante, que é negociada entre as partes e pode variar de acordo com as condições do mercado.

Exemplo 1.8 *Um banco emprestou R\$ 2.500.000 para outro banco, via CDI, por um período de 4 dias úteis, recebendo ao final R\$ 2.506.000,00. Determine o valor da taxa CDI média ao dia útil, dessa operação interbancária.*

Resolução: Inicialmente, vamos calcular a taxa percentual durante o período de 4 dias úteis $\frac{2.506.000}{2.500.000} - 1 = 1,0024 - 1 = 0,0024 = 0,24\%$.

Daí, sendo i_{cdi} o valor da taxa CDI ao dia útil (a.u.), tem-se

$$1 + 0,0024 = (1 + i_{_cdi})^4 \Rightarrow _cdi \approx 0,0006 = 0,06\%a.u.$$

O conceito de indexador - O conceito de índice e indexador baseia-se na Teoria dos Números Índices. A ideia é a de fixar um spread* e deixar flutuar a parcela que apresenta volatilidade, sendo, esta última medida por um indexador ou índice reconhecidamente oficial. Por exemplo: $TR + 3\%$ a.a.; ou $INPC + 4,5\%$ a.a. Nestes exemplos, o spread foi fixado em 3% a.a. e $4,5\%$ a.a., enquanto que a parcela que apresenta volatilidade, flutuará e será calculada no final da operação através dos indexadores contratados nas operações.

A política cambial tem como objetivo tomar medidas para controlar o câmbio e tornar a balança comercial mais favorável. Promove uma melhor distribuição de renda entre a população. Um exemplo dessa política é a fixação do salário mínimo.

O que vemos no Brasil é que quando muda o governo, muda a política econômica, pois não existe um compromisso de continuidade, o que dificulta a implementação de uma política consistente em longo prazo.

1.7 Investimentos

Nesta seção serão descritas formas de investimentos financeiros utilizados com o objetivo de obtenção de rendimentos dos valores poupados. Não é o objetivo deste trabalho se aprofundar nas peculiaridades dos vários produtos de investimento existentes no mercado, portanto serão apresentadas as modalidades mais utilizadas atualmente mercado financeiro.

Uma das formas das pessoas evitarem o endividamento é aprendendo a planejar e organizar as finanças, tentando separar uma parte do orçamento familiar para investir e, assim, possuir uma reserva em casos de emergências. Ao ensinar os tipos de investimentos, a escola colabora para a formação de cidadãos mais capazes de tomar decisões mais sadias para sua vida financeira.

Quando o investidor procura a melhor opção para investir, está buscando uma alternativa de investimento que melhor combine os atributos básicos: rentabilidade, liquidez e segurança. Apesar da dificuldade de encontrar os três atributos em um mesmo investimento, o investidor terá que optar pelo equilíbrio dos fatores que julgar mais importantes.

Existem basicamente dois tipos de investimentos, os de renda fixa e de renda variável.

1.7.1 Renda Fixa

Investimentos de renda fixa são aqueles em que o investidor empresta dinheiro a uma instituição financeira ou empresa e, em troca, recebe uma remuneração predeterminada. Essa remuneração pode ser paga por meio de juros, dividendos ou outras formas de rendimentos.

Os investimentos de renda fixa são considerados mais conservadores e menos arriscados do que os investimentos de renda variável, como ações e fundos imobiliários, por exemplo. Isso ocorre porque, no geral, os rendimentos dos investimentos de renda fixa são previsíveis e menos sujeitos às flutuações do mercado.

A taxa de juros de investimentos em renda fixa pode ser estabelecida de maneira pré-fixada ou pós-fixada. Na taxa pré-fixada, é conhecido a taxa nominal de juros no ato da compra do título, sabendo de forma antecipada a rentabilidade do título, por exemplo 10% a.a., 0,5% a.m. entre outros. Já na taxa pós-fixada, a rentabilidade é estabelecida por meio de uma taxa de juros estabelecida previamente somada a um fator de correção de um indexador, que pode variar ao longo do tempo, dessa forma, a rentabilidade só será conhecida ao fim do investimento. Os indexadores mais comuns utilizados são o *IPCA*, *IPGPM*, *SELIC* e *CDI*, por exemplo 6,5% a.a. + *IPCA*, 120% *CDI*, *SELIC*.

(CERBASI, 2009) aponta que cada modalidade, pós-fixada e pré-fixada, são melhores em situações distintas.

Se a tendência dos juros da economia é de queda, você fará melhor negócio optando por taxas prefixadas, que garantirão rentabilidades próximas às atuais por mais tempo. Por outro lado, se a queda nos juros de mercado for certa, seu banco lhe oferecerá uma taxa menor na renda fixa prefixada. Se o futuro é incerto, as taxas pós-fixadas são melhores, pois, em caso de inflação, o governo tende a elevar juros para desaquecer a economia e seu dinheiro crescerá mais. (CERBASI, 2009, p. 130)

Exemplo 1.9 *Oswaldo é uma investidor com perfil mais conservador e decidiu investir R\$ 2.500,00 em títulos de renda fixa por um período de 2 anos. Do valor disponível, R\$ 1.000,00 foram investidos à uma taxa pré-fixada de 8% a.a., e os outros R\$ 1.500,00 foram investidos a taxa pós-fixada de 2% a.a. + IPCA. Sabendo que a taxa IPCA nesses dois anos foi, respectivamente, igual a 4,5% a.a. e 7% a.a., qual foi o montante resgatado por Oswaldo após esses dois anos?*

Solução:

- Título pré-fixado: $M_1 = 1.000(1 + 0,08)^2 = 1.166,40$

- Título pós-fixado: $M_2 = 1.500(1 + 0,02)(1 + 0,045)(1 + 0,02)(1 + 0,07) = 1744,98$

$$M_T = M_1 + M_2 = 1.166,40 + 1.744,98 = 2.911,38.$$

Assim, o montante resgatado foi de R\$ 2.911,38.

Existem vários tipos de investimentos de renda fixa, cada um com suas características específicas. Entre os mais comuns, podemos citar:

Títulos públicos: são títulos emitidos pelo governo federal e negociados no Tesouro Direto. Podem ser prefixados, pós-fixados ou híbridos, e têm diferentes prazos de vencimento.

As aplicações de renda fixa normalmente oferecem maior estabilidade ao investidor, pois, para valores até R\$ 250 mil, existe uma garantia do Fundo Garantidor de Crédito (FGC), que mesmo que a empresa entre em falência, ela garante o valor ao investidor, ação essa inexistente para os fundos de renda variável. Os rendimentos de renda fixa podem ser fracionados em 3 tipos:

- **Pré-fixados:** a taxa prefixada traduz-se em um rendimento fixo, por exemplo, 6% ao ano. Essa taxa continuará a mesma até o final do prazo da aplicação. Os investimentos desse tipo costumam ser indicados quando há estimativa de queda nas taxas de juros. Exemplos de aplicações desse tipo são o Tesouro Direto, que é um programa criado pelo governo federal que possibilita que as pessoas físicas possam comprar títulos públicos. Ele é considerado um investimento seguro, pois é improvável que o governo quebre. O Certificado de Depósito Bancário (CDB) é muito parecido com o Tesouro Direto, mas é como se o investidor tivesse emprestado dinheiro para o banco, enquanto que as Letras de Câmbio (LC), você empresta para instituições financeiras.

- **Pós fixados:** eles têm a rentabilidade associada a um indexador, a exemplo da Taxa Selic ou o CDI. Ou seja, se os indexadores sobem, a rentabilidade aumenta. Mas se eles descem, a rentabilidade diminui. Alguns exemplos são o Tesouro Direto Selic, a Letra de Crédito Imobiliário (LCI) e a Letra de Crédito do Agronegócio (LCA), que se dá quando você empresta para o ramo imobiliário ou para o agronegócio. A grande vantagem desses dois investimentos é que são isentos de Imposto de Renda e IOF para pessoas físicas.

- **Híbridos:** a rentabilidade híbrida apresenta uma parte fixa e uma variável, tal como, 6% mais o IPCA. Os investimentos desta espécie geralmente possuem como indexadores de inflação o Índice de Preços ao Consumidor Amplo (IPCA) ou o Índice Geral de Preços do Mercado (IGP-M). Alguns exemplos são o Tesouro Direto IPCA o Debênture, quando você empresta dinheiro para uma empresa e a Letra

de Crédito do Agronegócio (LCA), que é quando você investe no agronegócio. Este também é livre de imposto de renda.

Caderneta de Poupança

A caderneta de poupança é uma forma de investimento de baixo risco cuja operação é regida por regras específicas estabelecidas pelo governo federal.

Existem duas regras para remuneração de acordo com a Medida Provisória 567 de 03 de maio de 2012 (convertida na lei 12.703 de 07 de agosto de 2012):

- Para os depósitos anteriores à Medida Provisória a remuneração é a TR - Taxa Referencial (taxa mensal criada em 1991 para servir de referência para a taxa de juros no Brasil) mais 0,5 ponto percentual ao mês.
- Para os depósitos posteriores à Medida Provisória a remuneração passa a ser de TR mais 70% da meta da taxa da *Selic* (taxa básica de juros - definida pelo Banco Central) sempre que a meta for igual ou menor que 8,5%. Caso a meta da taxa Selic seja superior a 8,5% a remuneração das cadernetas permanece como TR mais 0,5 ponto percentual ao mês.

O rendimento da poupança é calculado mensalmente de acordo com a data do depósito, ou seja, é capitalizado somente a cada 30 dias.

Exemplo 1.10 *Calcular a taxa de juros efetiva anual da Caderneta de Poupança sabendo que seu rendimento anual é de 6% com capitalização mensal. Resp.: 6,17% ao ano*

Solução:

A taxa de juros efetiva anual (TJEA) da Caderneta de Poupança pode ser calculada usando a fórmula:

$$TJEA = (1 + i/n)^n - 1$$

onde i é a taxa de juros nominal anual, n é o número de vezes que os juros são compostos durante o ano.

Para este exemplo, a taxa de juros nominal anual é de 6%, com capitalização mensal. Portanto, temos: Temos,

$$i = 6\% = 0,06 \text{ (taxa de juros nominal anual)}$$

$$n = 12 \text{ (capitalização mensal)}$$

Substituindo na fórmula acima, temos:

$$TJEA = (1 + 0,06/12)^{12} - 1$$

$$TJEA = 1,005^{12} - 1$$

$$TJEA = 0,061678 \text{ ou aproximadamente } 6,17\%$$

Portanto, a taxa de juros efetiva anual da Caderneta de Poupança com rendimento de 6% ao ano e capitalização mensal é de aproximadamente 6,17%.

A caderneta de poupança é considerada de baixo risco, pois até um determinado limite a pessoa tem seu investimento garantido pelo Fundo Garantidor de Créditos (associação civil que administra mecanismos de proteção a correntistas, poupadores e investidores). Essa proteção garante que, mesmo que a instituição financeira decreta falência, o investidor receba o dinheiro que aplicou até esse limite. A partir de maio de 2013, esse valor passou a ser de R\$ 250.000,00 por CPF. Além da caderneta de poupança, outras opções também são garantidas, como o crédito em conta corrente, CDB e RDB.

Os investimentos de baixo risco pagam taxas mais baixas porque é pequena a probabilidade de que o retorno obtido na aplicação seja diferente daquilo que é esperado. Já os investimentos de risco requerem um pouco mais de estudo e disciplina financeira.

CDBs (Certificados de Depósito Bancário)

O Certificado de Depósito Bancário (*CDB*) é um título de renda fixa emitido por bancos como forma de captação de recursos. Funciona como um empréstimo que você faz ao banco e que, em troca, recebe juros por isso.

Ao investir em um *CDB*, o investidor empresta dinheiro ao banco por um prazo determinado e, em troca, recebe juros sobre o valor emprestado. Os juros podem ser pré-fixados, ou seja, o investidor sabe exatamente quanto irá receber no final do prazo, ou pós-fixados, atrelados a algum índice de referência, como a *taxa Selic* ou o *CDI*.

O *CDB* pode ser uma boa opção para investidores que buscam segurança, pois é protegido pelo Fundo Garantidor de Créditos (FGC), que garante o reembolso de até R\$ 250.000,00 por CPF em caso de falência do banco emissor.

Além disso, existem diversas modalidades de *CDB*, como os prefixados, os pós-fixados, os indexados à inflação, entre outros, o que permite que o investidor escolha aquele que melhor se adapta aos seus objetivos e perfil de investimento.

Por fim, é importante lembrar que o *CDB* pode apresentar liquidez diária ou não, o que deve ser considerado na hora de escolher o título mais adequado

para sua estratégia de investimento.

Na hora de aplicar um dinheiro num *CDB*, o investidor deve pesquisar o banco que oferece uma maior rentabilidade. Não é necessário ser correntista do banco pra adquirir um *CDB*. O *IR* é cobrado apenas na data do resgate ou no vencimento da aplicação. Desde janeiro de 2005, o imposto de renda passou a ser cobrado sobre os rendimentos de acordo com o prazo de permanência dos recursos - conforme tabela abaixo:

Tabela 5 – **CALCULO DO I.R. NOS CDBs**

Tempo de permanência	Alíquota
Até 180 dias	22,5%
de 181 dias a 360 dias	20%
de 361 dias a 720 dias	17,5%
acima de 720 dias	15%

Dessa forma, quanto mais tempo os recursos permanecerem aplicados, menor será a alíquota do imposto cobrado. Existe incidência de *IOF* nos resgates realizados antes de 30 dias, limitado ao rendimento da aplicação e em função do prazo, conforme tabela padrão para todas as instituições financeiras. A partir do trigésimo dia, a aplicação fica isenta da cobrança de *IOF* (Imposto sobre Operações Financeiras). Também costuma-se cobrar taxas administrativas sobre o rendimento. Portanto, o investidor pode ter um rendimento baixo após os descontos da taxa de administração e do Imposto de Renda.

Existem também os *CDBs* com carência, isto é, o investidor só poderá resgatar o capital com o rendimento após um certo período definido no ato da operação. Estes *CDBs* apresentam restrições nos resgates antecipados, porém costumam oferecer melhor rentabilidade. *CDBs* com liquidez diária não apresentam restrições nos resgates, mas tendem a oferecer menor rentabilidade.

Títulos públicos

Os títulos públicos são títulos de dívida emitidos pelo governo federal para captar recursos no mercado financeiro. Eles são uma opção de investimento em renda fixa considerada segura e acessível, que permite que investidores de diferentes perfis possam aplicar seu dinheiro em dívida pública, com baixo risco e rentabilidade.

Os títulos públicos são negociados por meio do Tesouro Direto, uma plataforma online disponibilizada pelo Tesouro Nacional para a compra e venda desses

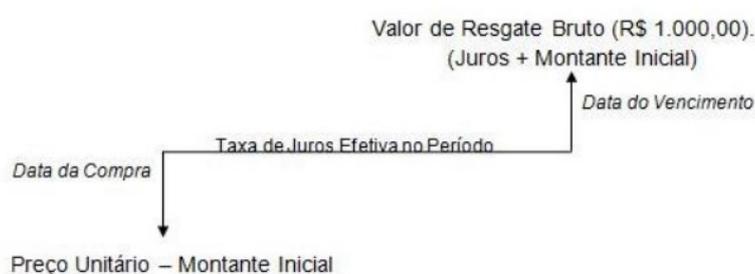
títulos de forma simples e segura.

Existem diferentes tipos de títulos públicos disponíveis no Tesouro Direto, que se diferenciam pela forma de cálculo da rentabilidade, prazo de vencimento e risco. A seguir, apresentamos os principais tipos de títulos públicos:

- Tesouro Prefixado (LTN): é um título prefixado cuja rentabilidade é definida no momento da compra e não sofre variações ao longo do tempo. O Tesouro Prefixado é indicado para investidores que querem garantir uma taxa fixa de rendimento durante o prazo do investimento.

As LTNs são títulos públicos com o fluxo de caixa simples, com valor nominal de R\$ 1.000,00, isto é, o investidor na *data zero* da negociação paga pelo preço de compra e após um período pré-definido resgatará o valor os R\$ 1.000,00.

Figura 4 – Fluxo Caixa



Fonte: Brasil (2018, s.p).

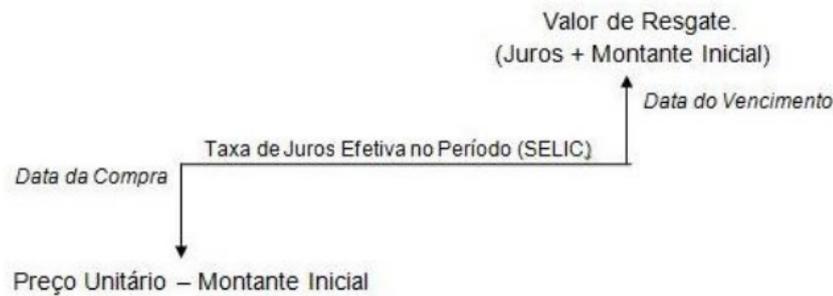
- Tesouro Selic (LFT): é um título pós-fixado cuja rentabilidade é atrelada à taxa Selic, a taxa básica de juros da economia brasileira. É considerado o título mais conservador do Tesouro Direto, pois seu rendimento acompanha a variação da Selic, que é uma taxa estável e de baixo risco. O valor do capital principal é atualizado usando a taxa Selic diária acumulada no período, desde a data zero até a data do resgate (vencimento).

A seguir um exemplo do fluxo de caixa de uma LFT.

O Preço Unitário – Montante Inicial que é marcado pela data de compra, trata-se da quantia total que o investidor irá gastar comprando o(s) título(s). A taxa de juros efetiva no período é o total acumulado de juros da taxa Selic durante o período compreendido entre a compra até a venda do título. Essa taxa acumulada da Selic é calculada no site <<https://tinyurl.com/bcbgovbr>>

O Valor de Resgate, marcado pela data de venda, é a quantia líquida que o

Figura 5 – Fluxo Caixa



Fonte: Brasil (2018, s.p).

investidor irá receber. Este valor compreende o montante que o investidor gastou comprando o título mais os juros acumulado no período.

- Tesouro Prefixado com Juros Semestrais (NTN-F): é um título prefixado que paga juros semestralmente. É indicado para investidores que precisam de uma renda periódica para complementar a renda.
- Tesouro IPCA+ (NTN-B): é um título pós-fixado cuja rentabilidade é atrelada ao IPCA, o índice oficial de inflação do país. O Tesouro IPCA+ é indicado para investidores que querem proteger seu investimento da inflação e obter uma taxa real de rendimento.
- Tesouro IPCA+ com Juros Semestrais (NTN-B): é um título pós-fixado que paga juros semestralmente e tem sua rentabilidade atrelada ao IPCA. É indicado para investidores que precisam de uma renda periódica para complementar a renda e querem proteger seu investimento da inflação.

Esses são os principais tipos de títulos públicos disponíveis no Tesouro Direto. Eles permitem que investidores de diferentes perfis possam diversificar sua carteira de investimentos e obter rendimentos com baixo risco e transparência.

Debêntures

São títulos emitidos pelas empresas para captar recursos. Podem ser simples ou conversíveis em ações, e têm diferentes prazos de vencimento e taxas de remuneração.

Letras de Crédito

São títulos emitidos por instituições financeiras com o objetivo de financiar o setor imobiliário (LCIs) ou o agronegócio (LCAs). Têm prazos e taxas de remuneração específicos.

Antes de investir em renda fixa, é importante avaliar as características de cada tipo de investimento e escolher aquele que melhor se adequa ao seu perfil de investidor e aos seus objetivos financeiros.

1.8 Renda Variável

O investimento em renda variável se caracteriza por investimentos em ativos financeiros cujos valores flutuam ao longo do tempo, como ações, opções, derivativos, fundos imobiliários, entre outros.

Ao contrário dos investimentos em renda fixa, onde o retorno é previsível, o investimento em renda variável envolve um grau maior de incerteza, já que os preços dos ativos podem variar significativamente em curtos períodos de tempo. Isso significa que há um risco maior de perdas em comparação com investimentos em renda fixa.

No entanto, o investimento em renda variável também oferece a possibilidade de obter retornos muito mais altos do que os investimentos em renda fixa, especialmente no longo prazo. Além disso, o investimento em renda variável permite que os investidores diversifiquem suas carteiras de investimento, reduzindo assim o risco de perdas.

Uma das principais formas de investir em renda variável é através da bolsa de valores. Os investidores podem comprar ações de empresas listadas na bolsa de valores, com o objetivo de lucrar com a valorização das ações ao longo do tempo. Além disso, existem outras opções de investimentos em renda variável, como opções, futuros, fundos de investimento em ações, fundos imobiliários, entre outros.

É importante lembrar que, como em qualquer tipo de investimento, é fundamental estudar bem as opções disponíveis antes de investir em renda variável.

Neste tipo de investimento é importante ter conhecimento sobre a dinâmica do mercado financeiro e estar ciente dos riscos envolvidos. Além disso, é recomendado que o investimento em renda variável seja feito com uma estratégia de longo prazo, com o objetivo de obter retornos consistentes ao longo do tempo.

No mercado de renda variável, as transações são feitas pela bolsa de valores, um “ambiente eletrônico de negociação que recebe, por meio das corretoras

de valores, ordens de compra e venda enviadas pelos investidores, pela internet ou por meio da mesa de operações das corretoras” (CORDEIRO; COSTA; SILVA, 2018). Hoje, o ambiente responsável por controlar essas transações é a B3, e em seu sítio eletrônico são disponibilizados cursos gratuitos sobre investimentos, principalmente em Renda Variável. Esses cursos são importantes, uma vez que para e investir em Renda Variável – e em Renda Fixa – é importante que o investidor tenha conhecimento sobre economia, mercado e empresas.

A seguir, apresentamos alguns tipos de investimentos em Renda Variável.

Ações

Uma ação é a menor fração do capital social de uma empresa. Isso significa que qualquer pessoa pode ser dona de grandes empresas, como Vale e Petrobrás, comprando uma parte do seu capital. Ainda há a possibilidade de escolher o tipo de ação que será adquirida: as ordinárias, que dão direito a voto nas assembleias de cotistas (donos), ou as preferenciais, que têm prioridade no recebimento de dividendos e outros proventos em detrimento da possibilidade de voto.

O termo acionista é dado a quem possui ações de uma empresa, e essa ação pode ser vendida em qualquer momento pelo acionista na bolsa de valores. (PITZER, 2018). Ao investir no mercado de ações, esperasse um rendimento a longo prazo, sendo este um dos princípios que difere um investidor de um *trader*.

(CERBASI, 2009) aponta que investidores compram o que as ações representam: um pedaço da propriedade de alguma empresa, com o intuito de se tornarem de fato um coproprietário. E, mesmo que o mercado de ações oscile, o investidor mantém o investimento a longo prazo baseado em seus estudos e conhecimentos da empresa em que se tornou um acionista. Já o trader não tem interesse em ser coproprietário da empresa, o foco do trader é o preço das ações, e por isso são conhecidos como pessoas que realizam transações de compra e venda de ativos na bolsa de valores com objetivo de ganhos a curto prazo.

Para comprar ações é necessário se cadastrar em uma corretora de valores, que é o agente autorizado a operar as ordens de compra e venda dos clientes na Bolsa de Valores. O preço da ação será determinado pela oferta e demanda do papel, ou seja, quando houver mais solicitações de compra do que de venda o preço da ação subirá, o inverso fará o preço cair. Este ambiente é regulado e fiscalizado pela Comissão de Valores Mobiliários (CVM), visando garantir a lisura e o correto procedimento na precificação das ações, para que o investidor não seja prejudicado.

Uma vez habilitado a comprar e vender ações, o investidor precisa saber

qual o momento ideal para realizar a operação. Existem dois tipos básicos de análise para determinar o valor justo de uma ação. A primeira é a análise técnica, ou gráfica, onde os preços das ações se movem de acordo com padrões repetitivos e identificáveis através de gráficos elaborados a partir de ferramentas estatísticas. O segundo tipo de análise é a fundamentalista cujo o foco do estudo são os dados econômico e financeiro da empresa. Projeções de receitas futuras são trazidas a valor presente para determinar a atratividade ou não de uma ação. Não existe um modelo de análise mais assertivo, sendo ambos úteis na tomada de decisão, e em alguns casos utilizados em conjunto. O mais importante é o investidor ter disciplina nos seus investimentos, determinando valores claros para compra e venda das ações, e mensurar quanto está disposto a deixar de ganhar (risco) e quando quer ganhar (retorno).

O retorno financeiro oriundo da posse de ações é, geralmente, obtido de três formas: com o aumento do valor de mercado das ações, com os dividendos (parte do lucro de uma empresa distribuídos em dinheiro), sem cobrança de impostos e com os juros sobre o capital próprio, com cobrança de 15% de imposto de renda sobre o valor a receber. Esses valores a serem distribuídos aos acionistas dependem de uma série de fatores que envolvem o desempenho da empresa.

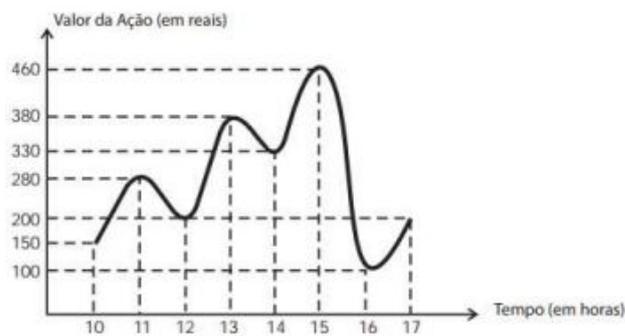
No que tange a ações, outro ponto importante são os índices do mercado. Eles servem como balizamento para o desempenho de uma carteira de ações. O mais conhecido deles, o Ibovespa, é composto pelas ações mais representativas em termos de volume de negociação. No entanto, existem diversos outros índices que, geralmente, representam ou um setor (ICON- Índice do Setor de Consumo) ou uma estratégia (SMLL-Índice Small Caps).

Índice IBOVESPA

O IBOVESPA, principal índice de ações da Bolsa de Valores de São Paulo, é o resultado de uma carteira de ativos que objetiva representar um indicador do desempenho médio das cotações dos ativos de maior negociabilidade e representatividade do mercado de ações brasileiro.

Exemplo 1.11 (*ENEM/2012*) *O gráfico fornece os valores das ações da empresa XPN, no período das 10 às 17 horas, num dia em que elas oscilaram acentuadamente em curtos intervalos de tempo.*

Figura 6 – Questão ENEM 2012



Fonte: Inep.2023

Neste dia, cinco investidores compraram e venderam o mesmo volume de ações, porém em horários diferentes, de acordo com a seguinte tabela.

Figura 7 – Questão ENEM 2012

Investidor	Hora da Compra	Hora da Venda
1	10:00	15:00
2	10:00	17:00
3	13:00	15:00
4	15:00	16:00
5	16:00	17:00

Fonte: Inep, 2023

Em relação ao capital adquirido na compra e venda das ações, qual investidor fez o melhor negócio?

Resolução: Considere a Tabela 6 abaixo.

Tabela 6 – Resolução Questão ENEM/2012

Investidor	1	2	3	4	5
Compra	R\$150	R\$150	R\$380	R\$460	100
Venda	R\$460	R\$200	R\$460	R\$100	R\$200
Lucro_ (Venda-Compra)	R\$310	R\$50	R\$80	R\$-360	R\$100

Fonte: Autor

E, portanto, o investidor que fez o melhor negócio foi o investidor 1, uma vez que obteve maior lucro em suas negociações.

Fundos de Investimentos

Os fundos de investimento funcionam como uma forma de investimento coletivo, onde diversos investidores aplicam seu dinheiro em um fundo gerido por uma equipe de gestores profissionais. Essa equipe é responsável por selecionar e gerir os ativos financeiros que compõem o portfólio do fundo.

Os investidores adquirem cotas do fundo de acordo com o valor que desejam investir, e recebem uma parte proporcional dos rendimentos obtidos pelo fundo. Os rendimentos podem vir tanto da valorização dos ativos financeiros quanto dos rendimentos gerados por eles, como juros, dividendos e aluguéis.

Os fundos de investimento podem ser classificados de acordo com o tipo de ativos financeiros em que investem, como ações, renda fixa, imóveis, entre outros. Também podem ser classificados de acordo com a estratégia de investimento adotada, como fundos de curto prazo, longo prazo, multimercado, entre outros.

Alguns dos principais benefícios dos fundos de investimento são a diversificação de risco, a possibilidade de investir em ativos financeiros que talvez não estariam acessíveis individualmente e a gestão profissional do dinheiro investido. No entanto, é importante lembrar que todo investimento possui riscos, e que é fundamental estudar bem as opções disponíveis antes de investir.

A diversidade dos tipos de fundos é grande, oferecendo estratégias para todo tipo de investidor. Existem fundos de ações, fundos de renda fixa, fundos cambiais, fundos imobiliários, fundos multimercados, fundo de índices de ações (ETFs).

Exemplo 1.12 *O BOVA11 é um fundo de índice que apresenta rentabilidade semelhante ao Ibovespa (IBOV), que é o principal indicador de desempenho das ações negociadas na B3 e reúne as empresas mais importantes do mercado de capitais brasileiro. Ricardo decidiu comprar R\$1.500,00 na ETF do BOVA11, e após 2 anos, revendeu-as por R\$1.658,25. Nessas condições, determine a taxa anual aproximada de rendimento desse fundo*

Resolução: Vamos determinar inicialmente a taxa I de rendimento total nesse período de 2 anos. Para isso, faremos a razão do valor das ações vendidas e compradas,

$$\frac{1.658,25}{1.500} = 1,1055 = 110,55\%$$

I de rendimento total nesse período é

$$I = 110,55\% - 100\% = 10,55\%.$$

Agora, chamando i da taxa média anual de rendimento desse fundo, temos para o tempo $t = 2$ anos, $(1+i)^2 = 1 + I$, daí

$$(1+i)^2 = 1 + 10,55\% \implies (1+i)^2 = 1,1055 \implies 1+i \approx 1,0514 \implies i \approx 5,14\%$$

No livro intitulado *Pai Rico, Pai Pobre*, escrito por Robert T. Kiyosaki (KIYOSAKI; LECHTER, 2000), afirma-se que um ativo é tudo o que pode gerar renda a quem o possui, enquanto um passivo refere-se ao que gera despesa.

Dentro da ampla definição de ativos, destacam-se os ativos financeiros, que são direitos econômicos tendo valor definido contratualmente. O próprio dinheiro pode ser visto desta forma. Alguns ativos financeiros bem conhecidos são as ações negociadas na bolsa de valores, os títulos de renda fixa e os fundos imobiliários. Estes ativos são os elementos de conjuntos chamados de **carteiras de investimentos**, ou **portfolios**.

Para formar uma carteira de investimentos, vamos supor que sejam adquiridos ativos financeiros numerados por $1, \dots, n$. Denotamos por $P_1(0), \dots, P_n(0)$ os respectivos preços pagos por cada ativo e por $\alpha_1(0), \dots, \alpha_n(0)$ as respectivas quantidades de títulos adquiridos de cada ativo.

Assim, o valor investido no ativo i é $\alpha_i(0)P_i(0)$ e o valor total da carteira no tempo 0 é

$$V(0) = \alpha_1(0)P_1(0) + \dots + \alpha_n(0)P_n(0). \quad (1.4)$$

Sendo $x_i(0)$ o percentual do capital inicial alocado no ativo i , temos que

$$x_i(0) = \frac{\alpha_i(0)P_i(0)}{V(0)} \quad (1.5)$$

Esse percentual é chamado de peso do ativo i na carteira. Formalmente, a carteira de investimentos (em $t = 0$) composta por estes ativos é definida como o vetor dos pesos

$$X(0) = (x_1(0), \dots, x_n(0)), \quad (1.6)$$

e será assumido que $x_1(0) + \dots + x_n(0) = 1$, isto é, o valor total da carteira representa o valor investido.

Exemplo 1.13 *Para formar uma carteira de ações, um investidor comprou 300 ações de Petrobrás S.A, cujo código de negociação (ticker) é PETR4, ao preço de R\$27,54 cada uma, 200 ações do Banco Itaú (ITUB4), a R\$ 23,90 e 800 ações da fabricante de ônibus Marcopolo (POMO4), a R\$ 2,62 cada. Numerando os ativos por 1, 2 e 3, respectivamente, tem-se, segundo a Equação 1.4 $V(0) =$*

$300 \cdot 27,54 + 200 \cdot 23,90 + 800 \cdot 2,62 = R\$15.138,00$. Conforme a Equação 1.5, temos os pesos

$$\begin{aligned}x_1(0) &= \frac{300 \cdot 27.54}{15.138.00} \approx 54.58\%, \\x_2(0) &= \frac{200 \cdot 23.90}{15.138.00} \approx 31.58\%, \\x_3(0) &= \frac{800 \cdot 2.62}{15.138.00} \approx 13.84\%.\end{aligned}$$

Portanto, a carteira é $X(0) = (0,54580, 31580, 1384)$. Note que $0,5458 + 0,3158 + 0,1384 = 1$.

Carteiras de investimentos são dinâmicas. Além da possível variação de preço dos ativos e a consequente mudança no valor da carteira, podem ocorrer também movimentações financeiras que naturalmente vão alterar sua formação. Por exemplo, o investidor pode vender ou comprar algum ativo, alocar mais capital e ainda pode ocorrer o pagamento de proventos, também chamados de rendimentos.

Em se tratando de ações, os proventos podem ser dividendos ou juros sobre o capital próprio pagos pela empresa aos seus acionistas. No caso de alguns títulos de dívida pública, podem ser pagos juros antes do vencimento dos títulos. Fundos imobiliários pagam mensalmente valores referentes a alugueis de imóveis que tenham participação. Esses valores são depositados na conta corrente da corretora onde o investidor possui seus ativos.

Uma vez formada uma carteira de investimentos, acompanhar sua rentabilidade regularmente ao longo do tempo é algo indispensável, principalmente porque permite comparar o desempenho com outras carteiras e reavaliar regularmente sua composição.

O caso mais simples é de um único investimento em um período também único, no qual essencialmente deve-se calcular uma variação percentual de preço, considerados proventos.

Assim, sendo r_t e P_t o **retorno simples** e o **preço** no tempo t :

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

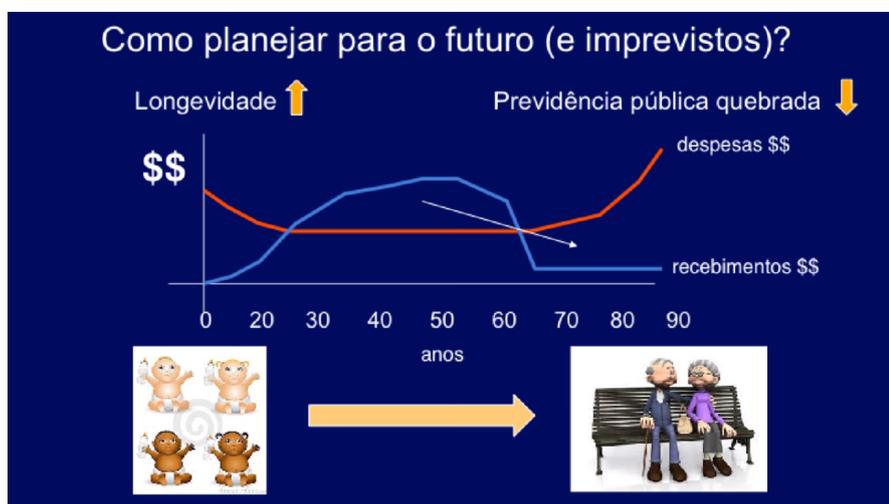
Previdência

Existem, basicamente, dois tipos de previdência, a social e a privada. A Previdência Social funciona basicamente como um seguro para a pessoa que contribuiu, durante um período mínimo, para o Instituto Nacional de Seguridade Social - INSS e deixa de atuar no mercado de trabalho em decorrência de idade, doença ou invalidez. Para ter direito, o trabalhador paga uma contribuição mensal

durante determinado tempo, que depende do tipo de aposentadoria escolhido ou tipo trabalho exercido. A soma de todas as contribuições dos cidadãos forma um fundo com a finalidade de pagar ao contribuinte quando ele para de trabalhar e se aposenta.

Com o aumento da expectativa de vida das pessoas, é cada vez mais comum ouvir se falar da importância de poupar e investir pensando em manter o mesmo padrão de vida no futuro. A alternativa de maior procura com este objetivo é a previdência privada em que o contribuinte busca complementar sua aposentadoria pelo INSS. Neste caso, o poupador paga uma mensalidade para uma instituição financeira administrar, com o propósito de receber um benefício no momento que deixar de trabalhar e perder a remuneração que possui enquanto está trabalhando.

Figura 8 – Planejar para o Futuro



Fonte: Adaptado

2 O USO DE RECURSOS COMPUTACIONAIS PARA O ENSINO DE MATEMÁTICA

Com o avanço da tecnologia e a popularização dos dispositivos eletrônicos, o uso de recursos computacionais no ensino da matemática tornou-se cada vez mais frequente e importante. Os recursos computacionais podem ser usados para complementar e enriquecer as aulas, tornando-as mais interativas, dinâmicas e atraentes para os estudantes.

(MORÁN, 2015) explica que a transmissão das informações pelos professores fazia sentido quando o acesso à informação era difícil, mas que “com a internet e a divulgação aberta de muitos cursos e materiais, podemos aprender em qualquer lugar, a qualquer hora e com muitas pessoas diferentes”. Dessa forma, evidenciando o fácil acesso a fórmulas e conceitos, os objetivos das aulas de matemática podem expandir, visando a formação do aluno para a interpretação dos dados encontrados, para o desenvolvimento da criatividade, da capacidade de falar em público, entre outros. Os professores por meio de novas práticas ou metodologias conseguem instigar, motivar e incentivar seus alunos para o desenvolvimento do conhecimento matemático de forma gradual ou progressiva. Desde o primeiro contato com a matemática, um mundo de possibilidades pode se abrir, e assim, a busca do conhecimento nesta disciplina pode significar descoberta e construção do saber.

Para isso, o professor precisa adaptar as metodologias para acompanharem os objetivos pretendidos. (MORÁN, 2015) indica que

se queremos que os alunos sejam proativos, precisamos adotar metodologias em que os alunos se envolvam em atividades cada vez mais complexas, em que tenham que tomar decisões e avaliar os resultados, com apoio de materiais relevantes. Se queremos que sejam criativos, eles precisam experimentar inúmeras novas possibilidades de mostrar sua iniciativa.

Nessa perspectiva, a utilização de softwares na elaboração de atividades pode ser importante para a obtenção dos objetivos, por possibilitar que os alunos

encontrem variados caminhos na busca por soluções; por esse tipo de atividade admitir diferentes soluções; por permitir que, a cada investigação, sejam exploradas novas estratégias de resolução; por ser qualitativamente diferente de uma atividade baseada no uso de lápis e papel; e por ser exploratória do ponto de vista educacional (BORBA; SILVA; GADANIDIS, 2018). Assim, o aluno é convidado a explorar novas possibilidades e tem espaço favorável para desenvolver as habilidades pretendidas. Dessa forma também são propiciados caminhos para formação de conjecturas, realização de testes, refinamento de conjecturas e familiarização com notações, como indicado por (BORBA; SILVA; GADANIDIS, 2018). Esses são elementos imprescindíveis para o estudo de matemática, para o desenvolvimento do pensamento lógico e utilização de conceitos matemáticos. Em contrapartida, é criticado o uso de muitos exercícios encontrados em livros didáticos pois, além de já terem as respostas publicadas, são facilmente resolvidos por softwares matemáticos disponíveis de forma online. Conforme (BORBA; SILVA; GADANIDIS, 2018) o trabalho com projetos é possibilidade para uma sala de aula que aceite internet e faz mais sentido na sociedade atual em que os alunos estão interagindo e se desenvolvendo. Atualmente, o uso de recursos computacionais tem ocupado um lugar de destaque dentro da sociedade e, principalmente, no ensino de matemática, uma vez que possibilita que uma disciplina que até algum tempo era ensinada de maneira estática, adquira dinamismo, passando a ser mais atrativa para os estudantes, conforme aponta Zarpelon (ZARPELON et al., 2016) . No início do desenvolvimento tecnológico, Papert (1928-2016) desenvolveu a teoria na qual o computador é utilizado como meio para a obtenção do conhecimento. Sua teoria foi denominada como Teoria do Construcionismo, na qual o indivíduo com um mediador, cria pontes de conhecimento por meio do computador (PAPERT; HAREL, 1991). Diante do atual contexto em que as tecnologias se fazem presentes no cotidiano, o uso de recursos tecnológicos, computadores, tablets e softwares podem vir a ser um grande aliado para o professor em tempos tão tecnológicos conforme mostram os Parâmetros Curriculares Nacionais (2017):

Contudo, também é imprescindível que a escola compreenda e incorpore mais as novas linguagens e seus modos de funcionamento, desvendando possibilidades de comunicação (e também de manipulação), e que eduque para usos mais democráticos das tecnologias e para uma participação mais consciente na cultura digital. Ao aproveitar o potencial de comunicação do universo digital, a escola pode instituir novos modos de promover a aprendizagem, a interação e o compartilhamento de significados entre professores e estudantes. (BRASIL, 2017, p.59).

A respeito do uso dos softwares educacionais, é possível encontrar nos Parâmetros Curriculares Nacionais (2017) a seguinte referência:

Utilizar processos e ferramentas matemáticas, inclusive tecnologias digitais disponíveis, para modelar e resolver problemas cotidianos, sociais e de outras áreas de conhecimento, validando estratégias e resultados. (BRASIL, 2017, p.265).

Assim, a teoria construcionista de Papert, o professor sendo o mediador e o computador um meio para obter conhecimento, se tornam um meio a mais para o desenvolvimento cognitivo dos alunos.

É evidente que a utilização de computadores pode ser um dos métodos inovadores no ensino de matemática, pois “O computador pode ser usado como elemento de apoio para o ensino (banco de dados, elementos visuais), mas também como fonte de aprendizagem e como ferramenta para o desenvolvimento de habilidades” (BRASIL,1997, p.35). A matemática é uma das disciplinas que pode ser trabalhada com diferentes métodos de ensino com o uso da informática, devido a grande demanda de softwares educativos matemáticos existentes, compreende-se que “As dimensões da inovação tecnológica permitem a exploração e o surgimento de cenários alternativos para a educação e, em especial para o ensino e aprendizagem de Matemática.” (BORBA; SILVA; GADINIDIS, 2014,p.17). Possibilitando a interação entre alunos e professores, despertando o interesse do aluno para a disciplina de matemática e abrindo novos caminhos para a aprendizagem. É relevante também considerar a importância do desenvolvimento do pensamento computacional, que é indicado na Base Nacional Comum Curricular (BNCC) como um dos objetivos relacionados à área de matemática das séries finais do ensino fundamental e do ensino médio. Na BNCC (2018, p. 273) encontra-se o seguinte trecho

outro aspecto a ser considerado é que a aprendizagem de álgebra, como também aquelas relacionadas a números, geometria e probabilidade e estatística, podem contribuir para o desenvolvimento do pensamento computacional dos alunos, tendo em vista que eles precisam ser capazes de traduzir uma situação dada em outras linguagens, como transformar situações-problema, apresentadas em língua materna, em fórmulas, tabelas e gráficos e vice-versa.

Nesse contexto, a BNCC incluiu na Educação Básica conhecimentos, habilidades, atitudes e valores referentes ao pensamento computacional, ao mundo digital e à cultura digital. E define (2018, p. 474) que:

1. pensamento computacional: envolve as capacidades de compreender, ana-

- lisar, definir, modelar, resolver, comparar e automatizar problemas e suas soluções, de forma metódica e sistemática, por meio do desenvolvimento de algoritmos;
2. mundo digital: envolve as aprendizagens relativas às formas de processar, transmitir e distribuir a informação de maneira segura e confiável em diferentes artefatos digitais – tanto físicos (computadores, celulares, tablets etc.) como virtuais (internet, redes sociais e nuvens de dados, entre outros) –, compreendendo a importância contemporânea de codificar, armazenar e proteger a informação;
 3. cultura digital: envolve aprendizagens voltadas a uma participação mais consciente e democrática por meio das tecnologias digitais, o que supõe a compreensão dos impactos da revolução digital e dos avanços do mundo digital na sociedade contemporânea, a construção de uma atitude crítica, ética e responsável em relação à multiplicidade de ofertas midiáticas e digitais, aos usos possíveis das diferentes tecnologias e aos conteúdos por elas veiculados, e, também, à fluência no uso da tecnologia digital para expressão de soluções e manifestações culturais de forma contextualizada e crítica.

Especificamente para o Ensino Médio, a BNCC (2018, p. 474) orienta:

[...] dada a intrínseca relação entre as culturas juvenis e a cultura digital, torna-se imprescindível ampliar e aprofundar as aprendizagens construídas nas etapas anteriores. Afinal, os jovens estão dinamicamente inseridos na cultura digital, não somente como consumidores, mas se engajando cada vez mais como protagonistas. Portanto, na BNCC dessa etapa, o foco passa a estar no reconhecimento das potencialidades das tecnologias digitais para a realização de uma série de atividades relacionadas a todas as áreas do conhecimento, a diversas práticas sociais e ao mundo do trabalho. [...]

Portanto, o uso do computador na escola não deve se limitar apenas à função dos editores de texto ou de slides; os estudantes devem aprender a utilizá-lo como uma extensão das faculdades cognitivas e capacidades humanas. A sociedade contemporânea demanda um grande conhecimento tecnológico, não apenas em relação ao uso das tecnologias de maneira eficaz, mas à elaboração de soluções, seja para problemas cotidianos, seja para problemas complexos de qualquer natureza. Desse modo, destaca-se a importância de um ensino da Matemática aplicado à realidade e vinculado à utilização de tecnologias digitais. Seu uso pode facilitar e ampliar o processo de resolução de problemas, reforçando o raciocínio lógico, a formulação de hipóteses e a argumentação, além de inspirar os estudantes a apren-

der cada vez mais e de maneira significativa os conteúdos desta disciplina. Neste trabalho, em diferentes momentos, os estudantes e o professor têm a oportunidade de trabalhar com planilhas eletrônicas, softwares de construção de gráfico e de geometria dinâmica.

2.1 O Pensamento Computacional.

A expressão “pensamento computacional” surgiu em 2006, no artigo Computational Thinking, da pesquisadora Jeannette Wing. Nele, Wing relaciona o termo à resolução de problemas de maneira sistemática, decompondo um problema complexo em subproblemas e automatizando a solução, de forma que possa ser executada por uma máquina. O pensamento computacional se apoia em quatro pilares. São eles:

1. Decomposição: consiste em quebrar um problema em partes menores (subproblemas) ou etapas, de maneira que a resolução de cada uma das partes ou etapas resulta na resolução do problema inicial. Dessa maneira, um problema ou situação complexa podem ser resolvidos aos poucos, com estratégias e abordagens diversas.
2. Reconhecimento de padrões: ocorre ao se perceber similaridade da situação enfrentada com outra previamente resolvida, o que permite o reaproveitamento de uma estratégia conhecida. Esse reconhecimento de padrões pode se dar entre instâncias distintas de um problema ou dentro dele mesmo, quando há repetições de etapas ou padrões em sua resolução.
3. Abstração: no contexto do pensamento computacional, significa filtrar as informações e dados relevantes à resolução, eliminando dados desnecessários, permitindo uma modelagem do problema mais limpa e eficaz.
4. Algoritmo: a aplicação dos pilares anteriores pode facilitar o surgimento de um algoritmo, que é uma generalização da resolução e permite resolver toda uma família de problemas similares. Um algoritmo pode ser definido como uma sequência finita de passos cuja finalidade é resolver um problema ou executar uma tarefa.

É importante salientar que, dependendo do problema, nem todos os pilares serão necessários e estarão presentes. Além disso, para ensinar o pensamento computacional e trabalhar com ele em sala de aula, apesar de a intenção ser a implementação computacional de uma solução, não é necessário um computador. A todo momento realizamos tarefas, as quais são organizadas mentalmente de maneira

consciente ou inconsciente. Por exemplo, durante um dia qualquer pretende-se lavar roupas, fazer uma lista de compras, ir ao mercado, limpar e organizar a casa, preparar o almoço e o jantar. Aparentemente, será uma corrida contra o tempo para realizar todos os afazeres. De que maneira é possível concluir todas essas tarefas até o fim do dia? Podemos executá-las seguindo uma ordem de preferência. Entretanto, cabe lembrar que algumas tarefas podem possuir pré-requisitos. Vejamos: preparar o almoço requer que os ingredientes estejam disponíveis; limpar a casa requer produtos de limpeza em quantidade suficiente. Assim, uma opção de ordenação dessas tarefas inclui, primeiramente, fazer a lista de compras, e, em seguida, ir ao mercado, deixando para realizar as demais tarefas posteriormente. Pensar acerca das tarefas que serão realizadas nos leva a identificar e extrair informações relevantes, dividir o dia em momentos oportunos e coerentes com os pré-requisitos de cada tarefa e as ordenar para que tudo flua bem. Esse raciocínio e essa organização podem ser associados aos pilares do pensamento computacional, pois envolvem a abstração das situações, a decomposição das tarefas e a criação de um algoritmo, isto é, praticamente um passo a passo para realizar as tarefas. Além disso, o reconhecimento de padrões pode ser identificado na realização de tarefas como lavar roupas e cozinhar um alimento. Destaca-se, assim, o pensamento computacional como um conjunto de habilidades que viabiliza a modelagem e a automatização de resoluções de problemas, podendo ser estudado e aplicado sem, necessariamente, envolver um computador.

Figura 9 – Pilares do Pensamento Computacional



Fonte: Adaptado

Uma das maneiras de trabalhar o pensamento computacional proposta pela BNCC é por meio da Álgebra. Ao interpretar e elaborar algoritmos incluindo aqueles que podem ser representados por fluxogramas, os estudantes têm a chance

de desenvolvê-lo, sendo “capazes de traduzir uma situação dada em outras linguagens, como transformar situações apresentadas em língua materna, em fórmulas, tabelas e gráficos” (BNCC, p. 271).

O Pensamento Computacional é uma das competências tão necessárias, quanto saber ler e escrever, para o exercício pleno da cidadania no século XXI. Esse pensamento é essencial para que o indivíduo possa desenvolver diversas habilidades. Ressalta-se, ainda, a necessidade de se discutir com professores da educação básica sobre como eles podem desenvolver suas práticas pedagógicas, usando as novas tecnologias educacionais e também o Pensamento Computacional, para que possam auxiliar os alunos a desenvolver habilidades e competências tão necessárias no século XXI que estão diretamente relacionadas ao uso das tecnologias. (BARCELOS; SILVEIRA, 2012) Complementam que “incorporar o Pensamento Computacional à educação básica envolve a análise sistemática de sua potencial sinergia com outras áreas do conhecimento”. O pensamento computacional é uma abordagem para a resolução de problemas que envolve a decomposição do problema em partes menores, identificação de padrões e algoritmos, desenvolvimento e testes de soluções e, finalmente, a aplicação de tais soluções. Embora o pensamento computacional seja frequentemente associado à ciência da computação, ele pode ser aplicado a uma ampla variedade de disciplinas, incluindo matemática.

Na matemática, o pensamento computacional pode ser utilizado para abordar problemas de uma forma sistemática e algorítmica. Por exemplo, ao resolver um problema de matemática, o pensamento computacional pode ser usado para identificar padrões e regras, que podem então ser usados para criar um algoritmo que solucione o problema. Além disso, o pensamento computacional pode ajudar a simplificar problemas complexos, dividindo-os em problemas menores que possam ser resolvidos separadamente.

Alguns exemplos específicos de como o pensamento computacional pode ser aplicado à matemática incluem:

- Identificação de padrões: O pensamento computacional pode ajudar os alunos a identificar padrões em conjuntos de dados ou sequências matemáticas. Ao encontrar padrões, os alunos podem criar equações ou algoritmos que descrevam o padrão, tornando mais fácil a resolução de problemas futuros.
- Divisão de problemas em subproblemas: O pensamento computacional pode ajudar os alunos a dividir problemas matemáticos complexos em problemas menores e mais gerenciáveis. Isso pode tornar mais fácil a compreensão do problema como um todo e permitir que os alunos abordem o problema de

uma forma mais sistemática.

- Uso de algoritmos: O pensamento computacional pode ajudar os alunos a criar algoritmos para solucionar problemas matemáticos. Um algoritmo é uma sequência de etapas que pode ser seguida para resolver um problema. Ao criar algoritmos, os alunos podem reduzir a complexidade do problema e tornar mais fácil a solução.
- Visualização de dados: O pensamento computacional pode ajudar os alunos a visualizar dados matemáticos usando gráficos ou outras ferramentas de visualização. Ao visualizar dados, os alunos podem identificar padrões e tendências, o que pode levar a uma compreensão mais profunda dos conceitos matemáticos.

O pensamento computacional se encaixa nas aulas de matemática por diversos motivos. Segundo (ROCHA; BASSO; NOTARE, 2020) “ao pensar por procedimentos o sujeito precisa descrever o processo com precisão de forma que possa ser executado pela máquina [...] [e] quando escreve um procedimento, suas intuições tornam-se mais evidentes, permitindo a reflexão e aproximando-o do conhecimento formal”. De certa forma, o aluno precisa “explicar” para a o leitor de códigos o que é necessário ser feito e, ao partir a descrição em partes que possam ser entendidas, ele está estudando e desenvolvendo o seu próprio entendimento. Assim, da mesma forma que a matemática contribui para o pensamento computacional, o pensamento computacional também tem contribuições diretas para a aprendizagem de matemática. Segundo a CIEB, “pensamento computacional refere-se à capacidade de resolver problemas a partir de conhecimentos e práticas da computação, englobando sistematizar, representar e analisar”(CIEB, 2022).

Neste capítulo, exploramos a relevância do pensamento computacional e seus pilares no aprimoramento dos alunos. Esses elementos capacitam os estudantes a desenvolver habilidades cognitivas essenciais para enfrentar desafios complexos. Ao adquirirem domínio desses conceitos, os alunos estarão aptos a aplicar os pilares do pensamento computacional - decomposição de problemas, reconhecimento de padrões e abstração - no contexto da matemática financeira e na utilização da linguagem de programação Python.

3 INTRODUÇÃO À PROGRAMAÇÃO COM PYTHON

Nesse trabalho sugere-se a escolha da linguagem de programação Python, como ferramenta utilizada nesta experiência didática. Este capítulo faz uma introdução a programação, visando compreender o que é uma linguagem de programação e suas estruturas e resolver problemas usando uma linguagem de programação em python para sistematizar, representar e analisar conhecimentos de Matemática Financeira.

As principais referências usadas foram os livros (SUMMERFIELD, 2012; PYTHON, ; SILVA; SILVA, 2019; LUTZ; ASCHER, 2007)

3.1 Noções Básicas de Python

O Python é uma linguagem de programação muito intuitiva e de fácil aprendizado. Tal linguagem possui uma sintaxe concisa e clara, além de contar com muitas bibliotecas para análise de dados, tornando-a assim a linguagem mais adequada para machine learning (aprendizado de máquina). Para Silva e Silva (2019, p. 59)(SILVA; SILVA, 2019):

Uma linguagem simples e elegante voltada para o desenvolvimento ágil. O Python segue algumas metodologias de desenvolvimento como, RAD (Rapid Application Development -Desenvolvimento rápido de aplicações). É uma metodologia focada na redução dos desperdícios e, melhorando assim a qualidade do código, diminuindo tempo, e custos de produção.

Para se inserir o Python como uma metodologia de ensino da matemática na Educação Básica, é preciso conhecermos o básico, termos alguns saberes pertinentes a máquina. Vamos começar falando sobre plataforma de desenvolvimento colaborativo Google Colab.

O Jupyter Notebook é um dos principais ambiente para rodar o python que oferece uma abordagem de Notebooks, com um visual simples e muito fácil de utilizar. Ele é aberto no navegador, mas funciona localmente.

O Google Colab é uma customização do Jupyter Notebook que roda no

Google Cloud.

Uma diferença entre o Google Colab e o Jupyter Notebook é a facilidade de utilizá-lo sem a necessidade de instalar algo em seu computador. Ele roda diretamente da nuvem, bastando acesso à internet e uma conta Google. É um ambiente muito amigável e logo ao abrir pela primeira vez ele te dá as boas-vindas com um pequeno tutorial de utilização.

Para facilitar a utilização pelos alunos, sugere-se a utilização do Google Colaboratory (ou somente Colab), que lê e executa códigos em linguagem Python e funciona de forma online.

O Google Colab usa um ambiente virtual que é executado em um servidor remoto e é acessado por meio do navegador do usuário. Quando o usuário executa uma célula de código, o código é enviado para o servidor remoto, onde é interpretado e executado. O resultado é então enviado de volta para o navegador do usuário para ser exibido.

Além disso, o Google Colab oferece uma ampla variedade de bibliotecas pré-instaladas para aprendizado de máquina e ciência de dados, como Pandas e Matplotlib. Isso permite que os usuários implementem rapidamente soluções para problemas de aprendizado de máquina e análise de dados.

A figura 10 mostra a tela do ambiente Google Colab - é necessário ter uma conta Google para utilizá-lo.

Figura 10 – Ambiente Google Colab



```

Desconto.ipynb
Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Todas as alterações foram salvas
Comentário Compartilhar
RAM Disco
+ Código + Texto
1 preco=float(input("Qual o preco original do produto?"))
2 desconto=float(input("Qual o desconto em percentual?"))
3 precocomdesconto=preco*(1-desconto/100)
4 print("Preco com desconto é ", precocomdesconto)
5
Qual o preco original do produto?80
Qual o desconto em percentual?20
Preco com desconto é 64.0

```

LinK:

<<https://tinyurl.com/IntrodPython>>

Exemplos de Comando Básicos com o Python

O interpretador do Python funciona como uma calculadora e, desse modo, podemos utilizar as operações matemáticas de maneira simples. Utilizamos, em

Python, os operadores $+$, $-$, e $/$ de maneira usual e os parênteses $(())$ para agrupar as expressões.

Os **operadores aritméticos** são: adição ($+$), subtração ($-$), multiplicação ($*$), divisão inteira ($//$), divisão real ($/$) e resto da divisão inteira ($\%$).

Figura 11 – Operadores Aritméticos

Símbolo	Comando	Retorna:
$+$	$a + b$	A soma
$-$	$a - b$	A subtração
$*$	$a * b$	A multiplicação
$/$	a/b	A divisão
$**$	$a ** b$	A potência
$//$	$a//b$	O quociente em \mathbb{Z}
$\%$	$a\%b$	A resto da divisão em \mathbb{Z}

Obs₁.: Para ficar mais legível, as linhas de código serão antecedidas de `>>>` ou... para blocos indentados e as saídas não serão precedidas de nenhum símbolo.

Exemplo 3.1 *Exemplo simples de operações matemáticas em python.*

```

1 >>> 2 + 3
2     5
3 >>> (40 - 4 * 5) / 2
4     10.0
5 >>> 15 / 7
6     2.142857142857143 # Divisão sempre retorna um ponto flutuante.
7 >>> 5 * 2
8     25

```

Obs₂.: Comentários em Python são introduzidos pelo caractere `#` e se estende até o final da linha física. Utilizando o caractere `/`, teremos sempre como retorno um ponto flutuante (*float*). Em programação e computação, o ponto flutuante é um tipo de dado numérico usado para representar números reais, ou seja, números que possuem casas decimais, e para fazer uma divisão e receber um inteiro devemos utilizar o operador `//` e para calcular o resto de uma divisão devemos usar o `%`.

Para obter como resultado um número inteiro na divisão devemos operar com números inteiros, caso queiramos como resultado ponto flutuantes, devemos operar com pontos flutuantes.

Operadores relacionais são: maior que (`>`), maior que ou igual a (`>=`),

menor que ($<$), menor que ou igual a ($<=$), igualdade ($=$), desigualdade ($!=$)

Figura 12 – Operadores Relacionais

Símbolo	Comparação (Pergunta)
$==$	igualdade
$!=$	diferente
$>$	maior do que
$<$	menor do que
$>=$	maior ou igual
$<=$	menor ou igual

Operadores booleanos são: conjunção (and), disjunção (or) e negação (not).

O que difere o Python de uma calculadora convencional, é o algoritmo, que quando escrito na linguagem do Python, poderá ser executado ao toque de um enter e ser adaptado para outros problemas, como já mencionado no reconhecimento de padrões no pensamento computacional. Além disso, o algoritmo pode ter quantas linhas se queira, podendo assim o aluno resolver uma expressão numérica contendo tantas quantas forem as incógnitas necessárias.

Link:

<https://tinyurl.com/operadores12345>

Variáveis

Variáveis são formas de armazenar dados para usá-los posteriormente. Para definir uma variável no Python devemos digitar o seu nome, o sinal de igualdade e o valor a ser armazenado.

As Variáveis devem ser iniciadas por letras minúsculas. O sinal ($=$) é usado para atribuir um valor a uma variável, por exemplo se criarmos a variável "a" e atribuirmos o valor 10, isto é, $a = 10$.

Obs.: Dentro do Python a igualdade matemática é representada por dois sinais de igualdade, isto é, $==$.

Em Python não há declaração de variáveis, sendo os tipos inferidos em tempo de execução. Os principais tipos de variáveis:

Numéricas: Quando armazena números inteiros (int) ou pontos flutuantes (float), isto é, números não inteiros.

String: É uma sequência de caracteres que deve ser delimitada por aspas simples ('...') ou aspas duplas ("...") possibilitando a utilização de textos no Python.

```
Exemplo 3.2 1 >>> Texto = "Python na Escola"
                2 >>> Texto
                  'Python na Escola'
```

Podemos concatenar strings com o operado (+) e repeti-las com o operador (*).

```
1 >>> a = 'Eu'
2 >>> b = 'Tu'
3 >>> 3*(a+b)
4 'EuTuEuTuEuTu'
```

Podemos também indexar (subscrever) e fatiar strings utilizando índices que da esquerda para a direita se inicia do 0 e da direita para esquerda inicia de -1.

```
Exemplo 3.3 1 >>> palavra = 'Matemática'
                2 >>> palavra [0]
                  'M'
                4 >>> palavra[-1]
                  'a'
                6 >>> palavra [0 : 3]
                7 'Mat'
```

As strings do Python são imutáveis, isto é, atribuir a uma posição indexada resulta em um erro. Desse modo, em caso de necessidade de uma string diferente temos que criar uma nova.

A função len() retorna o comprimento da string.

```
Exemplo 3.4 1 >>> palavra = 'Matemática'
                2 >>> len (palavra)
                3          10
```

Link:

<https://tinyurl.com/variaveisEdados>

Entrada e Saída de Dados

- Entrada de dados é realizada com o comando: `<varstring>=input(<mensagem>)`, ou seja, sempre o resultado de uma leitura será uma string (cadeia). Para conversões para os tipos inteiro e real devem ser utilizados `int(<cadeia>)` ou `float(<cadeia>)`, respectivamente.

A função **input** é destinada para interagir com o usuário, isto é, solicita dados que podem ter o formato numérico ou de string. O programa para e espera a digitação da informação seguida do ENTER

- Saída de dados é realizada com o comando: `print(<lista de variáveis ou literais>)`
- Atribuições são realizadas com o comando: `<variável> = <expressão>`. O sinal de igual pode ser substituído por `+=`, `-=`, `*=`, `/=` que correspondem a uma auto operação. Por exemplo: `i += 1` corresponde a `i = i + 1`

Vejamos exemplos de variáveis numéricas no Python:

Exemplo 3.5 *Veja o exemplo simples de um algoritmo em Python que solicita seu nome e faz uma saudação.*

```
1 nome = input("Digite seu nome : ") # Solicita ao usuário que digite seu nome
  e armazena na variável "nome"
2 print ("Muito prazer %s , bons estudos." % nome )
3 Digite seu nome : Arlinda
4 ... Muito prazer Arlinda , bons estudos
```

Na versão do Python utilizada nesse trabalho, o comando **input(...)** sempre entrega um valor do tipo **string**. Dessa maneira, caso queiramos utilizar a entrada como um número inteiro, precisamos utilizar um comando junto com a entrada, convertendo o valor do tipo **string** para o tipo **int**.

Os exemplos a seguir destacam a situação descrita acima:

Exemplo 3.6 *Entrada sem conversão*

```
1 a= input ("Digite o primeiro número : ")
2 b= input ("Digite o segundo número : ")
3 print (a +b )# O print retorna a concatenação de 40 com 12
4 Digite o primeiro número : 40
5 Digite o segundo número : 12
6 ... 4012
```

Exemplo 3.7 *Entrada de dados com conversão*

```

1 a = int( input ( " Digite o primeiro número : " ) )
2 b =int( input ( " Digite o segundo número : " ) )
3 print (a +b) # O print nessa caso retorna a soma de 40 com 12
4 Digite o primeiro número : 40
5 Digite o segundo número : 12
6 ... 52

```

Observe que a conversão dos dados de entrada no código do exemplo anterior permite que o Python entenda 40 e 12 como números.

Outro comando para a conversão da cadeia de caracteres lida na entrada é o `float(...)`, cujo funcionamento segue a mesma lógica do comando `int(...)`, mas converte o valor de entrada para o tipo float (número real).

Exemplo 3.8 *Veja o exemplo de um algoritmo em Python que recebe um número real da entrada e devolve o seu dobro como saída.*

```

1 | numero = float(input("Digite um número real: "))
2 | dobro = numero * 2
3 | print("O dobro do número é: ", dobro)

```

Se o usuário digitar, por exemplo, `-7.8`, a saída produzida será:

```

Digite um número real: -7.8
O dobro do número é: -15.6

```

Note a utilização do comando `print(...)` no exemplo acima. Quando queremos que o valor de uma variável apareça no texto que estamos escrevendo para o usuário como saída, o texto vai entre aspas e, separado dele por uma vírgula, inserimos o nome da variável. Caso queiramos que haja texto antes e depois, a linha três deverá ficar assim:

```

3 |print("O dobro do número",numero,"é:",dobro)

```

A interação com o usuário ficaria assim:

```

Digite um número real: -7.8
O dobro do número -7.8 é: -15.6

```

Link:

<https://tinyurl.com/EntradaDados>

Ferramentas de Controle de Fluxo

As ferramentas de controle fluxo possibilita condicionar a execução de comandos a um valor lógico (falso ou verdadeiro).

O comando **if**, se em português, é utilizado para executar determinada ação somente quando a condição dada for verdadeira. Em alguns casos é necessário executar ações de forma alternativa e para isso utilizamos o comando **elif** que é a abreviação de **else if**, senão se em português, temos ainda, **else**, senão em português, usado executar determinada ação caso nenhuma das anteriores forem executadas.

Exemplo 3.9 Comandos *if*, *else* e *elif*

```

1 >>> idade = int (input ( "Quantos anos você tem ? : " ) )
2 >>> if idade < 18 :
3 ...     print ( "Você não é adulto." )
4 >>> elif idade >= 18 and idade < 65:
5 ...     print ( "Você é adulto." )
6 >>> else :
7 ...     print ( "Você é idoso" )

```

Link:

<https://tinyurl.com/ControleFluxo>

Os comandos **for** e **while** são métodos de iteração em Python, o comando **for** em Python itera sobre os itens de qualquer sequência, na mesma ordem em que aparece na sequência, por sua vez, o comando **while** repete a instrução do seu bloco enquanto o condicional de seu cabeçalho for verdadeiro.

Exemplo 3.10 Estrutura Enquanto - exibição de 1 a 10

```

1 i=1 # declarando a variável
2 while (i<=10): # executar enquanto x<=10
3     print(i,end=",")
4     i+=1 # o valor da variável i é incrementado em 1. O operador "+=" é
    uma forma abreviada de atribuição e adição, ou seja, é o mesmo que i = i + 1. Isso
    significa que a variável i aumenta em 1 a cada iteração do loop.
5 ... 1,2,3,4,5,6,7,8,9,10,

```

A variável **i** inicia com valor 1, a condição de parada incluiu a igualdade

(<=) e houve inversão na ordem dos comandos de exibição do valor de **i** e seu incremento em uma unidade.

No Python, uma indentação errada causa problema na lógica do processamento e, conseqüentemente, no resultado final. O end="," no comando print define um valor final, no caso uma vírgula.

Link:

<<https://tinyurl.com/ForWhile>>

A função **range()** gera progressões aritméticas, o que é muito útil caso se queira iterar sobre sequências numéricas. A notação do comando **range** é range (start, stop, step), em que :

1. start é o valor do parâmetro inicial, se não for fornecido será 0;
2. stop é o valor do parâmetro final;
3. step é o valor do parâmetro incremento, se não for fornecido será 1.

Exemplo 3.11 Estrutura range

```
1 >>>range ( 0 , 20 , 3)
2 >>>b = list (range ( 0 , 20 , 3 ) )
3 >>> p r i n t ( b )
4     [ 0 , 3 , 6 , 9 , 12 , 15 , 18 ]
```

```
1 >>>range ( 0, 50, 3)
2 >>>a = list (range (0 ,50 ,3))
3 >>> print (a)
4     [ 0,3,6,9,12,15,18,21,24,27,30,33,36,39,42,45,48 ]
```

```
1 >>>a = range (1,10)
2 >>> for i in range (1 ,10) :
3 ...     print (i)
4     1
5     2
6     3
7     4
8     5
9     6
10    7
11    8
```

Exemplo 3.12 *Estrutura Para (for)- exibição dos pares de 2 a 20*

Este código utiliza uma estrutura de controle de repetição PARA (com variável de controle) para exibir os números pares de 2 a 20.

```

1 for i in range(2,21,2):
2   print(i,end=",")
3 RESTART :
4 ... 2,4,6,8,10,12,14,16,18,20,
```

Nos exemplos seguintes mostraremos mais duas estruturas de controle de repetição do REPITA ATÉ QUE e REPITA < n > VEZES. A primeira é semelhante a estrutura ENQUANTO, mas com a inversão da condição de finalização. Já a estrutura REPITA < n > VEZES é uma forma abreviada da estrutura PARA utilizada para repetir a execução de < comandos > por < n > vezes. A contagem é realizada de forma implícita.

Exemplo 3.13 *Estrutura Repita até que:*

Vamos dar o exemplo com a exibição de 1 a 10.

```

1 i=0
2 while not(i==10):
3   i+ = 1
4   print(i,end=",")
```

A expressão `not(i == 10)` pode ser substituída por `(i != 10)`

Exemplo 3.14 *Repita n vezes: 1 a 10:*

```

1 for i in range(10):
2   print(i+1,end=",")
```

Resultado

```

3 1,2,3,4,5,6,7,8,9,10,
```

A função `range(10)` gera uma lista com 10 elementos (valores de 0 a 9). Por isso, a exibição dos valores $(i + 1)$ sucessores de i

Link:

<https://tinyurl.com/ForInRan>

Listas e Tuplas

Listas são estruturas de dados compostas usadas para agrupar itens. As listas (list) tem seus elementos separados por vírgula, entre colchetes.

Exemplo 3.15 a) Criando Listas

```
1 >>> primos = [2, 3, 5, 7, 11, 11, 13]
2 >>> primos
3 [2, 3, 5, 7, 11, 11, 13]
```

De modo análogo as strings, as listas podem ser fatiadas e concatenadas retornando uma nova lista contendo os elementos solicitados.

b) Selecionando Elementos

```
1 >>> primos = [2, 3, 5, 7, 11, 11, 13]
2 >>> primos[1 : 5] (# nesse caso, o código está selecionando os elementos da
3 [3, 5, 7, 11]
4 >>> primos + [17, 19, 23]
5 [2, 3, 5, 7, 11, 11, 13, 17, 19, 23]
```

(# nesse caso, o código está selecionando os elementos da posição 1 até a posição 4 da lista "primos"(lembrando que a contagem começa em 0).)

As listas diferem das strings, pois são mutáveis, isto é, há possibilidade de alteração de elementos individuais de uma lista.

c) Alterando Elementos

```
1 >>> primos = [2, 3, 5, 7, 11, 11, 13]
2 >>> primos [ 2 ] = 29
3 >>> primos
4 [2, 3, 29, 7, 11, 11, 13]
```

Podemos também adicionar novos elementos no final da lista, para isso, basta usar o método `.append()`

d) Adicionando elementos

```
1 >>> primos = [2, 3, 5, 7, 11, 11, 13]
2 >>> primos . a ppend ( 1 7 )
3 >>> primos
4 [ 2 , 3 , 5 , 7 , 1 1 , 1 1 , 1 3 , 1 7 ] .
```

Podemos criar listas contendo outras listas.

e) Listas contidas em listas

```
1 >>> p rimos = [ 2 , 3 , 5 , 7 , 1 1 , 1 1 , 1 3 ]
2 >>> l e t r a s = [ ' a ' , ' b ' , ' c ' ]
```

```

3 >>> lista = [ primos, 'casa', 'carro', letras ]
4 >>> lista
5 [ [ 2, 3, 5, 7, 11, 11, 13 ], 'casa', 'carro', [ 'a', 'b', 'c' ] ]

```

Link:

<https://tinyurl.com/CriandoListas>

As Tuplas *tuple* são sequências imutáveis, diferentemente da lista não há possibilidade de adicionar ou remover elementos. Uma tupla tem a forma de uma sequência de dados separados por vírgulas e podem vir ou não entre parênteses.

Exemplo 3.16 Tupla:

```

1 >>> t = 'casa', 10, 'olá!'
2 >>> t [ 2 ] # Observe que a contagem dos elementos de uma tupla começa do
3 'olá!'
4 >>> t
5 'casa', 10, 'olá!'

```

A diferença principal entre Listas e Tuplas é que os elementos de tuplas não podem ser alterados. Se tentarmos modificar um elemento de uma tupla teremos um erro.

Link:

<https://tinyurl.com/Tuplas>

3.1.1 Bibliotecas Científicas do Python

Uma biblioteca científica no contexto do Python refere-se a um conjunto de ferramentas e funcionalidades pré-programadas que são desenvolvidas para facilitar e acelerar a realização de tarefas relacionadas à análise de dados, visualização, cálculos matemáticos e científicos, modelagem estatística, aprendizado de máquina e outros aspectos da computação científica. No caso específico do Python, algumas bibliotecas científicas populares incluem NumPy, Pandas, Matplotlib entre outras. Essas bibliotecas fornecem um conjunto rico de recursos e funcionalidades para a comunidade científica, tornando o Python uma escolha popular para aplicações científicas e análise de dados.

NumPy é uma Biblioteca de Álgebra Linear para Python muito rápida pois tem seu core implementado em *C*. É a base de muitas outras bibliotecas,

como o Pandas que vamos ver a seguir.

Documentação oficial: <<https://numpy.org/doc/stable/>>

A classe array do Numpy é chamada de **ndarray** e oferece mais funcionalidades que a biblioteca array padrão do Python que trabalha apenas com matrizes unidimensionais.

Exemplo 3.17 *Importação*

```
1 >>> import numpy as np
```

Códigos-fonte Numpy

<<https://tinyurl.com/Numpy3>>

O **Pandas** é a mais completa biblioteca Python usada para manipulação, leitura e visualização de dados, sendo fundamental para se trabalhar com **análise de dados**.

Página Oficial: <<https://pandas.pydata.org/>>

Devemos importá-lo usando o código abaixo:

```
1 >>> import pandas as pd
```

Códigos-fonte Pandas

<<https://tinyurl.com/Pandas1234>>

<<https://tinyurl.com/PandasPratica>>

A **Matplotlib** é a principal biblioteca Python para a visualização de dados e plotagem de gráficos no Python. Ela oferece uma ampla gama de ferramentas para a criação de gráficos estáticos, animações e visualizações interativas.

Documentação Oficial: <<https://matplotlib.org/>>

Para utilizarmos essa biblioteca precisamos importar com o comando import da biblioteca matplotlib a parte de plotar gráficos pyplot utilizando o comando conforme abaixo:

Importação:

```
1 >>> import matplotlib.pyplot as plt
```

Códigos-fonte Matplotlib

<<https://tinyurl.com/matplotlib123>>

Módulos e Pacotes

Python vem com "baterias incluídas", ou seja contém ferramentas úteis para uma ampla variedade de tarefas.

Além disso, existe um amplo ecossistema de ferramentas e pacotes de terceiros que oferecem funcionalidades mais especializadas.

Para importar um pacote simplesmente utilizamos o comando **import**

Exemplo 3.18 *Vamos importar o pacote numpy.*

```
1 >>> import numpy
```

Importando podemos utilizar o pacote, conforme abaixo que ele retorna um array:

```
1 >>> numpy.ones(5)
```

```
2 array([1., 1., 1., 1., 1.])
```

Então essa função **ones** não estava disponível, mas a partir do momento que importamos o numpy a gente pode utilizar.

Exemplo 3.19 *Existe outra forma de importar que podemos dar uma abreviação ao módulo :*

```
1 >>> import numpy as np
```

```
1 >>> np.ones(5)
```

```
2 array([1., 1., 1., 1., 1.])
```

Podemos instalar outros pacotes que não estejam instalados no ambiente através do gerenciador de pacotes **PIP** como no exemplo a seguir com a instalação do pacote yfinance para operações financeiras.

Exemplo 3.20 *Instalando e importando pacote yfinance*

```
1 >>> ! pip install yfinance
```

Depois de instalado importamos:

```
1 >>> import yfinance as yf
```

Depois de importado podemos utilizar todos os métodos do pacote seguido do ponto (*yf .*), segundo exemplo a seguir :

Exemplo 3.21 *Download de cotações do índice bovespa*

```
1 >>> yf.download('^BVSP') (# O yfinance utiliza o acento agudo (^) para representar o código do índice financeiro da B3, conhecido como "Ibovespa").
```

Figura 13 – Cotação índice bovespa

1 yf.download('^BVSP')

[*****100%*****] 1 of 1 completed

Date	Open	High	Low	Close	Adj Close	Volume
1993-04-27	24.799999	25.400000	24.500000	24.500000	24.500000	0
1993-04-28	24.500000	24.600000	23.700001	24.299999	24.299999	0
1993-04-29	24.299999	24.799999	23.700001	23.700001	23.700001	0
1993-04-30	23.700001	24.200001	23.700001	24.100000	24.100000	0
1993-05-03	24.100000	24.400000	23.799999	24.100000	24.100000	0
...
2023-04-10	100822.000000	102196.000000	100819.000000	101847.000000	101847.000000	8148700
2023-04-11	101849.000000	106455.000000	101848.000000	106214.000000	106214.000000	18725700
2023-04-12	106218.000000	108277.000000	106217.000000	106890.000000	106890.000000	17345500
2023-04-13	106890.000000	107037.000000	106220.000000	106458.000000	106458.000000	12182900
2023-04-14	106458.000000	106701.000000	104934.000000	106279.000000	106279.000000	11501000

7417 rows x 6 columns

Fonte: Elaborada pelo autor

Link:

<https://tinyurl.com/ModulosPacotes>

4 PROPOSTAS DIDÁTICAS

Neste capítulo, apresentaremos uma sequência didática com atividades que exploram o potencial da linguagem Python e da matemática financeira no contexto educacional. A proposta inclui uma sequência estruturada, abrangendo a introdução à linguagem Python, matemática financeira com Python e o uso das bibliotecas NumPy, Pandas e Matplotlib. Os exemplos e atividades relacionados a esses temas estão disponíveis no anexo deste trabalho.

Primeiro, os alunos serão introduzidos à linguagem Python e ao ambiente Google Colab, proporcionando-lhes um primeiro contato com a programação e as ferramentas necessárias para implementar soluções em matemática financeira. Em seguida, serão apresentados os conceitos básicos de matemática financeira, como juros simples, juros compostos, valor presente e valor futuro, explorando sua relevância no cotidiano e suas aplicações práticas utilizando Python.

A partir desses fundamentos, os alunos serão desafiados a realizar cálculos financeiros utilizando Python, aprendendo a aplicar fórmulas e funções específicas da linguagem para resolver problemas relacionados à matemática financeira. Em seguida, terão a oportunidade de simular e comparar diferentes opções de investimentos de renda fixa e variável, utilizando Python no ambiente Google Colab, a fim de analisar e tomar decisões informadas.

Durante o curso, serão propostas atividades práticas com o computador, permitindo que os alunos desenvolvam habilidades práticas em Python e análise financeira. Além disso, serão fornecidos exercícios para reforçar os conceitos e as técnicas aprendidas em cada tópico. Essas atividades práticas e exercícios visam capacitar os alunos a aplicar os conceitos teóricos em situações reais e promover uma compreensão mais aprofundada dos temas abordados.

Ao final da sequência didática, os alunos serão convidados a realizar um projeto final de análise de dados financeiros, onde poderão aplicar todos os conhecimentos adquiridos ao longo do curso. Esse projeto permitirá que eles demonstrem suas habilidades na utilização do Python, das bibliotecas NumPy, Pandas e Matplotlib, além de aplicar a matemática financeira na análise de dados reais.

Nos planos de aula apresentados, buscamos explorar competências e ha-

habilidades específicas da Base Nacional Comum Curricular (BNCC) para o ensino de matemática. Como estabelece (BRASIL, 2017), habilidades são ideias que os alunos devem aprender para adquirir uma determinada competência, que por sua vez

“é definida como a mobilização de conhecimentos (conceitos e procedimentos), habilidades (práticas, cognitivas e socioemocionais), atitudes e valores para resolver demandas complexas da vida cotidiana, do pleno exercício da cidadania e do mundo do trabalho” (BRASIL, 2017, p. 8).

A seguir, descreveremos as competências e habilidades da BNCC, que serão trabalhadas através da execução dos planos de aula ao longo deste capítulo.

COMPETÊNCIA ESPECÍFICA 3 – “Utilizar estratégias, conceitos, definições e procedimentos matemáticos para interpretar, construir modelos e resolver problemas em diversos contextos, analisando a plausibilidade dos resultados e a adequação das soluções propostas, de modo a construir argumentação consistente” (BRASIL, 2017, p. 535).

COMPETÊNCIA ESPECÍFICA 5 - "Investigar e estabelecer conjecturas a respeito de diferentes conceitos e propriedades matemáticas, empregando recursos e estratégias como observação de padrões, experimentações e tecnologias digitais, identificando a necessidade, ou não, de uma demonstração cada vez mais formal na validação das referidas conjecturas"(BRASIL, 2017, p. 535).

Habilidades da BNCC:

- (EM13MAT203) “Aplicar conceitos matemáticos no planejamento, na execução e na análise de ações, envolvendo a utilização de aplicativos e a criação de planilhas (para o controle de orçamento familiar, simuladores de cálculos de juros simples e compostos, entre outros), para tomar decisões” (BRASIL, 2017, p. 534).
- (EM13MAT315) “Investigar e registrar, por meio de um fluxograma, quando possível, um algoritmo que resolve um problema.” (BRASIL, 2017, p. 537)
- (EM13MAT404) “Analisar funções definidas por uma ou mais sentenças (tabela do Imposto de Renda, contas de luz, água, gás etc.), em suas representações algébrica e gráfica, identificando domínios de validade, imagem, crescimento e decrescimento, e convertendo essas representações de uma para outra, com ou sem apoio de tecnologias digitais.” (BRASIL, 2017, p. 539)
- (EM13MAT405) “Utilizar os conceitos básicos de uma linguagem de progra-

mação na implementação de algoritmos escritos em linguagem corrente e/ou matemática.” (BRASIL, 2017, p. 539)

- (EM13MAT510) “Investigar conjuntos de dados relativos ao comportamento de duas variáveis numéricas, usando ou não tecnologias da informação, e, quando apropriado, levar em conta a variação e utilizar uma reta para descrever a relação observada.” (BRASIL, 2017, p. 541)

4.1 Sequência Didática: Introdução ao Python, Matemática Financeira com Python e Uso das Bibliotecas NumPy, Pandas e Matplotlib

Duração Total: (5 encontros com duração média de 4 horas)

Aula 1: Introdução ao Python

Descrição: Nesta aula introdutória, os alunos serão apresentados à linguagem de programação Python, seu ambiente de desenvolvimento e conceitos básicos de programação.

Roteiro das Atividades:

1. Apresentação do curso e dos objetivos de aprendizagem.
2. Introdução ao Python: sintaxe básica, tipos de dados, operadores e estruturas de controle.
3. Exercícios práticos para fixação dos conceitos básicos de programação.
4. Apresentação de recursos online e comunidades para suporte adicional.

Atividades Práticas com Computador:

1. Configuração do ambiente de desenvolvimento Python.
2. Execução de comandos Python no interpretador interativo.
3. Realização de exercícios práticos de programação usando Python.

Sugestão de Exercícios:

1. Escrever um programa que calcule e imprima a média de três notas.
2. Escrever um programa que converta uma temperatura em Celsius para Fahrenheit.
3. Escrever um programa que determine se um número é par ou ímpar.

Aula 2: Matemática Financeira com Python

Descrição: Nesta aula, os alunos serão introduzidos aos conceitos básicos de matemática financeira e aprenderão a aplicar esses conceitos usando a linguagem Python.

Roteiro das Atividades:

1. Revisão dos conceitos básicos de matemática financeira: juros simples, juros compostos, taxa de juros, capitalização.
2. Apresentação de fórmulas matemáticas para cálculos financeiros.
3. Uso de bibliotecas Python para cálculos financeiros: numpy finance.
4. Exemplos práticos de aplicação da matemática financeira em Python.

Atividades Práticas com Computador:

1. Instalação da biblioteca numpy finance.
2. Uso de funções da biblioteca numpy finance para cálculos financeiros.
3. Desenvolvimento de programas Python para cálculos financeiros.

Sugestão de Exercícios:

1. Calcular o montante final de um investimento a juros compostos.
2. Calcular a taxa de juros de um empréstimo com base no montante final, capital inicial e período.
3. Calcular o valor presente de um fluxo de caixa futuro.

Aula 3: Uso da Biblioteca Pandas

Descrição: Nesta aula, os alunos aprenderão a utilizar a biblioteca Pandas para análise de dados financeiros em Python. **Roteiro das Atividades:**

1. Introdução à biblioteca Pandas: estruturas de dados (Series e DataFrame), indexação, seleção e filtragem de dados.
2. Importação de dados financeiros a partir de arquivos (CSV, Excel) usando Pandas.
3. Manipulação e transformação de dados financeiros com Pandas. Visualização de dados financeiros usando Matplotlib.

Atividades Práticas com Computador:

1. Instalação da biblioteca Pandas.
2. Importação de dados financeiros e criação de estruturas de dados com Pand
3. Manipulação e análise dos dados utilizando as funcionalidades do Pandas.

Sugestão de Exercícios:

1. Importar um arquivo CSV contendo dados históricos de preços de ações, criar um DataFrame e realizar análises básicas, como média, máximo e mínimo.
2. Filtrar os dados para obter apenas os preços de fechamento de uma ação específica e plotar um gráfico de linha com a série temporal.
3. Calcular o retorno diário de uma ação e realizar uma análise descritiva dos retornos.

Aula 4: Uso da Biblioteca Matplotlib

Descrição: Nesta aula, os alunos aprofundarão seus conhecimentos na biblioteca Matplotlib para criação de gráficos mais complexos e personalizados.

Roteiro das Atividades:

1. Revisão dos conceitos básicos do Matplotlib: criação de figuras, adição de eixos, plotagem de gráficos simples.
2. Plotagem de gráficos mais avançados: gráficos de barras, histogramas, gráficos de dispersão.
3. Personalização dos gráficos: rótulos, títulos, legendas, cores, estilos.
4. Aplicação dos conhecimentos em visualização de dados financeiros.

Atividades Práticas com Computador:

1. Instalação da biblioteca Matplotlib.
2. Criação de diferentes tipos de gráficos utilizando o Matplotlib.
3. Personalização dos gráficos de acordo com os requisitos específicos.

Sugestão de Exercícios:

1. Plotar um gráfico de barras comparando o desempenho de diferentes ações ao longo de um período.
2. Criar um histograma para visualizar a distribuição dos retornos de um conjunto de ações.
3. Plotar um gráfico de dispersão para analisar a relação entre o preço de uma ação e seu volume de negociação.

Projeto Final: Análise de Dados Financeiros

Descrição: O projeto final permitirá que os alunos apliquem todos os conceitos e habilidades aprendidos ao longo do curso para realizar uma análise de dados financeiros completa.

Roteiro das Atividades:

1. Definição do escopo do projeto: seleção dos dados financeiros a serem analisados.
2. Importação dos dados utilizando o Pandas.
3. Limpeza e preparação dos dados para análise.
4. Realização de análises estatísticas e gráficas utilizando o Matplotlib.
5. Apresentação dos resultados da análise em um relatório ou apresentação.

Atividades Práticas com Computador:

1. Coleta dos dados financeiros relevantes.
2. Importação e preparação dos dados utilizando o Pandas.

3. Realização das análises e criação de visualizações utilizando o Matplotlib.
4. Documentação dos resultados e apresentação do projeto final.

Sugestão de Exercícios:

1. Realizar uma análise exploratória de um conjunto de dados financeiros, identificando padrões e tendências.
2. Aplicar técnicas de regressão para prever o preço futuro de uma ação com base em dados históricos.
3. Realizar uma comparação entre diferentes estratégias de investimento utilizando dados históricos.

Obs.: No anexo A, estão disponíveis Exemplos e Exercícios Sugeridos para os Planos de Aula relacionados à proposta didática apresentada neste trabalho. Cada exemplo e exercício possui um link de acesso para teste da execução no Google Colab.

Esses recursos adicionais fornecem aos alunos a oportunidade de praticar e aplicar os conceitos e habilidades aprendidos durante o curso. Os exemplos demonstram o uso prático das bibliotecas NumPy, Pandas e Matplotlib, além da linguagem Python, na análise financeira. Já os exercícios sugeridos permitem que os alunos consolidem seu conhecimento e desenvolvam suas habilidades práticas, resolvendo problemas financeiros específicos utilizando Python.

Ao acessar os exemplos e exercícios no Google Colab, os alunos poderão interagir com o código, realizar modificações e experimentar diferentes abordagens, promovendo uma compreensão mais profunda e uma aprendizagem ativa dos tópicos abordados.

Esses recursos complementares são uma oportunidade valiosa para os alunos colocarem em prática seus conhecimentos, explorarem cenários reais e se familiarizarem com as ferramentas e bibliotecas discutidas na sequência didática. Isso contribui para um aprendizado mais significativo e uma maior confiança na aplicação da programação e da análise financeira utilizando Python.

5 CONSIDERAÇÕES FINAIS

Ao realizar uma análise detalhada, percebe-se que a incorporação da Linguagem de Programação como uma ferramenta digital e tecnológica em sala de aula apresenta inúmeras vantagens para a formação educacional dos estudantes e complementa de maneira significativa o trabalho pedagógico do professor. A sua integração nas aulas de Matemática do Ensino Médio viabiliza a exploração de conceitos e estimula discussões sobre estratégias para a resolução de problemas.

Assim, o presente trabalho propõe aos professores de matemática a utilização da Linguagem de Programação no ensino básico, com a finalidade de se adequar as novas realidades e deixar o ensino de matemática mais significativo, visando combater o desinteresse dos alunos, visto que, temas que envolve tecnologia são agradáveis aos estudantes tornando as aulas mais atrativas e participativas.

De modo particular, o Python é uma linguagem de programação que facilita essa abordagem devido a sua sintaxe simples, sua clareza e objetividade. Sendo assim, infere-se que esta linguagem de programação específica possui um grande potencial pedagógico.

A composição desta dissertação promove uma assistência completa aos professores, sendo um tanto autossuficiente neste quesito, uma vez que oferece um tutorial com exemplificações, uma variedade de problemas e aplicações, bem como possibilidades reais de aplicação.

Portanto, ao concluir este trabalho, busca-se auxiliar os professores de Matemática do Ensino Médio na implementação de estratégias pedagógicas que envolvam o uso da Linguagem de Programação Python. Além disso, pretende-se motivá-los a explorar regularmente esse poderoso recurso digital e tecnológico. Acredita-se que a utilização consistente da linguagem de programação no contexto educacional contribua para aprimorar o ensino de matemática, facilitando a compreensão de conceitos e promovendo uma aprendizagem mais efetiva nessa disciplina.

ANEXO A Exemplos e Exercícios Sugeridos para os Planos de Aula

ANEXO A.1 Primeiros cálculos financeiros com Python

Uma aplicação de R\$ 50.000,00 investida no prazo de 180 dias resultou em um rendimento de R\$ 8.250,00. Qual a taxa anual correspondente a esta aplicação?

Solução em Python:

```
1 # Define as variáveis do problema
p = 50000 # valor principal investido
r = 8250 # rendimento obtido
t = 180/360 # prazo da aplicação em anos
2| # Calcula a taxa anual correspondente
taxa_anual = (r/p) * (1/t)
3|# Exibe o resultado
print(f"A taxa anual correspondente é de taxa_anual: . 2%")
```

RESTART :

taxa anual correspondente é de 33.00%

Exemplo ANEXO A.1 Calcule o montante de uma capitalização composta, cujo capital aplicado é de R\$ 1.000,00, a uma taxa de 4 % ao mês, durante 5 meses.

Solução:

```
1|capital = 1000 # capital inicial de R$ 1.000,00
2|taxa = 0.04 # taxa mensal de 4%
3|periodos = 5 # duração em meses
4|montante = capital * (1 + taxa)**periodos
5|print(f"O montante final é de R$ montante: . 2f")
6| RESTART :
7| O montante final é de R$ 1216.65
```

Exemplo ANEXO A.2 *Alice ganha R\$ 1.200,00 em seu novo emprego, com um aumento anual de 3,3% . João hoje ganha R\$ 900,00 e recebe um aumento anual de 6,8% . Em quantos anos o salário de João ultrapassará o de Alice ?*

Usaremos o **software Python** como ferramenta facilitadora na resolução.

Solução:

```
alice_salario = 1200
```

```
joao_salario = 900
```

```
anos = 0
```

```
while joao_salario <= alice_salario:
```

```
    alice_salario *= 1.033
```

```
    joao_salario *= 1.068
```

```
    anos += 1
```

```
print("Em", anos, "anos o salário de João ultrapassará o de Alice")
```

Está disponível em: <<https://tinyurl.com/PrimeirosCalculos>>

Exemplo ANEXO A.3 *ValorFuturo(FV)*

Qual seria o Valor Futuro (FV) se eu investisse R\$ 1000 por 1 ano, com um retorno de 5% ao ano?

$$FV = PV(1+r)^n$$

Solução em Python:

```
1 >>> PV = 1000
```

```
2 r = 0.05
```

```
3 n = 1
```

```
1 >>> FV = PV * ((1+r) ** n)
```

```
2 FV
```

```
3... 1050.0
```

Exemplo ANEXO A.4 *ValorPresente(PV)*

Qual seria o Valor Presente (PV), se eu tivesse que descontar R\$ 1.050 a uma taxa anual de 5% por um período de 1 ano?

$$PV = \frac{FV}{(1+r)^n}$$

Solução em Python:

```
1 >>> FV = 1050
```

```
2 r = 0.05
```

```
3 n = 1
```

```
1 >>> PV = FV/((1+r)**n)
```

2 PV

...1000.0

Link:

<<https://tinyurl.com/ValorDinheiro>>

Exemplo ANEXO A.5 *Um bem cujo preço a vista é de R\$ 2.000,00 será pago em 6 prestações mensais e iguais ao final de cada mês. Considerando que a taxa de juros compostos é de 3% ao mês, calcular o valor das prestações.*

Solução em Python :

```

1 >># Define as variáveis
2 preco_a_vista = 2000.00
3 num_prestacoes = 6
4 taxa_juros_mensal = 0.03
5# Calcula o valor da prestação
6 prestacao = (preco_a_vista * ((1 + taxa_juros_mensal) ** num_prestacoes)
* taxa_juros_mensal) / (((1 + taxa_juros_mensal) ** num_prestacoes) - 1)
7# Exibe o valor da prestação
8 print("Valor da prestação mensal: R$ % . 2f" % prestacao)

```

Exemplo ANEXO A.6 *Quem aplicou R\$ 1000,00 na caderneta de poupança em 2 de janeiro de 2015 viu seu dinheiro crescer cerca de 8,15% em 31 de dezembro de 2015 e obteve um saldo aproximado de R\$1081,50 (Selic acima de 8,5% ao ano, durante todo o ano, e TR anual de 1,80%). Considerando que a inflação de 2015 foi de 10,67%, houve uma perda real de 2,28% sobre o valor investido em um ano.*

```

1 # Definindo as variáveis
2 investimento = 1000.0 # valor investido
3 taxa_nominal = 0.0815 # taxa de juros nominal
4 taxa_inflacao = 0.1067 # taxa de inflação
5 taxa_selic = 0.085 # taxa SELIC anual
6 tr_anual = 0.018 # taxa TR anual
7 perda_real = 0.0 # inicializando a variável de perda real
8 # Calculando a taxa real de juros
9 taxa_real = (1 + taxa_nominal) / (1 + taxa_selic + tr_anual) - 1
10 # Calculando o valor final do investimento com juros simples
11 valor_final = investimento * (1 + taxa_nominal)
12 # Calculando o valor corrigido pela inflação
13 valor_corrigido = valor_final / (1 + taxa_inflacao)
14
15 # Calculando a perda real
16 perda_real = 1 - valor_corrigido / investimento
17 # Imprimindo o resultado
18 print("A perda real do investimento foi de {:.2%}".format(perda_real))

```

↳ A perda real do investimento foi de 2.28%

Exemplo ANEXO A.7 Um investidor depositou R\$ 30.000,00 na poupança no dia 10/10/2020. Porém, devido a um imprevisto financeiro, precisou retirar no dia 09/01/2021. Qual será o valor final resgatado? Considere que a rentabilidade da TR no período analisado foi de 0% e a SELIC foi de 7% ao ano.

Resolução:

- i. **Período de Investimento:** Nesse período, só há dois meses completos, de 10/10 a 10/11 e 10/11 a 10/12. Como a data de resgate foi 09/01, um dia antes do aniversário da Poupança, e não há taxas proporcionais para períodos menores que um mês, então o período de investimento foi 2 meses.
- ii. **Rentabilidade:** Como a taxa SELIC anual foi inferior a 8,5%, a rentabilidade da Poupança será de 70% da taxa da SELIC. Como a capitalização é feita mensalmente, primeiramente, devemos encontrar a taxa mensal equivalente. Daí, a taxa SELIC

$$1 + 0.07 = (1 + i)^{12} \Rightarrow 1.07^{\frac{1}{12}} = 1 + i \Rightarrow i \approx 0.0057 = 0.57\% \text{ a.m.}$$

E, portanto, o rendimento da Poupança é de $0,7 \cdot 0,57\% = 0,399\%$ a.m.

- iii. **Valor Final:** Conhecendo o capital inicial aplicado, a taxa de juros e o período

de aplicação, temos que o valor final resgatado, M , será de

$$M = 30.000 \times (1 + 0.00399)^2 = 30.239,88 \text{ reais}$$

Solução em Python:

```
valor_inicial = 30000.00
taxa_selic_anual = 0.07
# Calcular a taxa mensal equivalente
taxa_mensal = (1+ taxa_selic_anual) ** (1/12) - 1
# Calcular o rendimento da Poupança
rendimento_poupanca = 0.7* taxa_mensal
# Calcular o valor final resgatado
valor_final = valor_inicial * (1+ rendimento_poupanca) ** 2
print("Valor final resgatado: R${:.2f}" . format(valor_final))
```

Exemplo ANEXO A.8 *Oswaldo é um investidor com perfil mais conservador e decidiu investir R\$ 2.500,00 em títulos de renda fixa por um período de 2 anos. Do valor disponível, R\$ 1.000,00 foram investidos à uma taxa pré-fixada de 8% a.a., e os outros R\$ 1.500,00 foram investidos a taxa pós-fixada de 2% a.a. + IPCA. Sabendo que a taxa IPCA nesses dois anos foi, respectivamente, igual a 4,5% a.a. e 7% a.a., qual foi o montante resgatado por Oswaldo após esses dois anos?*

Solução em Python:

```
# Dados do investimento pré-fixado
valor_pre_fixado = 1000.00
taxa_pre_fixada = 0.08

# Dados do investimento pós-fixado
valor_pos_fixado = 1500.00
taxa_pos_fixada = 0.02
taxa_IPCA = [0.045,0.07]

# Cálculo do montante para o investimento pré-fixado
montante_pre_fixado = valor_pre_fixado * (1+ taxa_pre_fixada) ** 2

# Cálculo do montante para o investimento pós-fixado
montante_pos_fixado = valor_pos_fixado * (1+ taxa_pos_fixada) * (1+
taxa_IPCA[0]) * (1+ taxa_pos_fixada) * (1+ taxa_IPCA[1])
```

```
# Cálculo do montante total resgatado após dois anos
montante_total = montante_pre_fixado + montante_pos_fixado

print("Montante resgatado por Osvaldo após dois anos: R$ {:.2f}".format(montante_total))
>>> Montante resgatado por Osvaldo após dois anos: R$ 2.911,38
```

Exemplo ANEXO A.9 *Um título privado do CDB de valor R\$ 3.000,00 está disponível no mercado com vencimento em 750 dias com taxa pós-fixada de 120% do CDI. Sabe-se que nesse período a taxa CDI foi de 6,35% a.a.*

- a) *Qual será o valor líquido resgatado no vencimento?*
 b) *Considerando que a inflação, medida pelo índice IPCA, foi de 6% a.a., qual a taxa real de rendimento?*

Solução em Python:

```
valor_nominal = 3000.00
taxa_cdi = 0.0635
taxa_pos_fixada = 1.2 * taxa_cdi
periodo = 750
tax_inflacao = 0.06
# a) Calcular o valor líquido resgatado no vencimento
valor_liquido = valor_nominal * (1 + taxa_pos_fixada) ** (periodo/365)

print("Valor líquido resgatado no vencimento: R${:.2f}" . format(valor_liquido))

# b) Calcular a taxa real de rendimento taxa_real_rendimento = (1+ taxa_pos_fixada)
/ (1+ taxa_inflacao) -1

print("Taxa real de rendimento: {:.2%}" . format(taxa_real_rendimento))
```

Exemplo ANEXO A.10 *Um imóvel de R\$ 200.000,00 foi comprado da seguinte forma: Uma entrada de R\$ 56.000,00 e o restante financiado, por um banco, em 360 meses, a juros de 10% a.a., utilizando o SAC. Faça uma planilha de amortização com os valores das 6 primeiras prestações, sem considerar taxa de abertura de crédito, seguros, etc*

Solução em Python:

```
1 >>> import pandas as pd
```

```

# Dados iniciais
valor_imovel = 200000
entrada = 56000
prazo_meses = 360
taxa_anual = 0.1
# Cálculos
saldo_devedor = valor_imovel - entrada
taxa_mensal = (1 + taxa_anual) ** (1/12) - 1
prestacao = 400 + saldo_devedor * taxa_mensal
juros = saldo_devedor * taxa_mensal
amortizacao = 400
# Criação do dataframe
data = {"Saldo Devedor": [saldo_devedor],
        "Prestação": [prestacao],
        "Juros": [juros],
        "Amortização": [amortizacao]}
df = pd.DataFrame(data)
# Cálculo das prestações seguintes e atualização do dataframe
for i in range(2,7): saldo_devedor = saldo_devedor - amortizacao
    juros = saldo_devedor * taxa_mensal
    amortizacao = 400
    prestacao = amortizacao + juros
    data = {"Saldo Devedor": [saldo_devedor],
            'Prestação': [prestacao],
            'Juros': [juros],
            'Amortização': [amortizacao]}
    df = df.append(pd.DataFrame(data), ignore_index=True)
print(df)

```

```

↳ Saldo Devedor    Prestação    Juros    Amortização
0      144000    1548.276222    1148.276222    400
1      143600    1545.086566    1145.086566    400
2      143200    1541.896909    1141.896909    400
3      142800    1538.707253    1138.707253    400
4      142400    1535.517597    1135.517597    400
5      142000    1532.327941    1132.327941    400
<ipython-input-16-77091a0a54b0>:34: FutureWarning: The f

```

Exemplo ANEXO A.11 *Um financiamento de R\$ 60.000,00 será pago em 12 prestações mensais aplicando-se juros efetivos de 2,5% ao mês. Se o início dos pagamentos deve ocorrer ao término de um período de 3 meses, calcular o valor das prestações.*

Solução em Python :

Define o valor presente (PV), taxa de juros mensal (i) e número de períodos (n)

PV = 60000

i = 0.025

n = 12

Calcula o valor futuro (FV) após o período de carência de 2 meses

FV = PV * (1 + i) ** 2

Calcula o valor das prestações (P)

P = FV * (i * (1 + i) ** n) / ((1 + i) ** n - 1)

Imprime o valor das prestações

print("O valor das prestações é de R\${:.2f}".format(P))

Exemplo ANEXO A.12 *Ana completou hoje 24 anos e analisou a possibilidade de contribuir para um plano de aposentadoria privada. Pensa em aposentar-se ao completar 55 anos, quando gostaria de contar, a partir do seu aniversário, com uma renda mensal de R\$ 3.700,00. Sabendo que a taxa em vigor no mercado para planos de aposentadoria é de 0,64% ao mês, quanto ela deverá começar a depositar hoje de forma a ter o que deseja?*

Solução em Python:

1 >>> import numpy_financial as npf

2 # Dados do problema

3 idade_atual = 24

4 idade_aposentadoria = 55

3 renda_mensal_desejada = 3700

4 taxa_juros_mensal = 0.0064

Cálculo do valor a ser depositado mensalmente

5 num_meses = (idade_aposentadoria - idade_atual) * 12

6 fv = 3700/0.0064

7 pmt = npf.pmt(rate = taxa_juros_mensal, nper = num_meses, pv = 0, fv =

–fv, when = 'end')

```
7 print (f'Ana deve depositar R$ {pmt:.2f}' mensalmente a partir de hoje para
ter uma renda {renda_mensal_desejada:.2f} aos 55 anos.')
```

```
8 ... Ana deve depositar R$ 380.20 mensalmente a partir de hoje para ter uma
renda de R$ 3700.00 aos 55 anos.
```

Exemplo ANEXO A.13 *Um funcionário de uma empresa recebe um salário bruto de R\$ 5.300,00 e não tem dependentes. Escreva um algoritmo em Python para calcular o valor de IR a pagar, de acordo com as faixas da Base de Cálculo, considerando a alíquota de INSS de 11%. As alíquotas e faixas de tributação para essa tabela são as seguintes:*

- Até R\$ 1.903,98: isento
- De R\$ 1.903,99 a R\$ 2.826,65: alíquota de 7,5% e parcela a deduzir de R\$ 142,80
- De R\$ 2.826,66 a R\$ 3.751,05: alíquota de 15% e parcela a deduzir de R\$ 354,80
- De R\$ 3.751,06 a R\$ 4.664,68: alíquota de 22,5% e parcela a deduzir de R\$ 636,13
- Acima de R\$ 4.664,68: alíquota de 27,5% e parcela a deduzir de R\$ 869,36.

Solução em Python

Definir o valor do salário bruto e a alíquota de INSS

```
salario_bruto = 5300
```

```
aliquota_inss = 0.11
```

Calcular o valor do desconto de INSS

```
desconto_inss = salario_bruto * aliquota_inss
```

Definir as faixas de cálculo do imposto de renda para o ano base 2022

```
faixas =
```

```
0: (0, 1903.98, 0, 0),
```

```
1: (1903.99, 2826.65, 0.075, 142.80),
```

```
2: (2826.66, 3751.05, 0.15, 354.80),
```

```
3: (3751.06, 4664.68, 0.225, 636.13)
```

```
4: (4664.69, float('inf'), 0.275, 869.36)
```

```
salario_bruto = 5300.00 # salário bruto do funcionário
```

```
aliquota_inss = 0.11 # alíquota de INSS de 11%
```

```
parcela_deduzir = 869.36 # parcela a deduzir do IR
```

```

base_calculo = salario_bruto - (salario_bruto * aliquota_inss) # base de cálculo
do IR
if base_calculo <= 1903.98: aliquota_ir = 0.00
    deducacao_ir = 0.00
elif base_calculo <= 2826.65:
    aliquota_ir = 0.075
    deducacao_ir = 142.80
elif base_calculo <= 3751.05:
    aliquota_ir = 0.15
    deducacao_ir = 354.80
elif base_calculo <= 4664.68:
    aliquota_ir = 0.225
    deducacao_ir = 636.13
else: aliquota_ir = 0.275
    deducacao_ir = 869.36

    ir_pagar = base_calculo * aliquota_ir - deducacao_ir # valor de IR a
pagar
print(f"Valor de IR a pagar: R$ ir_pagar:.2f")

```

Este código considera as faixas de Base de Cálculo e suas respectivas alíquotas e deduções para o ano-base 2023. O valor de *IR* a pagar é calculado utilizando a fórmula $IR = Bc * iR - Pd$.

Link Códigos-fonte

<https://tinyurl.com/Atividades123>

ANEXO A.2 Cálculos financeiros com Numpy financial

NumPy Financial

- Instalando e importando módulos:

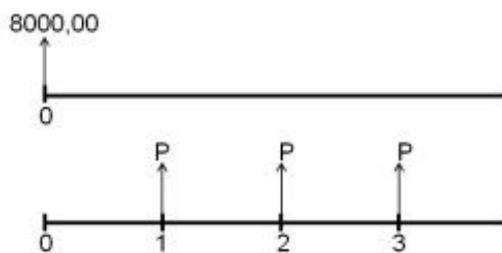

```

1 >>>!pip install numpy_financial
1 >>>>import numpy_financial as npf

```

Exemplo ANEXO A.14 *Um comerciante contraiu um empréstimo de R\$ 8000,00 a juros semestrais de 10%. O pagamento foi realizado em duas parcelas, uma de R\$ 5808,00 após um ano da contratação do empréstimo e a outra seis meses após a primeira.*

(a) Caso o comerciante optasse por quitar a dívida em 3 parcelas semestrais fixas, a primeira a partir do 1º semestre após a contratação do empréstimo, qual seria o valor das parcelas?



Solução em Python:

```

1 >>> ! pip install numpy_financial
2 >>> import numpy_financial as npf
3 npf.pmt(rate, nper, pv, fv=0, when="end")
4 pv = -8000
5 n = 3
6 i = 0.10
7 pmt = npf.pmt(i, n, p)
8 print(f"O valor das prestações serão = R$ {pv:.2f}")
9 ... O valor das prestações serão = R$ 3.216,92

```

Alguns exemplos de funções disponíveis no NumPy Financial incluem:

1. Valor Presente (PV) - **npf.pv**:

Exemplo ANEXO A.15 Se uma pessoa deseja obter R\$ 20.000,00 dentro de um ano, quanto deverá ela depositar na poupança que rende 0,5% de juros compostos ao mês?

Resolução:

$$PV = \frac{FV}{(1+r)^n}$$

```

npf.pv(rate, nper, pmt, fv=0, when='end')
fv = -20000
n = 12
i = 0.005
pv = npf.pv(i, n, 0, fv)
print(f'Deposito Inicial Necessário = R$ {pv:.2f}')

```

Deposito Inicial Necessário = R\$ 18.838,11

2. Valor Futuro (FV) - **npf.fv**:

Exemplo ANEXO A.16 Qual o valor de resgate ao aplicar R\$ 10.000,00 em um investimento pelo prazo de 8 meses à taxa de juros composta de 0,5% a.m.?

```
npf.fv(rate, nper, pmt, pv, when='end') npf.pv(rate, nper, pmt, pv, when='end')
```

```
pv = -10000
```

```
n = 8
```

```
i = 0.005
```

```
fv = npf.fv(i, n, 0, pv)
```

```
print(f'Valor de Resgate = R$ {fv:.2f}')
```

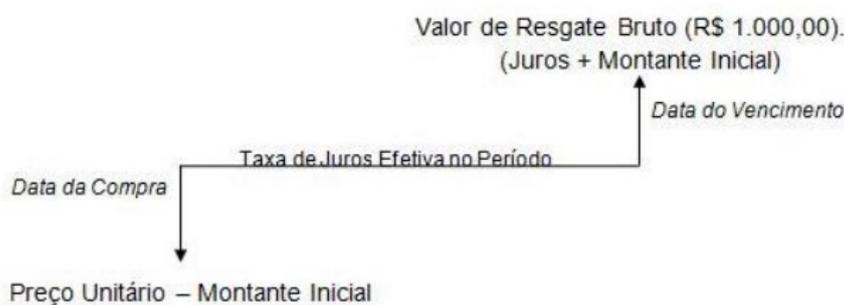
Valor de Resgate = R\$ 10.407,07

3. Valor Presente Líquido (NPV) - **npf.npv**:

Exemplo ANEXO A.17 Uma empresa esteja avaliando um investimento no valor de R\$ 750.000,00 do qual esperam-se benefícios anuais de caixa de R\$ 250.000,00 no primeiro ano, R\$ 320.000,00 no segundo ano, R\$ 380.000,00 no terceiro ano e R\$ 280.000,00 no quarto ano.

A empresa definiu em 20% ao ano a taxa de desconto a ser aplicada aos fluxos de caixa do investimento.

Figura 14 – Diagrama de Fluxo de Caixa



Fonte: Elaborada pelo autor

$$NPV = \sum_{j=1}^n \frac{FC_j}{(1+i)^j} - FC_0$$

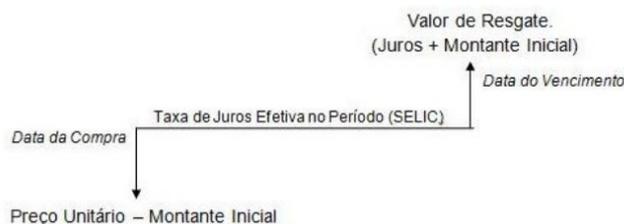
‘npf.npv(rate, values)’

```
1 >>>vpl = npf.npv(0.2, [-750000, 250000, 320000, 380000, 280000])
1 >>>print(f'VPL = R${vpl:.2f}')
VPL = R$ 35.493,83
```

4. Taxa Interna de Retorno (IRR) - **npf.irr** :

Exemplo ANEXO A.18 *Admita que um investimento de R\$ 70.000,00 gere caixa de R\$ 20.000,00, R\$ 40.000,00, R\$ 45.000,00 e R\$ 30.000,00, respectivamente, ao final dos próximos quatro anos.*

Figura 15 – Diagrama de Fluxo de Caixa



Fonte: Elaborada pelo autor

npf.irr()

```
ttir = npf.irr([-70000, 20000, 40000, 45000, 30000])
print(f'Taxa Interna de Retorno = {tir:.2%}')
Taxa Interna de Retorno = 30.03%
```

A partir do domínio dessas diretrizes e funcionalidades, torna-se possível explorar todo o potencial do módulo NumPy Financial para realizar cálculos financeiros de forma eficiente e precisa.

Códigos-fonte NumPy Financial

<<https://tinyurl.com/Numpyfinance>>

ANEXO A.3 Fontes de Dados Financeiros: Explorando yfinance e Pandas para Análise e Dados

I. Yahoo Finance

A seguir, apresentamos os principais aspectos envolvidos na obtenção de dados utilizando o módulo yfinance do Yahoo Finance, como fonte de dados financeiros.

O módulo yfinance se destaca como uma ferramenta fundamental na coleta e acesso a dados financeiros atualizados em tempo real. Por meio de uma simples integração com a linguagem de programação Python, torna-se possível extrair informações detalhadas sobre ações, índices, moedas e outros instrumentos financeiros.

Primeiramente, é necessário instalar e importar o módulo yfinance para o ambiente de desenvolvimento, garantindo que todas as funcionalidades estejam disponíveis. Em seguida, utilizamos o método "download" para especificar os ativos desejados, informando os símbolos correspondentes. É possível também definir a janela de tempo desejada, delimitando o período de dados a serem obtidos.

- Instalando o “yfinance”

```
1 >>>! pip install yfinance
```

```
1 >>>import yfinance as yf
```

- Códigos:

As empresas brasileira precisam do sufixo '.SA'. Ex.: VALE3.SA, BBDC4.SA

Índice Bovespa: ^BVSP

- Como por exemplo, Baixando Cotações do Índice Bovespa

Para isso utilizamos o método dowloud, conforme abaixo:

```
1 >>>yf . download()
```

```
ibov = yf . download('^BVSP')
```

```
1 >>>ibov
```

Date	Open	High	Low	Close	Adj Close	Volume
1993-04-27	24.799999	25.400000	24.500000	24.500000	24.500000	0
1993-04-28	24.500000	24.600000	23.700001	24.299999	24.299999	0
1993-04-29	24.299999	24.799999	23.700001	23.700001	23.700001	0
1993-04-30	23.700001	24.200001	23.700001	24.100000	24.100000	0
1993-05-03	24.100000	24.400000	23.799999	24.100000	24.100000	0
...
2023-04-20	103913.000000	104615.000000	103087.000000	104367.000000	104367.000000	12378000
2023-04-24	104367.000000	104822.000000	103247.000000	103947.000000	103947.000000	10935400
2023-04-25	103947.000000	103947.000000	102633.000000	103220.000000	103220.000000	10873200
2023-04-26	103220.000000	103668.000000	102233.000000	102312.000000	102312.000000	10115500
2023-04-27	102310.000000	103177.000000	101975.000000	102923.000000	102923.000000	0

7425 rows x 6 columns

Base de Dados:

<<https://tinyurl.com/yfinanceyahoo>>

II. Consulta - Banco Central do Brasil

Sistema Gerenciador de Séries Temporais do Bacen (SGS)

<<https://tinyurl.com/SGSBC>>

Endereço padrão da API:

http://api.bcb.gov.br/dados/serie/bcdata.sgs.codigo_serie}/dados?formato=json&dataInicial={dataInicial}&dataFinal={dataFinal}

Importando Bibliotecas

- `import pandas as pd`
- `from datetime import date`

1 >>> `import pandas as pd` # importa a biblioteca NumPy e renomeia como **pd**.

Obtendo Dados através da API do Banco Central do Brasil

```

1 def consulta_bc(codigo_serie, dataInicial='01/01/1900', dataFinal=date.today().strftime('%d/%m/%Y')):
2     url = f'http://api.bcb.gov.br/dados/serie/bcdata.sgs.{codigo_serie}/dados?formato=json&dataInicial={dataInicial}&dataFinal={dataFinal}'
3     df = pd.read_json(url)
4     df['data'] = pd.to_datetime(df['data'], dayfirst=True)
5     df.set_index('data', inplace=True)
6     return df

```

Exemplo ANEXO A.19 *Exemplos de Consultas à API do Banco Central do Brasil*

```
1 >>> ipca = consulta_bc(433)
```

```
1 >>> ipca
```

1 ipca	
	valor 
data	
1980-01-01	6.62
1980-02-01	4.62
1980-03-01	6.04
1980-04-01	5.29
1980-05-01	5.70
...	...
2022-12-01	0.62
2023-01-01	0.53
2023-02-01	0.84
2023-03-01	0.71
2023-04-01	0.61

520 rows x 1 columns

```
1 >>> ipca_mensal = consulta_bc(433, '01/01/2022', '31/12/2022')
```

```
1 >>> ipca_mensal
```

[]	
	valor 
data	
2022-01-01	0.54
2022-02-01	1.01
2022-03-01	1.62
2022-04-01	1.06
2022-05-01	0.47
2022-06-01	0.67
2022-07-01	-0.68
2022-08-01	-0.36
2022-09-01	-0.29
2022-10-01	0.59
2022-11-01	0.41
2022-12-01	0.62

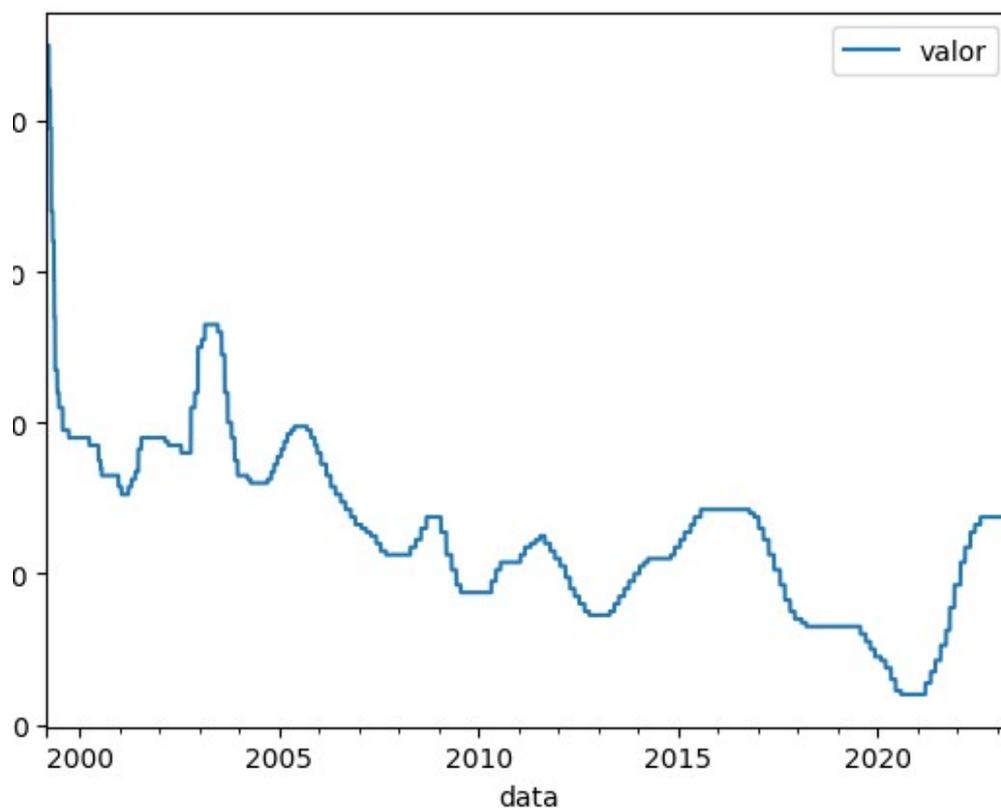
```
1 >>> igpm = consulta_bc(189)
```

```
2 >>> igpm
```

data	
1989-06-01	19.68
1989-07-01	35.90
1989-08-01	36.92
1989-09-01	39.92
1989-10-01	40.64
...	...
2022-12-01	0.45
2023-01-01	0.21
2023-02-01	-0.06
2023-03-01	0.05
2023-04-01	-0.95

407 rows x 1 columns

```
1 >>>selic_meta = consulta_bc(432)
2 selic_meta.plot(figsize=(12,8));
```



Link:

<https://tinyurl.com/BancoCentralBrasil>

Consulta Tesouro Nacional

Base de Dados: <<https://www.tesourotransparente.gov.br/ckan/dataset>>

Importações e Configurações Iniciais:

1 >>> import pandas as pd # importa a biblioteca NumPy e renomeia como **pd**.

Funções para Busca e Tratamento de Dados do Tesouro Direto.

```
[ ] 1 def busca_titulos_tesouro_direto():
2     url = "https://www.tesourotransparente.gov.br/ckan/dataset/df56aa42-484a-4a59-8184-7676580c81e3/resource/796d2059-14e9-44e3-80c9-2d9e30b405c1/download/Preco"
3     df = pd.read_csv(url, sep=';', decimal=',')
4     df['Data Vencimento'] = pd.to_datetime(df['Data Vencimento'], dayfirst=True)
5     df['Data Base'] = pd.to_datetime(df['Data Base'], dayfirst=True)
6     multi_index = pd.MultiIndex.from_frame(df.iloc[:, :3])
7     df = df.set_index(multi_index).iloc[:, 3:]
8     return df

[ ] 1

[ ] 1 def busca_vendas_tesouro():
2     url = "https://www.tesourotransparente.gov.br/ckan/dataset/f0468ecc-ae97-4287-89c2-6d8139fb4343/resource/e5f90e3a-8f8d-4895-9c56-4bb2f7877920/download/Venda"
3     df = pd.read_csv(url, sep=';', decimal=',')
4     df['Vencimento do Titulo'] = pd.to_datetime(df['Vencimento do Titulo'], dayfirst=True)
5     df['Data Venda'] = pd.to_datetime(df['Data Venda'], dayfirst=True)
6     multi_index = pd.MultiIndex.from_frame(df.iloc[:, :3])
7     df = df.set_index(multi_index).iloc[:, 3:]
8     return df

[ ] 1 def busca_recompras_tesouro():
2     url = "https://www.tesourotransparente.gov.br/ckan/dataset/f30db6e4-6123-416c-b094-be8dfc823601/resource/30c2b3f5-6edd-499a-8514-062bfa0f61a/download/Recompra"
3     df = pd.read_csv(url, sep=';', decimal=',')
4     df['Vencimento do Titulo'] = pd.to_datetime(df['Vencimento do Titulo'], dayfirst=True)
5     df['Data Resgate'] = pd.to_datetime(df['Data Resgate'], dayfirst=True)
6     multi_index = pd.MultiIndex.from_frame(df.iloc[:, :3])
7     df = df.set_index(multi_index).iloc[:, 3:]
8     return df
```

Preços e Taxas Históricas dos Títulos Negociados

Essa conjunto de dados contém as taxas e preços dos títulos ofertados pelo Tesouro Direto. Refletem o mercado secundário de títulos públicos federais.

Buscando dados

1 >>> titulos = busca_titulos_tesouro_direto

1 >>> titulos.sort_index(inplace=True)

2 titulos

			Taxa Compra Manha	Taxa Venda Manha	PU Compra Manha	PU Venda Manha	PU Base Manha
Tipo Titulo	Data Vencimento	Data Base					
Tesouro IGPM+ com Juros Semestrais	2005-07-01	2004-12-31	9.23	9.25	1776.67	1776.51	1827.87
		2005-01-03	9.23	9.25	1777.71	1777.55	1776.51
		2005-01-04	9.26	9.28	1778.52	1778.36	1777.32
		2005-01-05	9.28	9.30	1779.40	1779.25	1778.20
		2005-01-06	9.31	9.33	1780.21	1780.06	1779.01
...
Tesouro Selic	2021-03-01	2021-03-05	0.33	0.34	10585.34	10579.05	10578.07
		2021-03-08	0.33	0.34	10585.13	10578.85	10577.88
		2021-03-09	0.33	0.34	10585.06	10578.78	10577.80
		2021-03-10	0.33	0.34	10587.62	10581.35	10580.38
		2021-03-11	0.33	0.34	10587.11	10580.83	10579.86

1 >>> tipos_titulos = titulos.index.droplevel(level=1).droplevel(level=1).drop_duplicates().to_list()

2 tipos_titulos

```
['Tesouro IGPM+ com Juros Semestrais',
 'Tesouro IPCA+',
 'Tesouro IPCA+ com Juros Semestrais',
 'Tesouro Prefixado',
 'Tesouro Prefixado com Juros Semestrais',
 'Tesouro Selic']
```

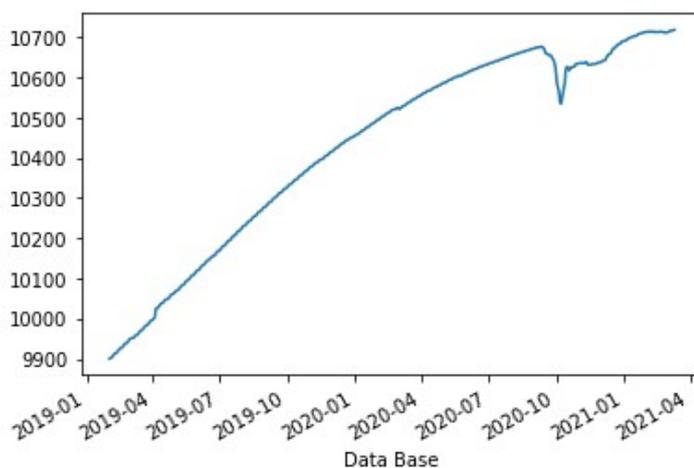
Tesouro Selic

```
1 >>>selic2025 = titulos.loc(['Tesouro Selic', '2025-03-01']) 2 selic2025
```

```
1 selic2025 = titulos.loc(['Tesouro Selic', '2025-03-01'])
2 selic2025
```

Data Base	Taxa Compra Manha	Taxa Venda Manha	PU Compra Manha	PU Venda Manha	PU Base Manha
2019-02-01	0.02	0.06	9926.99	9902.94	9900.45
2019-02-04	0.02	0.06	9929.43	9905.40	9902.91
2019-02-05	0.02	0.06	9931.88	9907.87	9905.36
2019-02-06	0.02	0.06	9934.34	9910.33	9907.83
2019-02-07	0.02	0.06	9936.79	9912.80	9910.29
...
2021-03-05	0.17	0.18	10720.90	10716.64	10715.72
2021-03-08	0.17	0.18	10721.48	10717.22	10716.30
2021-03-09	0.17	0.18	10722.22	10717.97	10717.05
2021-03-10	0.18	0.19	10722.64	10718.38	10717.46
2021-03-11	0.18	0.19	10723.08	10718.84	10717.92

```
[ ] 1 selic2025['PU Base Manha'].plot();
```



Vendas Tesouro Direto

Este conjunto de dados traz o volume de vendas diário do programa Tesouro Direto.

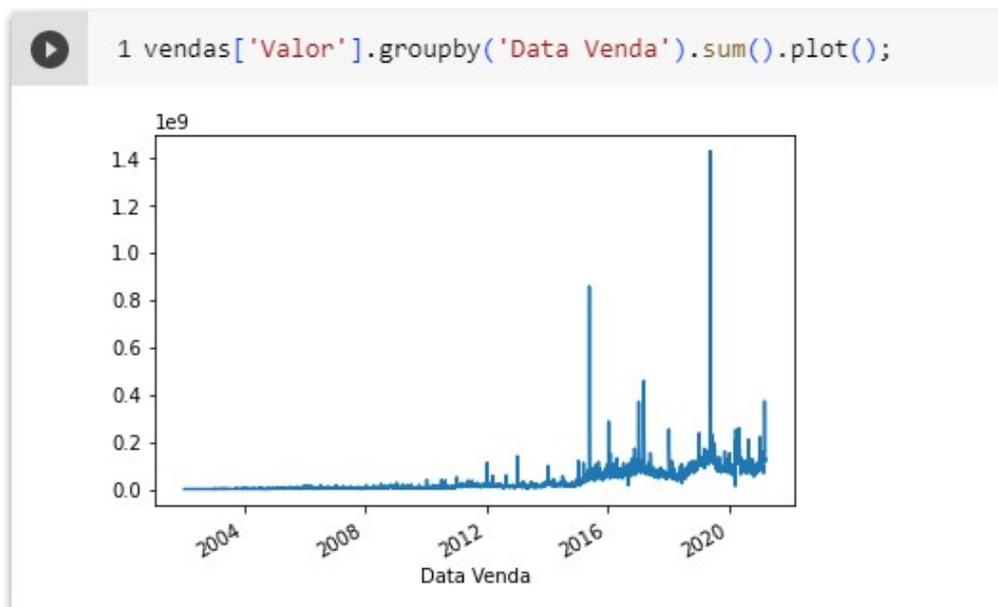
Buscando Dados

```
1 >>> selic2025 = titulos.loc(['Tesouro Selic', '2025-03-01']) vendas = busca_vendas_tesouro( )
1 >>> vendas.sort_index(inplace=True)
2 vendas
```

Tipo Titulo	Vencimento do Titulo	Data Venda	PU	Quantidade	Valor
Tesouro IGPM+ com Juros Semestrais	2005-07-01	2002-01-07	1038.150000	7.40	7682.31
		2002-01-08	1035.590000	6.00	6213.54
		2002-01-09	1035.400000	0.20	207.08
		2002-01-14	1038.166666	0.60	622.89
		2002-01-15	1038.640000	1.00	1038.64
...
Tesouro Selic	2027-03-01	2021-03-04	10582.345713	2786.96	29492574.20
		2021-03-05	10587.056401	2278.12	24118584.92
		2021-03-08	10585.334565	2382.97	25224534.70
		2021-03-09	10585.124474	2215.79	23454412.95
		2021-03-10	10585.055192	1910.57	20223488.89

65003 rows x 3 columns

Gráficos



Link:

<<https://tinyurl.com/TesouroNacio>>

CÁLCULO DE RETORNO

1. RETORNOS DIÁRIOS

- Inicialmente iremos instalar o yfinance, que nos permitirá acessar os dados do Yahoo Finance.
- Após a instalação bem-sucedida do yfinance, podemos começar a atividade importando as bibliotecas necessárias, incluindo o Pandas.
- Com as bibliotecas importadas, podemos prosseguir obtendo os dados históricos de preços do índice bovespa desejados do Yahoo Finance. Para isso, vamos usar a função yfinance.download() para fazer o download desses dados em um DataFrame. Nesse ponto, é importante especificar o símbolo da ação e o intervalo de datas desejado para obter os dados corretos.

```

✓ 5s 1 !pip install -q yfinance

✓ 2s [3] 1 import pandas as pd
      2 import yfinance as yf
      3 import seaborn as sns
      4 import matplotlib.pyplot as plt
      5 import numpy as np

✓ 0s [4] 1 prices = yf.download('^BVSP' , start='1994-12-29')
      [*****100%*****] 1 of 1 completed

✓ 1 prices
  
```

- Uma vez que os dados são obtidos e armazenados em um DataFrame, podemos utilizar as funcionalidades do Pandas para realizar o cálculo do retorno. O retorno é uma medida que indica o ganho ou perda percentual de um ativo financeiro ao longo de um determinado período de tempo. No contexto dessa atividade, estamos interessados em calcular o retorno com base nos dados reais do mercado

Sendo r_t e P_t o **retorno simples** e o **preço no tempo t** :

$$r_t \equiv \frac{P_t - P_{t-1}}{P_{t-1}} = \frac{P_t}{P_{t-1}} - 1$$

- Entendendo o deslocamento com shift()

```

1 # Entendendo o deslocamento com shift()
2 df = pd.DataFrame()
3 df['price'] = prices['Adj Close']
4 df['price_shift_1'] = prices['Adj Close'].shift(1)
5 df.head()

```

	price	price_shift_1
Date		
1994-12-29	4353.899902	NaN
1995-01-02	4301.000000	4353.899902
1995-01-03	4098.000000	4301.000000
1995-01-04	3967.899902	4098.000000
1995-01-05	4036.699951	3967.899902

- Calculando o retorno usando shift()

```

[10] 1 # Calculando o retorno usando shift()
2 retorno_diario = (prices['Adj Close'] / prices['Adj Close'].shift(1) - 1)

```

```

1 retorno_diario

```

Date	
1994-12-29	NaN
1995-01-02	-0.012150
1995-01-03	-0.047198
1995-01-04	-0.031747
1995-01-05	0.017339
	...
2023-06-09	0.013257
2023-06-12	0.002709
2023-06-13	-0.005054
2023-06-14	0.019924
2023-06-15	-0.001973

Name: Adj Close, Length: 7042, dtype: float64

- Calculando o retorno usando pct_change()

Podemos utilizar também, para fazer os mesmos cálculos, a função `pct_change()` fornecida pelo Pandas. Essa função calcula a variação percentual entre os preços consecutivos de um ativo. Após aplicar essa função ao DataFrame que contém os dados de preços, obteremos uma nova coluna com os retornos correspondentes a cada período.

```
[ ] 1 # Calculando o retorno usando pct_change()
     2 retorno_diario = prices['Adj Close'].pct_change()
```

```
1 retorno_diario
```

```
Date
1994-12-29      NaN
1995-01-02    -0.012150
1995-01-03    -0.047198
1995-01-04    -0.031747
1995-01-05     0.017339
...
2023-06-09     0.013257
2023-06-12     0.002709
2023-06-13    -0.005054
2023-06-14     0.019924
2023-06-15    -0.001973
Name: Adj Close, Length: 7042, dtype: float64
```

- Calculo pelo Retorno Logaritmo

O **retorno logarítmico** é dado por:

$$R_t \equiv \ln\left(\frac{P_t}{P_{t-1}}\right)$$

```
[ ] 1 retorno_diario_log = np.log(prices['Adj Close'] / prices['Adj Close'].shift(1))
```

```
1 retorno_diario_log
```

```
Date
1994-12-29      NaN
1995-01-02    -0.012224
1995-01-03    -0.048349
1995-01-04    -0.032262
1995-01-05     0.017191
...
2021-03-26     0.009023
2021-03-29     0.005543
2021-03-30     0.012322
2021-03-31    -0.001850
2021-04-01    -0.011911
Name: Adj Close, Length: 6491, dtype: float64
```

Link:

<<https://tinyurl.com/RetornoDiario1>>

<<https://tinyurl.com/CalculandoRetorno>>

2. RETORNO MENSAL

Nessa perspectiva iremos calcular o retorno mensal, utilizando a base de dados

anterior. No contexto do uso da biblioteca pandas do Python, vamos utilizar o método “**resample**” oferece uma forma de **reorganização** dos dados em **períodos de tempo diferentes**. Através desse método, é possível agrupar dados diários em períodos mensais, permitindo uma visão mais ampla e resumida das informações. O método *resample* é utilizado com o argumento ‘**M**’, indicando que desejamos reamostrar os dados em períodos mensais. Essa função divide os dados em grupos mensais, permitindo a aplicação de funções específicas para cada período. Para determinar qual informação é relevante em cada período mensal, é necessário especificar a função a ser aplicada. No caso apresentado, o interesse está no último dia de cada mês, o qual é comparado com o último dia do mês seguinte. Para obter esse resultado, a função **last()** é utilizada. Essa função seleciona o último dado disponível em cada grupo mensal, fornecendo uma representação do final de cada mês. Mas como queremos trabalhar com intervalos, vamos passar outro parâmetro chamado de “**kind**”, o tipo de agrupamento vai ser em períodos. Na sequência, para calcular o retorno vamos utilizar a função “**pct_change**” e finalmente para retornar um DataFrame, vamos utilizar o método “**to_frame**”, visando facilitar a inclusão de novos dados.

Retorno mensal

```
[13] 1 retorno_mensal = prices['Adj Close'].resample('M', kind='period').last().pct_change().
```

```
1 retorno_mensal.tail()
```

Date	Adj Close
2023-02	-0.075750
2023-03	-0.029066
2023-04	0.025029
2023-05	0.037374
2023-06	0.084391

A seguir, vamos aprimorar nossa análise de dados criando uma tabela que per-

mita a ordenação separada por anos e meses, a fim de visualizar essas informações em duas dimensões. Para isso, iremos adicionar duas colunas chamadas "mes" e "ano" ao DataFrame.

```

1 retorno_mensal['mes'] = retorno_mensal.index.month
2 retorno_mensal['ano'] = retorno_mensal.index.year

1 retorno_mensal.tail()

```

Date	Adj Close	mes	ano
2023-02	-0.075750	2	2023
2023-03	-0.029066	3	2023
2023-04	0.025029	4	2023
2023-05	0.037374	5	2023
2023-06	0.100485	6	2023

Para realizar essa análise, vamos utilizar o método *pivot* do pacote Pandas. Esse método nos permite reorganizar os valores da tabela de acordo com os meses e anos.

Nessa tabela, os valores representarão os retornos mensais para cada mês. A coluna correspondente a esses valores será a coluna de retorno. Os anos serão utilizados como índice, ou seja, a cada ano teremos uma linha na tabela. Além disso, os meses serão as colunas da tabela, onde cada coluna representa um mês específico.

Dessa forma, teremos uma tabela bidimensional, onde cada linha representará um ano e cada coluna representará um mês.

Essa tabela conterá os retornos mensais para cada período analisado, permitindo uma visualização clara dos dados em duas dimensões."

```
[17] 1 tab_retorno_mensal = retorno_mensal.pivot(values='Adj Close', columns='mes', index='ano')
```

```
1 tab_retorno_mensal.tail()
```

mes	1	2	3	4	5	6	7	8	9	10	11	12
2019	0.108173	-0.018584	-0.001768	0.009831	0.007026	0.040575	0.008369	-0.006650	0.035695	0.023629	0.009448	0.071429
2020	-0.018997	-0.084291	-0.299044	0.102520	0.085671	0.087560	0.082646	-0.034427	-0.047963	-0.006881	0.158975	0.095676
2021	-0.027652	-0.051480	0.059972	0.019377	0.061584	0.004643	-0.039439	-0.024795	-0.065684	-0.067382	-0.015324	0.028524
2022	0.072180	0.006709	0.060605	-0.101026	0.032213	-0.115033	0.046914	0.061629	0.004693	0.054527	-0.030602	-0.021825
2023	0.031818	-0.075750	-0.029066	0.025029	0.037374	0.084391	NaN	NaN	NaN	NaN	NaN	NaN

Em seguida, para uma melhor visualização da análise de dados, iremos criar um mapa de calor (heatmap) utilizando a função “sns.heatmap()” de forma a facilitar a interpretação dos resultados.

```
1 #@title
2 plt.figure(figsize=(12,12))
3 sns.set(font_scale=0.8)
4
5 ax = sns.heatmap(tab_retorno_mensal,
6                  annot = True,
7                  fmt = ".1%",
8                  cmap = "RdYlGn",
9                  vmin = -0.2,
10                 vmax = 0.2,
11                 center = 0,
12                 cbar=False)
13
14 ax.set_title('Retorno Mensal do IBOV', pad=15, fontdict={'fontsize':12, 'fontweight':600})
15 ax.tick_params(axis = "y", labelright =True, labelrotation=90 )
16 ax.tick_params(axis = "x", labeltop=True)
17
```

- Configure as dimensões da figura do gráfico utilizando a função `plt.figure(figsize=(12,12))`. Isso define as dimensões da visualização que será gerada.
- O parâmetro `annot=True` indica que queremos exibir os valores nas células do mapa de calor.
- O parâmetro `fmt=".1%"` formata os valores como porcentagens com uma casa decimal.
- O parâmetro `cmap="RdYlGn"` define a paleta de cores a ser usada no mapa de calor (vermelho-amarelo-verde).
- O parâmetro `vmin=-0.2` e `vmax= 0.2` definem os valores mínimo e máximo da escala de cores do mapa de calor.
- O parâmetro `center= 0` especifica que o valor zero será o ponto central da escala de cores.
- O parâmetro `cbar = False` remove a barra de cores lateral (barra de referência).

Nessas próximas linhas, estaremos personalizando o gráfico.

- `ax.set_title()` define o título do gráfico como “Retorno Mensal do IBOV”. O parâmetro `pad = 15` especifica a distância entre o título e o gráfico em pixels. O parâmetro `fontdict` permite definir o tamanho (fontsize) e peso (fontweight) da fonte do título.
- `ax.tick_params()` ajusta os parâmetros dos rótulos dos eixos. `axis = “y”` configura as opções dos rótulos do eixo *y* (vertical), enquanto `axis= “x”` configura as opções dos rótulos do eixo *x* (horizontal).
- `labelright=True` move os rótulos do eixo *y* para a direita.
- `labelrotation = .90` rotaciona os rótulos do eixo *y* em 90 graus.
- `labeltop = True` move os rótulos do eixo *x* para a parte superior do gráfico.

Retorno Mensal do IBOV													
	1	2	3	4	5	6	7	8	9	10	11	12	
1995	-10.8%	-15.8%	-8.9%	28.0%	-2.4%	-3.2%	7.6%	11.2%	8.3%	-11.6%	6.1%	-1.8%	1995
1996	19.8%	-3.8%	-0.1%	4.2%	10.9%	5.5%	1.3%	2.2%	3.0%	1.3%	2.0%	5.6%	1996
1997	13.1%	10.8%	2.4%	10.4%	13.7%	10.8%	2.4%	-17.6%	11.2%	-23.8%	4.6%	8.5%	1997
1998	-4.7%	8.8%	13.0%	-2.3%	-15.7%	-1.7%	10.6%	-39.6%	1.9%	6.9%	22.5%	-21.4%	1998
1999	20.5%	9.0%	20.0%	6.1%	-2.3%	4.8%	-10.2%	1.2%	5.1%	5.3%	17.8%	24.0%	1999
2000	-4.1%	7.8%	0.9%	-12.8%	-3.7%	11.8%	-1.6%	5.4%	-8.2%	-6.7%	-10.6%	14.8%	2000
2001	15.8%	-10.1%	-9.1%	3.3%	-1.8%	-0.6%	-5.5%	-6.6%	-17.2%	6.9%	13.8%	5.0%	2001
2002	-6.3%	10.3%	-5.5%	-1.3%	-1.7%	-13.4%	-12.4%	6.3%	-16.9%	17.9%	3.4%	7.2%	2002
2003	-2.9%	-6.0%	9.7%	11.4%	6.9%	-3.3%	4.6%	11.8%	5.5%	12.3%	12.2%	10.2%	2003
2004	-1.7%	-0.4%	1.8%	-11.4%	-0.3%	8.2%	5.6%	2.1%	1.9%	-0.8%	9.0%	4.3%	2004
2005	-7.0%	15.6%	-5.4%	-6.6%	1.5%	-0.6%	4.0%	7.7%	12.6%	-4.4%	5.7%	4.8%	2005
2006	14.7%	0.6%	-1.7%	6.4%	-9.5%	0.3%	1.2%	-2.3%	0.6%	7.7%	6.8%	6.1%	2006
2007	0.4%	-1.7%	4.4%	6.9%	6.8%	4.1%	-0.4%	0.8%	10.7%	8.0%	-3.5%	1.4%	2007

Link:

<https://tinyurl.com/RetornoMensal>

ANEXO A.4 Visualização de dados financeiros com Matplotlib

MPL Finance

Documentação Oficial:

<https://github.com/matplotlib/mplfinance>

A seguir relacionamos os principais pontos para plotagem de gráficos, utilizando o módulo Matplotlib.

- Instalando e importando módulos:


```
1 >>>import matplotlib.pyplot as plt
1 >>>import pandas as plt
```


- Vamos carregar a função `consulta_bc` responsável por consultar os dados do Banco Central. Essa função irá receber o código da série desejada, a data inicial e a data final como parâmetros.

```
[ ] 1 def consulta_bc(codigo_serie, dataInicial='01/01/1900', dataFinal=date.today().strftime('%d/%m/%Y')):
2     url = f'http://api.bcb.gov.br/dados/serie/bcdata.sgs.{codigo_serie}/dados?formato=json&dataInicial={dataInicial}&dataFinal={dataFinal}'
3     df = pd.read_json(url)
4     df['data'] = pd.to_datetime(df['data'], dayfirst=True)
5     df.set_index('data', inplace=True)
6     return df

[ ] 1 data_inicial = '01/01/2000'
2     data_final   = '25/03/2023'
```

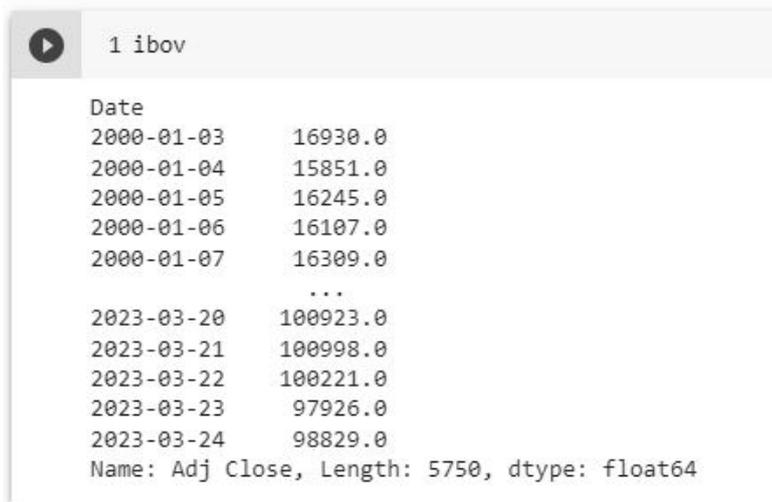
Essa função utiliza a API do Banco Central para obter os dados da série especificada. Ela formata a URL corretamente com o código da série, a data inicial e a data final. Em seguida, lê os dados JSON obtidos a partir da URL e realiza algumas transformações, convertendo a coluna de datas para o formato apropriado e definindo-a como índice do dataframe resultante. Por fim, retorna o dataframe com os dados.

- Na próxima etapa faremos o download dos dados históricos do IBOV (Índice Bovespa) utilizando a função `yf.download()` do Yahoo Finance.

```
1 ibov = yf.download('^BVSP', start=pd.to_datetime(data_inicial, dayfirst=True), end=pd.to_datetime(data_final, dayfirst=True))['Adj Close']
```

Neste exemplo, estamos utilizando o código `^BVSP` para representar o IBOV. A função `yf.download()` nos permite especificar a data de início e a data final dos dados que desejamos baixar. Utilizamos as funções `pd.to_datetime()` para converter as datas de início e fim para o formato correto, com o parâmetro `dayfirst=True` indicando que o formato da data é dia/mês/ano. A seguir, utilizamos `['Adj Close']` para selecionar apenas os preços de fechamento ajustados dos dados baixados.

Após a execução desta etapa, teremos o dataframe `ibov` contendo os preços de fechamento ajustados do IBOV no período especificado, do `data_inicial` até o `data_final`.



```

1 ibov
Date
2000-01-03    16930.0
2000-01-04    15851.0
2000-01-05    16245.0
2000-01-06    16107.0
2000-01-07    16309.0
...
2023-03-20    100923.0
2023-03-21    100998.0
2023-03-22    100221.0
2023-03-23     97926.0
2023-03-24     98829.0
Name: Adj Close, Length: 5750, dtype: float64

```

- Consulta dos dados do CDI: Utilize a função `consulta_bc()` para obter os dados do CDI do Banco Central. Passe o código da série do CDI (no exemplo, 12) e as datas de início e fim definidas anteriormente (`data_inicial` e `data_final`). O resultado será um dataframe chamado `cdi` contendo os dados do CDI no período especificado.
- Cálculo do acumulado do CDI: Adicione uma nova coluna chamada "acumulado" ao dataframe `cdi`, que representará o acumulado do CDI ao longo do tempo. Utilize o cálculo $(1 + (\text{cdi}[\text{'diaria'}] / 100)).\text{cumprod}().\text{shift}(1)$ para obter o acumulado diário do CDI. Essa fórmula calcula o acumulado como o produto cumulativo dos retornos diários, onde cada retorno é calculado como 1 mais a taxa diária do CDI dividida por 100.
- Ajuste do primeiro valor do acumulado: Defina o primeiro valor do acumulado do CDI como 1. Isso é necessário porque o cálculo do acumulado utiliza o deslocamento (`.shift(1)`), o que resulta em um NaN (valor ausente) para o primeiro elemento do acumulado. Portanto, é necessário substituir esse valor ausente por 1.
- Após executar essas etapas, teremos o dataframe `cdi` com a coluna "acumulado" representando o acumulado diário do CDI, onde o primeiro valor é igual a 1. Assim, pode prosseguir com a análise e comparação entre o IBOV e o CDI utilizando esses dados.

```
[11] 1 cdi['acumulado'] = (1 + (cdi['diaria'] / 100)).cumprod().shift(1)
     2 cdi['acumulado'].iloc[0] = 1
```

```
1 cdi
```

	diaria	acumulado
data		
1999-12-31	0.068485	1.000000
2000-01-03	0.068318	1.000685
2000-01-04	0.068218	1.001368
2000-01-05	0.068184	1.002052
2000-01-06	0.068218	1.002735
...
2023-03-20	0.050788	13.987329
2023-03-21	0.050788	13.994432

- Concatenando os dados do IBOV e CDI: Utilize a função `pd.concat()` para concatenar os dados do IBOV e do CDI em um único dataframe. Especificamente, iremos concatenar as colunas `ibov / ibov.iloc[0]` (retornos normalizados do IBOV em relação ao primeiro valor) e `cdi['acumulado']` (acumulado do CDI).
- O resultado será um novo dataframe chamado `df_ibov_cdi` que contém as colunas "IBOV" e "CDI", representando os retornos normalizados do IBOV e o acumulado do CDI. Utilize a propriedade `columns` do dataframe para renomear as colunas "IBOV" e "CDI" de acordo com as suas respectivas nomenclaturas.
- Após executarmos essas etapas, teremos o dataframe `df_ibov_cdi` que contém os retornos normalizados do IBOV e o acumulado do CDI, permitindo a comparação entre essas duas séries. Esse dataframe está pronto para ser utilizado em análises e visualizações.

```
[13] 1 df_ibov_cdi = pd.concat([ibov / ibov.iloc[0], cdi['acumulado']], axis=1).dropna()
```

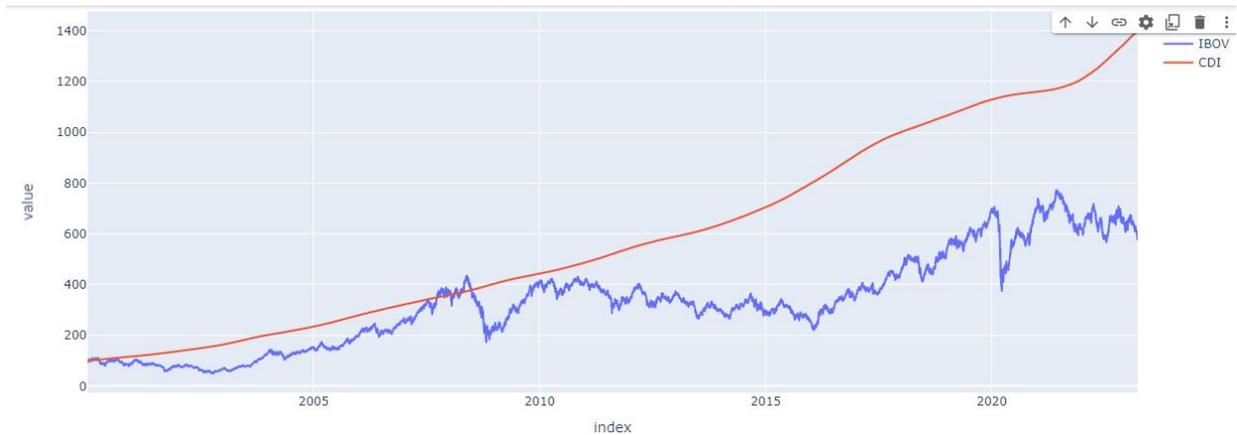
```
[14] 1 df_ibov_cdi.columns = ['IBOV', 'CDI']
```

```
1 df_ibov_cdi.tail()
```

	IBOV	CDI
2023-03-20	5.961193	13.987329
2023-03-21	5.965623	13.994432
2023-03-22	5.919728	14.001540
2023-03-23	5.784170	14.008651
2023-03-24	5.837507	14.015766

- Para finalizar, vamos plotar um gráfico para visualização dos dados em termos percentuais que irá exibir as curvas de retorno normalizado do IBOV e acumulado do CDI ao longo do tempo.

Figura 16 – IBOV vs CDI



Fonte: Elaborada pelo autor

No link abaixo temos um arquivo com a aplicação e código fonte dos itens :

<https://tinyurl.com/IBOVvsCDI>

REFERÊNCIAS

- BARCELOS, T. S.; SILVEIRA, I. F. Pensamento computacional e educação matemática: Relações para o ensino de computação na educação básica. In: SN. *XX Workshop sobre Educação em Computação, Curitiba. Anais do XXXII CSBC*. [S.l.], 2012. v. 2, p. 23.
- BORBA, M. C.; SILVA, R. S. R. d.; GADANIDIS, G. Fases das tecnologias digitais em educação matemática. ; 2 reimp. *Belo Horizonte: Autêntica Editora*, 2018.
- BOYER, C. B.; MERZBACH, U. C. *História da matemática*. [S.l.]: Editora Blucher, 2019.
- BRASIL, M. Base nacional comum curricular. *Brasília-DF: MEC, Secretaria de Educação Básica*, 2017.
- BRUNI, A. L. *Matemática financeira para concursos*. [S.l.]: Editora Atlas SA, 2000.
- CERBASI, G. *Investimentos inteligentes: Guia de estudos*. [S.l.]: Thomas Nelson Brasil, 2009.
- CIEB. *Currículo de Referência em Tecnologia e Educação*. 2022. Disponível em: <:http://curriculo.cieb.net.br/>. Acesso em: 01 de fevereiro 2022.
- CORDEIRO, N. J. N.; COSTA, M. G. V.; SILVA, M. N. da. Educação financeira no brasil: uma perspectiva panorâmica. *Ensino da Matemática em Debate*, v. 5, n. 1, p. 69–84, 2018.
- FARIA, R. G. de. *Matemática comercial e financeira*. [S.l.: s.n.], 1983.
- JARDIM, A. et al. Investimento em mercado de capitais como ferramenta de apoio ao investidor. um estudo documental. *PESQUISA & EDUCAÇÃO A DISTÂNCIA*, n. 7, 2016.
- KIYOSAKI, R. T.; LECHTER, S. L. Pai rico pai pobre. *Rio de Janeiro: Campus*, 2000.

- LIEDTKE, A. M. Introdução à matemática financeira. *Roteiro Pedagógico*. [20-]. Disponível em: <http://www.diaadiaeducacao.pr.gov.br/portals/roteiro pedagogico/relato/2197_Introducao_a_Matematica_Financeira.PDF>. Acesso em, v. 26, 2019.
- LUTZ, M.; ASCHER, D. *Aprendendo Python, 2*. [S.l.]: Bookman, 2007.
- MORÁN, J. Mudando a educação com metodologias ativas. *Coleção mídias contemporâneas. Convergências midiáticas, educação e cidadania: aproximações jovens*, v. 2, n. 1, p. 15–33, 2015.
- NETO, A. A. *Matemática financeira e suas aplicações*. [S.l.]: Atlas, 2000.
- NETO, A. A. *Matemática financeira e suas aplicações*. 2000.
- PAIVA, T. Y. Aprendizagem ativa e colaborativa: uma proposta de uso de metodologias ativas no ensino da matemática. 2016.
- PAPERT, S.; HAREL, I. Situating constructionism. *constructionism*, v. 36, n. 2, p. 1–11, 1991.
- PITZER, L. C. *Financiamentos e investimentos*:: uma proposta para o ensino médio. Dissertação (Mestrado) — Universidade do Estado de Santa Catarina, 2018.
- POMPEO, J. N.; HAZZAN, S. *Matemática financeira. Saraiva–São Paulo–2001*, 2007.
- PYTHON. *The Python Tutorial 3.8*. Disponível em: <<http://python.org/>>.
- ROCHA, K. C. d.; BASSO, M. V. d. A.; NOTARE, M. R. Aproximações teóricas entre pensamento computacional e abstração reflexionante. *RENTE: Revista Novas Tecnologias na Educação. Porto Alegre. Vol. 18, n. 02 (dez. 2020), p. 581-590.*, 2020.
- SAMANEZ, C. P. *Matemática financeira aplicações a anlise de investimento*. 3º ed prentice-hall. São Paulo, 2002.
- SILVA, R. O. da; SILVA, I. R. S. Linguagem de programação python. *TECNOLOGIAS EM PROJEÇÃO*, v. 10, n. 1, p. 55–71, 2019.
- SODRE, U. *Matemática Financeira*. Londrina: [s.n.], 2020. Disponível em: <<http://www.uel.br/projetos/matessencial/basico/financeira/prestac.html>>. Acesso em: 23 mar. 2023.

SUMMERFIELD, M. Programação em python 3: uma introdução completa à linguagem python. *Rio de*, 2012.

VIDOR, G. A história da cvm pelo olhar de seus ex-presidentes. *Rio de Janeiro: ANBIMA e BM&FBOVESPA*, p. 7–347, 2016.

ZARPELON, E. et al. O software maple como ferramenta auxiliar no processo ensino-aprendizagem de cálculo para alunos do curso de administração. *Revista ESPACIOS/ Vol. 37 (Nº 33) Año 2016*, 2016.