INTRODUÇÃO À ÁLGEBRA LINEAR

UTILIZANDO A LINGUAGEM DE PROGRAMAÇÃO *PYTHON* E O RECURSO DE DESCRIÇÃO DE IMAGENS PARA DEFICIENTES VISUAIS

AUTORES

Mariza Camargo
Cristiano Bertolini
Guilherme Bernardino da Cunha
Patricia Rodrigues Fortes
Sidnei Renato Silveira

LICENCIATURA EM COMPUTAÇÃO

INTRODUÇÃO À ÁLGEBRA LINEAR

UTILIZANDO A LINGUAGEM DE PROGRAMAÇÃO *PYTHON* E O RECURSO DE DESCRIÇÃO DE IMAGENS PARA DEFICIENTES VISUAIS

AUTORES

Mariza Camargo
Patricia Rodrigues Fortes
Cristiano Bertolini
Sidnei Renato Silveira
Guilherme Bernardino da Cunha

1ª Edição
UAB/CTE/UFSM

UNIVERSIDADE FEDERAL DE SANTA MARIA

Santa Maria | RS
2022

Este caderno foi elaborado pela Coordenadoria de Tecnologia Educacional da Universidade Federal de Santa Maria para os cursos da UAB.

COORDENADORIA DE TECNOLOGIA EDUCACIONAL

COORDENADORA DA CTE
Liziany Müller Medeiros

ELABORAÇÃO DO CONTEÚDO
Mariza Camargo, Patricia Rodrigues Fortes, Cristiano Bertolini, Sidnei Renato Silveira e Guilherme Bernardino da Cunha

REVISÃO LINGUÍSTICA
Grazielle da Silva dos Santos

APOIO PEDAGÓGICO
Maria Aparecida Nunes Azzolin
Patrícia Nunes Pezzini

EQUIPE DE DESIGN
Ana Luiza Mozzaquatro de Mattos – Capa
Daniela Pereira Barbosa – Diagramação

PROJETO GRÁFICO
Ana Letícia Oliveira do Amaral

C173i Camargo, Mariza
Introdução à álgebra linear [recurso eletrônico] : utilizando a linguagem de programação Python e o recurso de descrição de imagens para deficientes visuais / Mariza Camargo ...[et al.]. – 1. ed. – Santa Maria, RS : UFSM, CTE, 2022.
1 e-book : il.

Este caderno foi elaborado pela Coordenadoria de Tecnologia Educacional da Universidade Federal de Santa Maria para os cursos da UAB
Acima do título: Licenciatura em Computação
ISBN 978-65-88403-48-8

1. Ciência da Computação 2. Matemática 3. Álgebra Linear 4. Linguagem de Programação Python I. Universidade Aberta do Brasil II. Universidade Federal de Santa Maria. Coordenadoria de Tecnologia Educacional III. Título.

CDU 512.64
004.438

Ficha catalográfica elaborada por Lizandra Veleda Arabidian - CRB-10/1492
Biblioteca Central da UFSM

MINISTÉRIO DA EDUCAÇÃO
PÁTRIA AMADA BRASIL GOVERNO FEDERAL

CAPES
UAB UNIVERSIDADE ABERTA DO BRASIL
Universidade Federal de Santa Maria 1960
UFSM Pró-Reitoria de Graduação
COORDENADORIA DE TECNOLOGIA EDUCACIONAL UFSM

APRESENTAÇÃO

Este *e-book* tem o objetivo de trabalhar noções introdutórias da Álgebra, em especial da Álgebra Linear, a qual constitui um ramo da Matemática – uma Ciência que possibilita o estudo de quantidades e formas e se organiza a partir de uma linguagem própria, composta por regras e fórmulas que foram criadas para facilitar a vida humana. A Álgebra é um dos principais ramos da chamada Matemática Pura e estuda a manipulação formal de equações, operações matemáticas, polinômios e estruturas algébricas. Está dividida em: Álgebra Universal, Álgebra Abstrata, Álgebra Elementar, Álgebra Computacional e Álgebra Linear, sendo esta última nosso foco principal.

O presente material é uma adaptação do *e-book* da disciplina de Matemática II, do Curso de Licenciatura em Computação da UAB/UFSM (Universidade Aberta do Brasil/Universidade Federal de Santa Maria), elaborado por Fortes et al. (2019). A proposta deste novo material é a de incluir exemplos práticos, utilizando a linguagem de programação *Python* (para atender aos alunos dos cursos da área de Informática) e a descrição das imagens (permitindo que os deficientes visuais possam acompanhar o material por meio de um software leitor de tela).

Saiba mais: Para a leitura na íntegra do *e-book* da disciplina de Matemática II, do Curso de Licenciatura em Computação da UAB/UFSM, acesse: https://repositorio.ufsm.br/bitstream/handle/1/18331/Curso_Lic-Comp_Matematica-II.pdf?isAllowed=y&sequence=1

Nesse sentido, este *e-book* foi fundamentalmente construído a partir de experiências didáticas de professores da UFSM - Campus Frederico Westphalen/RS, visando a servir como material didático para as disciplinas de Álgebra Linear, voltadas aos cursos superiores da área de Computação. Para tanto, os exemplos práticos de aplicação dos conteúdos estudados são apresentados na linguagem de programação *Python*. O *e-book* foi estruturado para atender, também, pessoas com deficiência visual, contando com a descrição das imagens e quadros apresentados.

Para dar conta desta proposta, a Unidade 1 apresenta uma introdução à linguagem de programação *Python,* com exemplos de códigos-fonte desenvolvidos por meio da IDE (*Integrated Development Environment) IDLE Editor*. A Unidade 2 apresenta o estudo de matrizes, contando com exemplos e atividades práticas. A Unidade 3 aborda o estudo de Determinantes e a Unidade 4, encerrando este *e-book*, apresenta os Sistemas Lineares. Em todas as unidades existe uma série de atividades práticas, sendo que algumas delas foram selecionadas para serem desenvolvidas na linguagem de programação *Python*.

ENTENDA OS ÍCONES

ATENÇÃO: faz uma chamada ao leitor sobre um assunto, abordado no texto, que merece destaque pela relevância.

INTERATIVIDADE: aponta recursos disponíveis na internet (sites, vídeos, jogos, artigos, objetos de aprendizagem) que auxiliam na compreensão do conteúdo da disciplina.

SAIBA MAIS: traz sugestões de conhecimentos relacionados ao tema abordado, facilitando a aprendizagem do aluno.

TERMO DO GLOSSÁRIO: indica definição mais detalhada de um termo, palavra ou expressão utilizada no texto.

SUMÁRIO

1

INTRODUÇÃO À PROGRAMAÇÃO UTILIZANDO PYTHON

INTRODUÇÃO

Nesta unidade, iremos estudar alguns conceitos introdutórios sobre a linguagem de programação *Python*. Esta unidade não pretende abordar conceitos de algoritmos e de lógica de programação. Caso você tenha pouco conhecimento sobre como programar ou queira aprofundar seus conhecimentos, recomendamos que acesse o *e-book* "Introdução a Algoritmos" (PARREIRA et al., 2017).

SAIBA MAIS: Acesse o *e-book* de Introdução a Algoritmos em: https://repositorio.ufsm.br/bitstream/handle/1/15820/Licenciatura_Computacao_introducaoalgoritmos.pdf?sequence=1&isAllowed=y

Vamos estudar, nesta unidade, os conceitos básicos para construir programas utilizando a linguagem de programação *Python*, compreendendo variáveis, expressões lógicas e aritméticas, operadores relacionais, comandos de entrada e de saída de dados, comandos de seleção, comandos de repetição e estruturas de dados fundamentais, tais como vetores (ou matrizes unidimensionais) e matrizes bidimensionais. A intenção desta unidade é a de preparar o leitor para que possa desenvolver os exemplos e atividades práticas propostas nas demais unidades.

Para realizar os exemplos práticos em *Python,* é preciso fazer o *download* dos programas necessários no site *python.org*. A versão da linguagem de programação utilizada neste *e-book* foi a 3.8.0. Os exemplos de códigos-fonte em *Python* estão detalhados para que você possa executá-los por meio do *IDLE Editor*. Além disso, os códigos-fonte também estão disponíveis em Apêndice neste *e-book*.

Iniciaremos os exemplos apresentando os conceitos básicos da linguagem de programação *Python*, necessários para a compreensão das atividades práticas que serão desenvolvidas nas próximas unidades.

INTERATIVIDADE: Acesse o site http://python.org para fazer o *download*.

1.1
UTILIZANDO O *IDLE EDITOR*

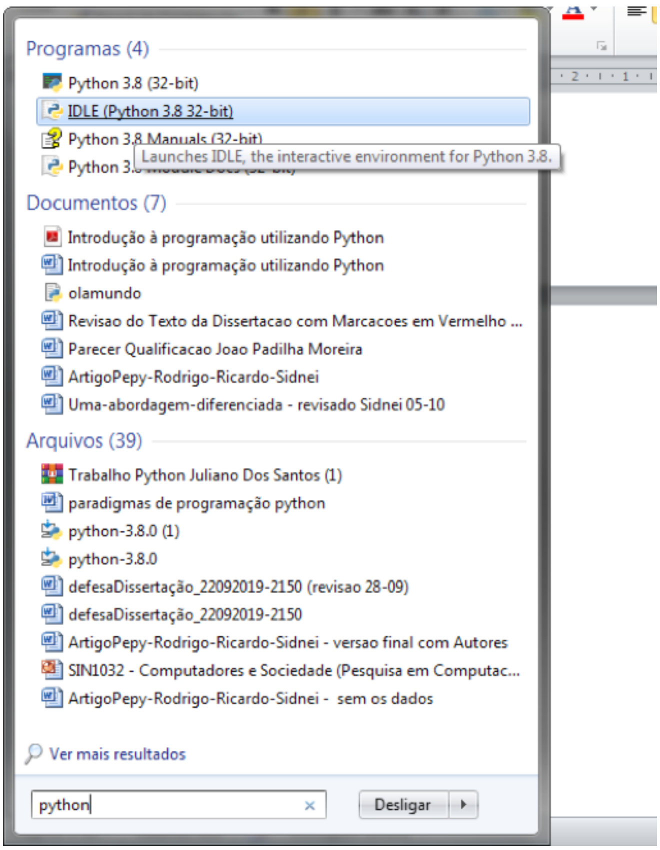
Após realizar o processo de instalação a partir do site oficial, o *Python* poderá ser executado. Você pode utilizar o *Python* por meio do editor *IDLE Editor*.

INTERATIVIDADE: Acesse o site http://python.org para fazer o download.

TERMO DO GLOSSÁRIO: O *IDLE Editor* é uma IDE (*Interface Development Envirnoment*), ou seja, um ambiente de desenvolvimento de software, que permite a construção e execução dos programas criados com a linguagem de programação *Python*.

Basta localizar o editor no menu de programas do Windows (como mostra a Figura 1).

FIGURA 1 – Localizando o *IDLE Editor*

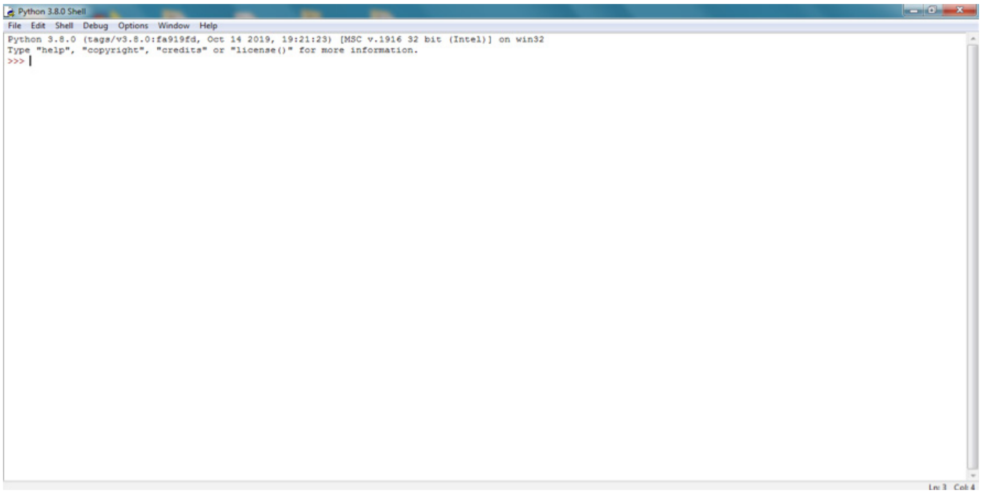


FONTE: Captura de tela do Programa *IDLE Editor* 3.8 (2021).

Descrição de Imagem: Figura 1 - captura da tela do menu *Iniciar* do Sistema Operacional *Windows* contendo: "Programas (4)", "Documentos (7)" e "Arquivos (39)". Abaixo, o campo de busca com a palavra "python" digitada. Obs.: Ao digitar a palavra "python", é possível localizar no menu, no acesso ao editor para a digitação dos códigos-fonte em *Python*, por meio do programa *IDLE* (Python 3.8 32-bit).

O *IDLE Editor*, em execução, é apresentado na Figura 2.

FIGURA 2 - *IDLE Editor* em execução



FONTE: Captura de tela do Programa *IDLE Editor* 3.8 (2021).

Descrição de Imagem: Figura 2- captura da tela do *IDLE Editor* em execução. Na tela, a versão da linguagem *Python* utilizada (versão 3.8.0). Na barra superior, "Phyton 3.8.8 Shell", abaixo, os comandos: "File", "Edit", "Shell", "Debug", "Options", "Window" e "Help". A seguir, o *prompt* de comando, representado por três sinais de maior >>>. Obs.: o *prompt* de comando permite a digitação dos comandos em *Python*.

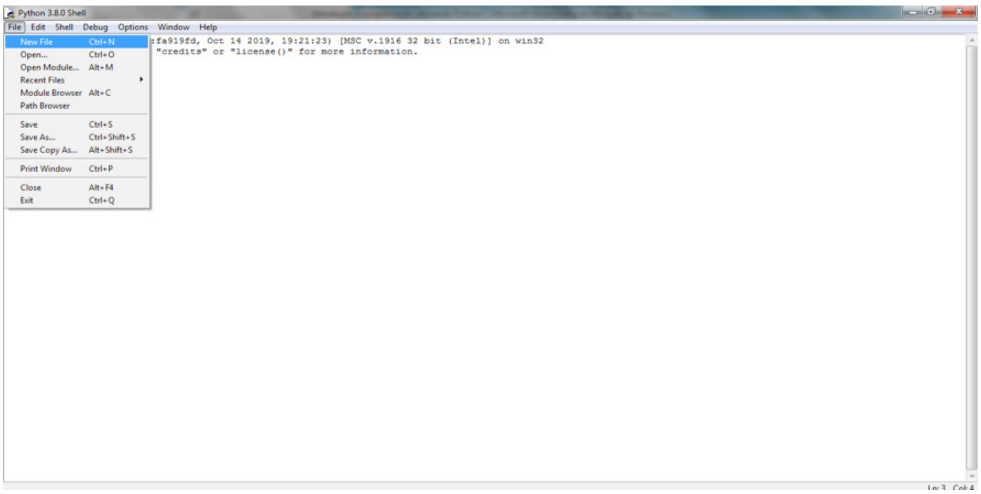
1.2
PRIMEIRO PROGRAMA EM *PYTHON*

Vamos iniciar o estudo introdutório da linguagem de programação *Python* criando um primeiro programa com o *IDLE Editor,* apresentado na Figura 2. Tradicionalmente, quando iniciamos o estudo de uma nova linguagem de programação (já conhecendo os conceitos de algoritmos e lógica de programação), criamos um programa denominado *Hello World* ou *Olá Mundo*. Para criarmos este programa, vamos utilizar o comando *print*, para mostrarmos uma mensagem na tela.

INTERATIVIDADE: Conheça mais informações sobre a linguagem de programação *Python*, acessando o *link* python.org.

No *IDLE Editor,* vamos selecionar a opção *New File*, no menu *File*, como mostra a Figura 3.

FIGURA 3 - Novo arquivo no *IDLE Editor*

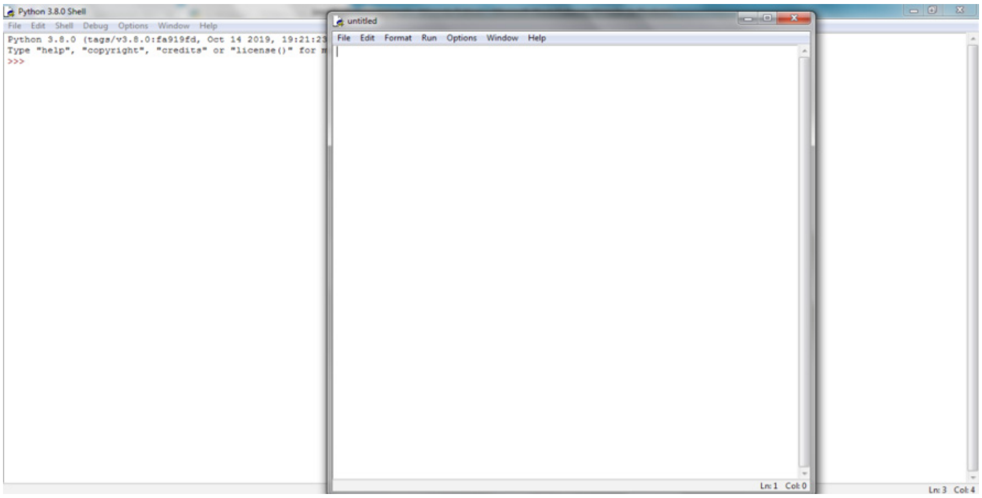


FONTE: Captura de tela do Programa *IDLE Editor* 3.8 (2021).

Descrição de Imagem: Figura 3- captura da tela do *IDLE Editor*. Na barra superior, os comandos: "File" (selecionado), "Edit", "Shell", "Debug", "Options", "Window" e "Help". Abaixo, à esquerda, uma nova janela com vários comandos e a opção "New File" selecionada em azul.

Assim, será aberta uma nova janela para que possamos digitar o código-fonte do nosso programa, como mostra a Figura 4.

FIGURA 4 - Tela para edição do novo arquivo no *IDLE Editor*



FONTE: Captura de tela do Programa *IDLE Editor* 3.8 (2021).

Descrição de Imagem: Figura 4- captura da tela de um novo arquivo no *IDLE Editor*. Na barra superior, o título "untitled*";* abaixo, uma barra contendo os comandos: "File", "Edit", "Format", "Run", "Options", "Window" e "Help". Abaixo, o restante da janela em branco.

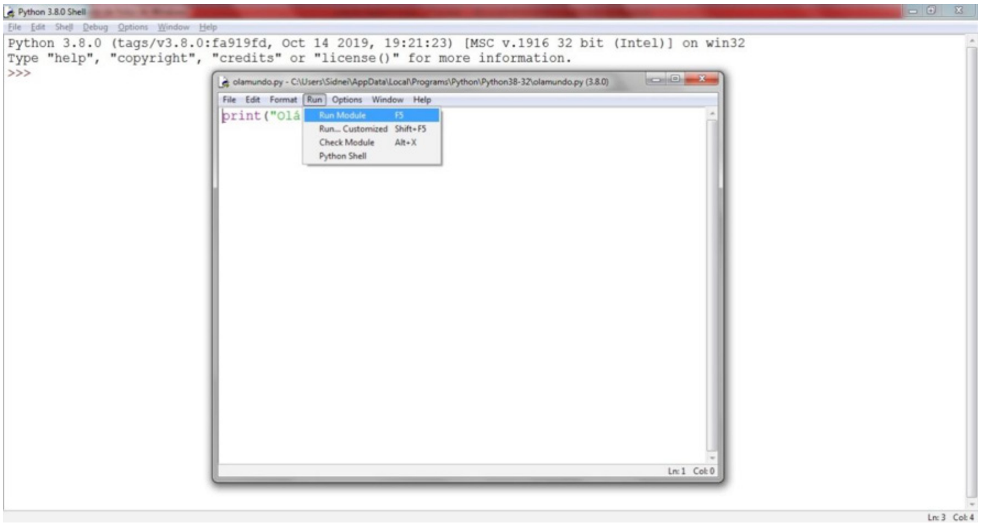
VAMOS PRATICAR

Na janela apresentada na Figura 4, com o nome *untitled*, devemos digitar o código-fonte do nosso programa. Vamos digitar o comando *print ("Olá mundo!")*. Depois, então, salvaremos o arquivo com o nome *olamundo*. A extensão *default* (padrão) dos programas escritos em linguagem *Python* é *.py.*

Após digitarmos o comando, vamos salvar o arquivo utilizando a opção Save As do menu File. Para executar o programa criado, devemos clicar no menu Run e clicarmos na opção Run Module, como mostra a Figura 5.

INTERATIVIDADE: Acesse o *IDLE Editor* para realizar a atividade proposta.

Figura 5 - Executando um programa

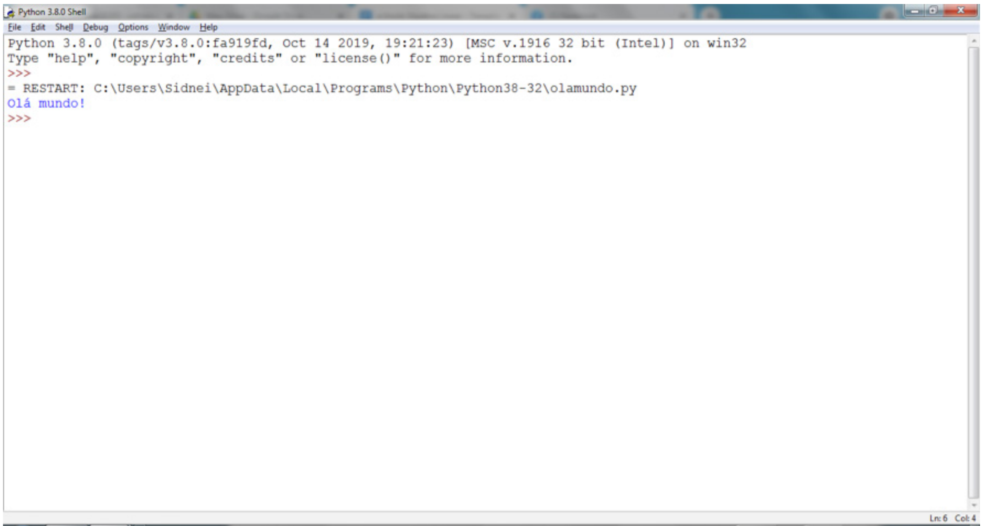


Fonte: Captura de tela do Programa *IDLE Editor* 3.8 (2021).

Descrição de Imagem: Figura 5- captura da tela do menu "Run" do *IDLE Editor*, sendo que a opção "Run Module" está selecionada. Na barra superior, os comandos: "File", "Edit", "Format", "Run", "Options", "Window" e "Help". Abaixo, digitado "print" ("Olá mundo!"). Restante da janela em branco.

A Figura 6 apresenta o resultado da execução do programa *olamundo.py*.

Figura 6 - Resultado da execução do programa *olamundo.py*



Fonte: Captura de tela do Programa *IDLE Editor* 3.8 (2021).

Descrição de Imagem: Figura 6- captura da tela do *IDLE Editor*, contendo o resultado da execução do programa criado anteriormente. Na tela, a mensagem em azul "*Olá mundo*"; na próxima linha, o *prompt* do editor, representado, em vermelho, por três sinais de maior >>>.

Você pode abrir um arquivo salvo anteriormente, por meio do menu *File*, opção *Open*.

1.3 CARACTERÍSTICAS IMPORTANTES DA LINGUAGEM DE PROGRAMAÇÃO *PYTHON*

Algumas características importantes, para iniciarmos o estudo da linguagem de programação *Python*, compreendem:

- não é necessário declarar as variáveis. Basta atribuir um valor para uma variável que a mesma será automaticamente declarada (declaração implícita) e assumirá o tipo de dado de acordo com o que for atribuído à mesma;

Exemplo:

Valor =10

A variável *Valor* (ou identificador *Valor)* recebe o valor inteiro 10 e a variável passa a ter o tipo de dados inteiro atrelado à mesma.
Nome = *"José"*
A variável *Nome* recebe a *string* (cadeia de caracteres) *José*

- a linguagem *Python* é *case sensitive*, ou seja, há diferenciação em utilizar letras maiúsculas ou minúsculas. Por exemplo: os identificadores "*Valor*" e "*valor*" não são iguais, pois *Valor* inicia com letra maiúscula e *valor* com letra minúscula. Neste caso, são dois identificadores diferentes;

- a indentação faz parte da sintaxe da linguagem;

Exemplo: na instrução abaixo, há um erro de sintaxe, pois o comando *print* não está corretamente indentado, ou seja, não faz parte do comando *if*.

```
If x>8: print("x é maior que 8")
```

O código-fonte correto é (o comando *print* tem que estar *dentro* do *if*):

```
If x>8:
  print("x é maior que 8")
```

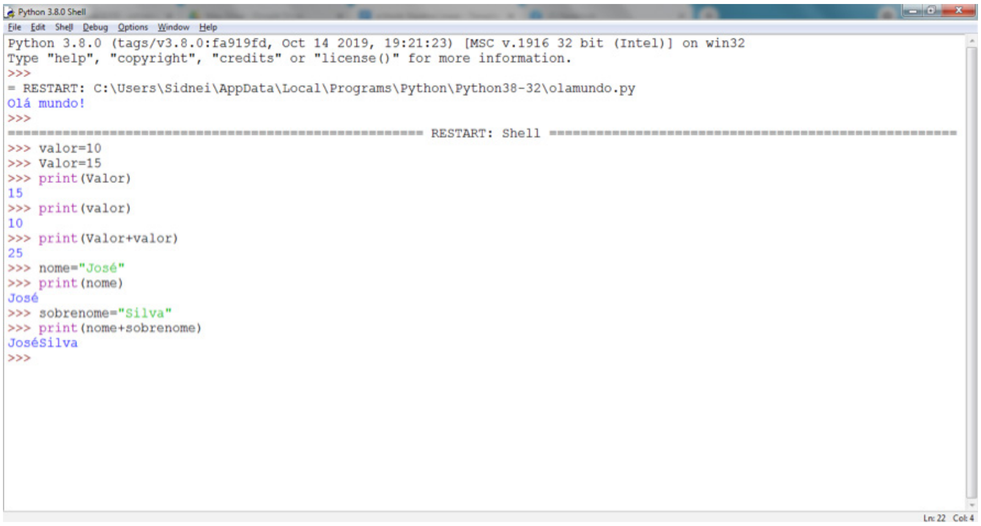
Após digitar o comando *if* e pressionar a tecla *Enter*, o *Python* já irá indentar a próxima linha.

- nas expressões condicionais que compõem as instruções *if*, *for* e *while*, o uso de parênteses é opcional.

A Figura 7 mostra alguns exemplos de atribuição de valores a variáveis e impressão dos valores das mesmas, bem como alguns resultados de expressões aritméticas.

ATENÇÃO: O código-fonte deste exemplo encontra-se no Apêndice deste *e-book*.

FIGURA 7 - Exemplos em *Python*



```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Sidnei\AppData\Local\Programs\Python\Python38-32\olamundo.py
Olá mundo!
>>>
===================================================== RESTART: Shell ====================================================
>>> valor=10
>>> Valor=15
>>> print(Valor)
15
>>> print(valor)
10
>>> print(Valor+valor)
25
>>> nome="José"
>>> print(nome)
José
>>> sobrenome="Silva"
>>> print(nome+sobrenome)
JoséSilva
>>>
Ln: 22 Col: 4
```

FONTE: Captura de tela do Programa *IDLE Editor* 3.8 (2021).

Descrição de Imagem: Figura 7- captura da tela do *IDLE Editor*, contendo o resultado da execução dos comandos em *Python* descritos a seguir.

*O sinal de >>> está em vermelho e a palavra "print" em roxo todas as vezes em que aparecem.

- Analisando a Figura 7 temos os seguintes exemplos:

>>> *valor=10*
Indicando a atribuição do número inteiro 10 à variável *valor;*

>>> *Valor =15*
Indicando a atribuição do número inteiro 15 à variável *Valor* (note que são variáveis diferentes, pois o 2º nome valor está com a letra V em maiúsculo);

>>>*print(valor)*
>>>*print(Valor)*
Realiza a impressão dos conteúdos das variáveis *valor* e *Valor;*

>>>*print(Valor+valor)*
Realiza a impressão da soma dos conteúdos das variáveis *valor* e *Valor;*

>>>nome="José" (em verde)
Atribui o nome *José* à variável *nome;*

>>>sobrenome="Silva" (em verde)
Atribui o sobrenome *Silva* à variável *sobrenome;*

>>>print(nome+sobrenome)
Faz a impressão da soma (concatenação) das *strings nome* e *sobrenome.*

Vamos praticar
Utilizando o *IDLE Editor*, digite e execute os comandos estudados nesta seção.

Interatividade: Acesse o *IDLE Editor* para realizar a atividade proposta.

1.4 OPERADORES ARITMÉTICOS E EXPRESSÕES ARITMÉTICAS

Os operadores aritméticos básicos da linguagem *Python* são: + (adição), - (subtração), * (multiplicação) e / (divisão).

Vamos praticar
Você pode fazer alguns testes diretamente no *prompt* do *IDLE* do *Python* e ver os resultados: 2+2 3-4 5*6 2.0*5 3.0*-2 10/3 2*4+5 2*(4+5)

Interatividade: Acesse o *IDLE Editor* para realizar a atividade proposta.

Também é possível utilizar o comando *print* para mostrar o resultado de operações aritméticas:

print(2+2)
*print(2.0*5)*

Podemos colocar uma mensagem junto com as operações aritméticas, por exemplo:

*print("Resultado",2*4+5)*
print("Resultado",2(4+5))*

O uso do ponto e vírgula (;) no final de uma instrução é opcional. Entretanto, se existir mais de uma instrução em uma mesma linha do código-fonte, o ponto e vírgula passa a ser obrigatório. Por exemplo:

*print("Soma",2+2);print("Multiplicação",2*2)*

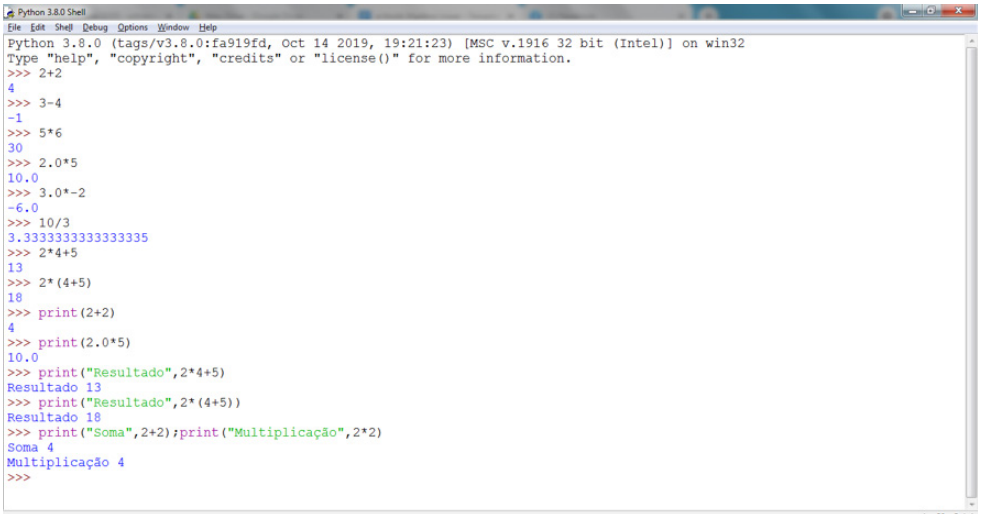
Os resultados aparecerão em linhas distintas:

Soma 4
Multiplicação 4

A Figura 8 apresenta a janela do *IDLE Editor* com a execução dos exemplos desta subunidade.

ATENÇÃO: O código-fonte dos exemplos da Figura 8 estão disponíveis no Apêndice deste *e-book*.

FIGURA 8 - Exemplos de expressões aritméticas em *Python*



FONTE: Captura de tela do Programa *IDLE Editor* 3.8 (2021).

Descrição de Imagem: Figura 8 - captura de tela do *IDLE Editor*, contendo exemplos de comandos desta seção ("2+2", "3-4", "2.0x5", entre outros) e seus respectivos resultados.

1.5
OPERADORES RELACIONAIS E LÓGICOS

Os operadores relacionais da linguagem *Python* são: > (maior que), >= (maior ou igual), < (menor que), <= (menor ou igual), == (igual), e != (diferente).

VAMOS PRATICAR

Vamos realizar alguns exemplos de aplicação destes operadores. Inicialmente, digite as seguintes linhas de código diretamente no *shell* do *IDLE Editor:*

a=5
b=10
a>10
a>b
b<10
b<=10
b>a
a<b
b<a
a==b
a!=b

INTERATIVIDADE: Acesse o *IDLE Editor* para realizar a atividade proposta.

Devemos lembrar que o símbolo de igualdade (=) não significa uma operação de comparação, e sim de atribuição de um valor a uma variável.

Exemplo**:**

- a=10 # o valor 10 está sendo atribuído à variável *a*
- a==10 # estamos perguntando (comparando) se o valor da variável *a* é igual a 10

Os operadores lógicos mais comuns são o *and* e o *or*, e seu funcionamento é igual ao de outras linguagens de programação. Quando o *and* (e) é utilizado, todas as condições devem ser satisfeitas para que a expressão lógica retorne verdadeiro (*True*). Utilizando o operador lógico *or* (ou), pelo menos uma das condições deve ser satisfeita para que a expressão lógica retorne *True*.

VAMOS PRATICAR

Seguem mais alguns exemplos que você pode testar na janela *shell* do *IDLE Editor*:
a=10
b=5
c=7
(a>b) and (b<a)
(a>b) or (b<a)
(a<c) and (b>c)
(a>c) or (b>c)
((a>c) and (b<a)) or (b>c)
((a<c) or (b<a)) or (b>c)
((a<c) or (b<a)) and (b>c)

INTERATIVIDADE: Acesse o *IDLE Editor* para realizar a atividade proposta.

Analisando estes exemplos, temos (sendo que *a* é igual a 10, *b* é igual a 5 e *c* tem armazenado o valor 7):

- *(a>b) and (b<a) # a* é maior que *b* – *True* e *b* é menor que *a* – *True*, *True and True* resulta *True;*

- *(a>b) or (b<a) # a* é maior que *b* – *True* e *b* é menor que *a True*, *True or True* resulta *True*;

- *(a>c) and (b>c) # a* é maior que *c* – *True* e *b* é maior que *c* – *False*, *True and False* retorna *False;*

- *(a<c) or (b<c) # a* é maior que *c* – *True* e *b* é menor que *c* – *True*, *True or True* retorna *True;*

- *((a>c) and (b<a)) or (b>c)* # neste exemplo usamos os parênteses para delimitar as expressões, ou seja, a resposta da primeira expressão com *and* será comparada com a segunda expressão com *or*: *a* é maior que *c* – *True* e *b* é menor que *a* - *True* – *True and True* retorna *True*. A expressão *b* é maior que *c* retorna *False*. Finalmente, *True or False* retorna *True;*

- *((a<c) or (b<a)) or (b>c)* # neste exemplo usamos os parênteses para delimitar as expressões, ou seja, a resposta da primeira expressão com *or* será comparada com a segunda expressão, também usando *or*: *a* é menor que *c* – *False* e *b* é menor que *a*, *True* – *False or True* retorna *True*. A expressão *b* é maior que *c* retorna *False*. Finalmente, *True or False* retorna *True;*

- *((a<c) or (b<a)) and (b>c)* # neste exemplo usamos os parênteses para delimitar as expressões, ou seja, a resposta da primeira expressão com *or* será comparada com a segunda expressão, usando *and*: *a* é menor que *c* – *False* e *b* é menor que *a*, *True* – *False or True* retorna *True*. A expressão *b* é maior que *c* retorna *False*. Finalmente, *True and False* retorna *False*.

1.6
PRINCIPAIS COMANDOS DA LINGUAGEM *PYTHON*

Esta subunidade apresenta os principais comandos da linguagem de programação *Python*, que serão utilizados nos exemplos e atividades práticas deste *e-book*.

1.6.1 Comandos de Seleção

Seleção Simples

if condição_1:
    comando_1

Se o resultado da *condição_1* for *True,* o comando da linha abaixo (*comando_1*), desde que esteja corretamente indentado, será executado. Por exemplo:

```
a=5
if a>=5:
    print("o valor de a é maior ou igual a 5")
```

O comando *print* deverá ficar alinhado com o sinal de maior da linha anterior. O resultado da execução será a mensagem: *o valor de a é maior ou igual a 5.*

Seleção Composta

```
if condição_1:
    comando_1
else:
    comando_2
```

Se o resultado da *condição_1* for *True,* o comando da linha abaixo (*comando_1),* desde que esteja corretamente indentado, será executado. Caso contrário, se o resultado da *condição_1* for *False,* o *comando_2* será executado.

Exemplo:

```
a=3
if a>=5:
    print("o valor de a é maior ou igual a 5")
else:
    print("o valor de a é menor que 5")
```

O resultado da execução será a mensagem: *o valor de a é menor que 5.*

Seleção de Múltipla Escolha

```
if condição_1:
    comando_1
elif condição_2:
    comando_2
elif condição_3:
    comando_3

    ...

elif condição_n
    comando_n
else
    comando_else
```

O comando *elif* é utilizado para verificação de vários casos, conhecida como seleção de múltipla escolha. Se a *condição_1* resultar *True*, será executado o *comando_1*. Se a *condição_2* resultar *True*, será executado o *comando_2* e assim sucessivamente, até o *comando_n*. Caso nenhuma das condições resulte *True*, será executado, após *else*, o *comando_else*.

Exemplo**:** supondo o seguinte trecho de código:

```
a=3
b=5
if a>b:
    print("o valor de a é maior que o valor de b")
elif a<b:
    print("o valor de a é menor que o valor de b")
elif a>=b:
    print("o valor de a é maior ou igual ao valor de b")
elif a<=b:
    print("o valor de a é menor ou igual ao valor de b")
```

A resposta, durante a execução, será: *o valor de a é menor que o valor de b*, ou seja, a primeira condição que resultar *True* na seleção encadeada.

Vamos praticar

Vamos testar os exemplos desta subunidade. Crie um novo arquivo utilizando a opção *New* do menu *File*. Digite, na janela de edição, o seguinte trecho de código-fonte:

```
# Exemplo de seleção simples
a=5
if a>=5:
  print("o valor de a é maior ou igual a 5")
# Exemplo de seleção composta
a=3
if a>=5:
   print("o valor de a é maior ou igual a 5")
else:
   print("o valor de a é menor que 5")
# Exemplo de seleção de múltipla escolha
a=3
b=5
if a>b:
   print("o valor de a é maior que o valor de b")
elif a<b:
   print("o valor de a é menor que o valor de b")
elif a>=b:
   print("o valor de a é maior ou igual ao valor de b")
elif a<=b:
    print("o valor de a é menor ou igual ao valor de b")
```

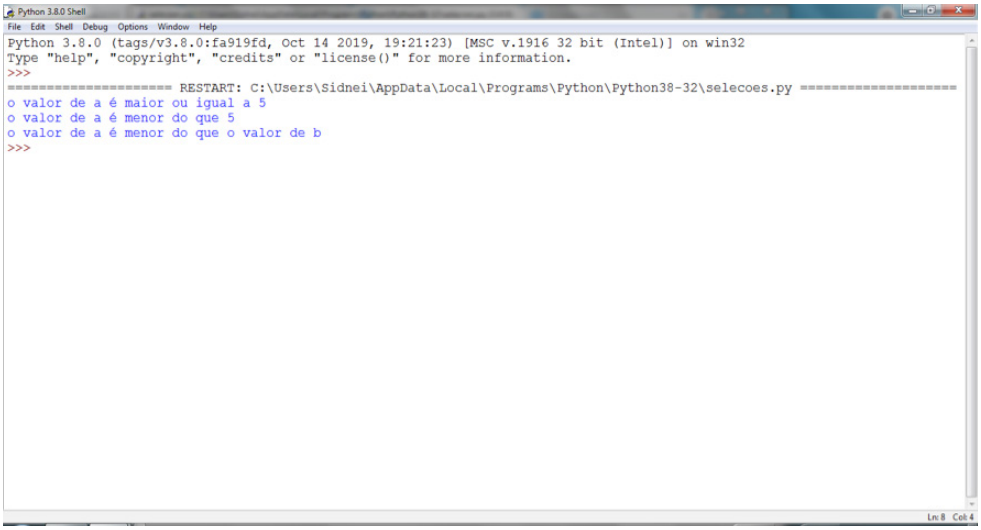
Interatividade: Acesse o *IDLE Editor* para realizar a atividade proposta.

Salve o arquivo utilizando a opção *Save As* do menu *File*. Para executar o código-fonte, utilize a opção *Run Module* do menu *Run*.

A Figura 9 apresenta, respectivamente, os resultados dos exemplos de seleção simples, seleção composta e seleção de múltipla escolha de execução deste código-fonte.

Atenção: O código-fonte dos exemplos da Figura 9 está disponível no Apêndice deste *e-book*.

FIGURA 9 - Exemplos de comandos de seleção em *Python*



FONTE: Captura de tela do Programa *IDLE Editor* 3.8 (2021).

Descrição de Imagem: Figura 9- captura de tela do *IDLE Editor*, contendo os exemplos de comandos de seleção desta seção e seus respectivos resultados ("o valor de a é maior ou igual a 5", "o valor de a é menor ou igual a 5", "o valor de a é menor que o valor de b").

1.6.2 Comandos de Repetição

Comando For

O comando *for* permite percorrer os itens de uma coleção (por exemplo, uma *string* ou uma estrutura de dados do tipo *lista*). Para cada um dos itens da coleção, será executado o comando declarado na estrutura de repetição (*loop*). Ao percorrer a lista de valores, a variável definida no comando *for* receberá, a cada iteração, um dos itens da coleção. A sintaxe do comando *for* é:

for variavel in lista:
    comandos

Por exemplo, utilizando o seguinte trecho de código:

nomes=[‘Ana’,’João’,’José’,’Maria’]
for n in nomes:
    print(n)

Inicialmente, criamos uma lista, denominada *nomes*, contendo quatro nomes (*Ana, João, José* e *Maria*). Após, utilizando o comando *for*, vamos percorrer toda a lista e, a cada iteração, um dos nomes, sequencialmente, será armazenado na variável *n*, que será impressa na tela. O resultado da execução deste trecho de código será a impressão dos nomes conforme estão armazenados na lista: *Ana, João, José* e *Maria*.

Podemos incluir o comando *else* ao final do comando *for*. Assim, um comando ou bloco de comandos será executado ao final da iteração. Por exemplo:

```
nomes=['Ana','João','José','Maria']
for n in nomes:
    print(n)
else:
    print("A lista de nomes foi impressa com sucesso!")
```

Também podemos usar o comando *for* para escrever todas as letras de um *string* (como se fosse soletrar as letras). Isso é possível porque uma *string* é uma lista de caracteres. Por exemplo:

```
for caracteres in 'Python':
    print('Letra:', caracteres)
```

Este trecho de código exibirá como resultado:

Letra: P
Letra: y
Letra: t
Letra: h
Letra: o
Letra: n

Vamos praticar

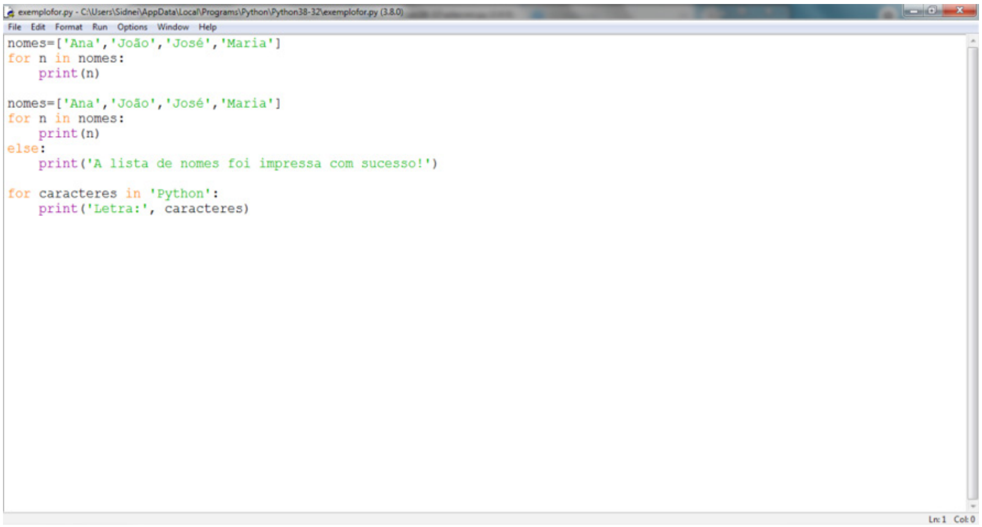
Como fizemos na seção anterior, em que estudamos exemplos de comandos de seleção, crie um novo arquivo no editor do *Python* e digite os exemplos de código-fonte com o comando *for*.

Interatividade: Acesse o *IDLE Editor* para realizar a atividade proposta.

A Figura 10 apresenta o arquivo contendo o código-fonte no editor.

Atenção: O código-fonte dos exemplos da Figura 10 está disponível no Apêndice deste *e-book*.

Figura 10 - Código-fonte dos exemplos com o comando *for*



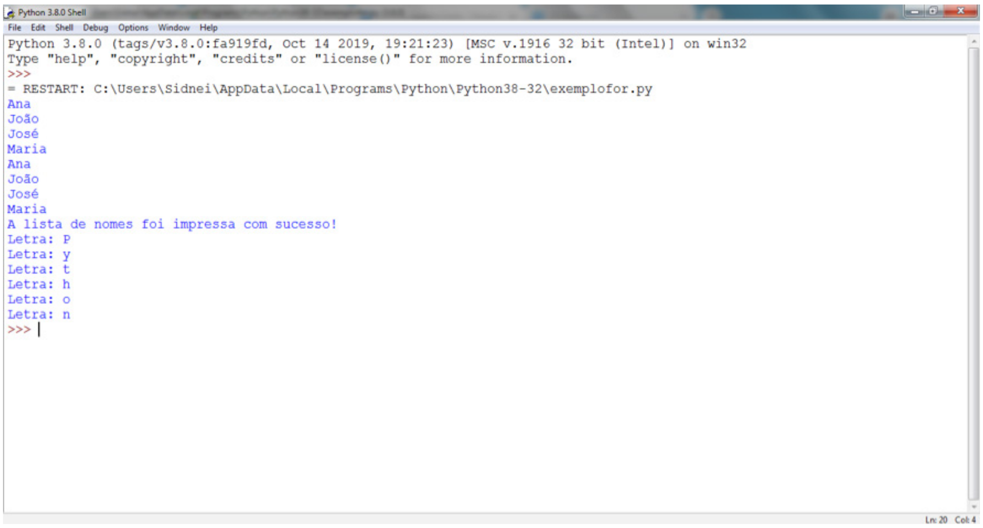
Fonte: Captura de tela do Programa *IDLE Editor* 3.8 (2021).

Descrição de Imagem: Figura 10- captura de tela do *IDLE Editor*, contendo o código-fonte em *Python* dos exemplos ("Ana, João, José, Maria") realizados com o comando *for* nesta seção.

A Figura 11 apresenta o resultado da execução dos respectivos comandos.

Atenção: O código-fonte dos exemplos da Figura 11 está disponível no Apêndice deste e-book.

Figura 11 - Resultado da execução dos exemplos com o comando *for*



Fonte: Captura de tela do Programa *IDLE Editor* 3.8 (2021).

Descrição de Imagem: Figura 11- captura de tela do *IDLE Editor*, contendo os resultados da execução (letras: p, y, t, h, o, n) dos exemplos ("Ana, João, José, Maria") realizados com o comando *for* nesta seção.

Comando While

O comando *while* faz com que um trecho de código seja repetido enquanto uma condição for verdadeira. Quando o resultado da condição for falso, a execução da repetição é interrompida (saindo do *loop*), passando para o próximo comando após o *while*.

A sintaxe do comando *while* é:

```
while (condição):
   comandos
```

Por exemplo:

```
contador=0
while (contador<5):
   print(contador)
   contador=contador+1
```

Neste exemplo de código, estamos repetindo, por meio do *while*, os comandos *print* e incrementando o contador 5 vezes, utilizando uma variável contadora *(contador)*. Inicializamos a variável *contador* com o valor zero. Enquanto o valor do contador for menor que 5 (enquanto o resultado desta condição for *True*), o trecho dentro do *while* será repetido. Lembre-se de que a endentação do código-fonte faz parte das regras de sintaxe da linguagem *Python*. Sendo assim, os comandos dentro do *while* devem estar corretamente indentados para que sejam executados na repetição. O resultado da execução deste trecho é apresentado a seguir:

```
0
1
2
3
4
```

Vamos praticar

Como fizemos na seção anterior, em que estudamos o comando *for*, crie um novo arquivo no editor do *Python* e digite os exemplos de código-fonte com o comando *while*.

Interatividade: Utilize o *IDLE Editor* para realizar a atividade proposta.

1.6.3 Conversão entre Tipos de Dados

Nesta subunidade, vamos estudar alguns comandos que permitem a conversão entre diferentes tipos de dados, que são úteis para o desenvolvimento de exemplos práticos em *Python*.

Função *str*

A função *str* converte um dado para o formato de texto (*string*).

Vamos praticar
Vamos realizar os exemplos a seguir*:* *str(12)* *a=3* *str(a)* *soma=a+5* *print(soma)* *a=str(a)* *print(a) #* *soma=a+5*

Interatividade: Utilize o *IDLE Editor* para realizar a atividade proposta.

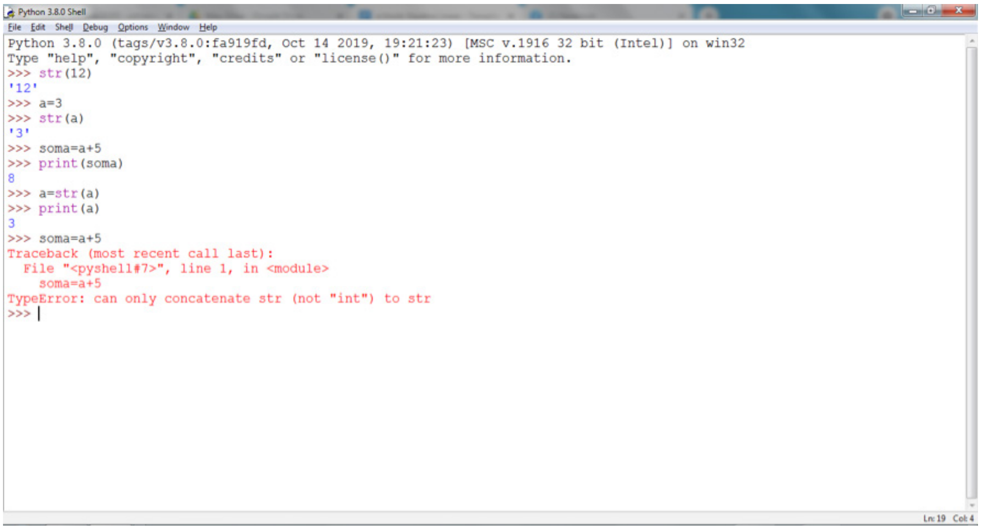
Vamos entender o que faz cada um dos comandos dos exemplos realizados:

» *str(12)* # resultado: '12', ou seja, converte o número 12 para *string*, colocando o valor entre apóstrofos
» *a=3*
» *str(a)* # resultado '3', converte o valor da variável *a* para *string*
» *soma=a+5*
» *print(soma)* # resultado: 8 (porque a variável *a* permaneceu como um número inteiro).
» *a=str(a)* # neste exemplo, estamos convertendo a variável *a* para uma *string* e alterando o seu valor por meio da atribuição. A partir de agora, a variável *a* não é mais um número inteiro.
» *print(a)* # resultado 3
» *soma=a+5* # Erro! A variável *a* passou a ser uma *string* na linha *a=str(a)*. Sendo assim, ela não pode ser manipulada como se fosse um número inteiro. Ocorrerá um erro de execução (*TypeError: can only concatenate str (not "int") to str*).

A Figura 12 apresenta os resultados da execução destes exemplos no *IDLE Editor*.

Atenção: O código-fonte dos exemplos da Figura 12 está disponível no Apêndice deste *e-book*.

Figura 12 - Resultados da execução dos exemplos com a função *str*



Fonte: Captura de tela do Programa *IDLE Editor* 3.8 (2021).

Descrição de Imagem: Figura 12- captura de tela do *IDLE Editor*, contendo os comandos e seus respectivos resultados de execução, de acordo com os exemplos realizados com a função *str* nesta seção.

Função *int*

A função *int* converte um dado para um número inteiro.
Por exemplo:
a='33' # o valor 33 entre apóstrofos representa uma *string*
int(a)
a # retorna 33

Função *type*

A função type retorna o tipo (classe à qual pertence o objeto). Exemplos:
a=10
type(a) # retorna class<'int'>
type('Python') # retorna class<'str'>

1.6.4 Atribuições Múltiplas

Em *Python*, é possível atribuir, simultaneamente, valores a mais de uma variável.
Por exemplo:
a=10

b='Python'
a,b = b,a # a variável *a* recebe o valor da variável *b* (a *string Python*) e a variável *b* recebe o valor da variável *a* (o valor 10).
c, d, e = 'sol' # a variável *c* recebe a *string* 's', *d* recebe a *string* 'o' e a variável *e* recebe a *string* 'l', respectivamente.

1.6.5 Entrada de Dados

Uma forma de realizar a entrada de dados em *Python* é por meio do comando *input.* O comando *input* abre o *prompt,* permitindo que o usuário entre com um valor.

VAMOS PRATICAR
Digite o código-fonte abaixo: *idade=input("Digite a sua idade:")*

INTERATIVIDADE: Utilize o *IDLE Editor* para realizar o exemplo proposto.

Supondo que, ao aparecer a mensagem *Digite a sua idade:,* o usuário entre com o valor 10, podemos imprimir o valor da idade digitado e armazenado na variável *idade:*

print(idade)
10

Ao verificarmos o tipo da variável *idade*, temos:

type(idade) # resultado: <class 'str'>

Apesar de termos digitado um valor inteiro (10) para a idade, a variável *idade* é do tipo *string,* ou seja, a entrada de um comando *input* é uma *string*. Para manipularmos a *idade* como um número inteiro, caso seja necessário em nosso programa, será preciso utilizar a função *int* para convertê-la.

1.6.6 Saída de Dados

O comando *print* é utilizado para a saída de dados.

Vamos praticar

Digite o código-fonte do exemplo a seguir e execute-o. O seguinte trecho de código converte uma temperatura de graus *Celsius* para *Fahrenheit*:

c=25
*f= 1.8*c + 32*
print(c,' graus Celsius equivalem a ', f, ' graus Fahrenheit')

Interatividade: Acesse o *IDLE Editor* para realizar a atividade proposta.

O resultado será: *25 graus Celsius equivalem a 77 graus Fahrenheit.*

1.7
EXEMPLO DE CALCULADORA BÁSICA EM *PYTHON*

Nesta subunidade, apresentamos um exemplo de um programa com mais recursos, que foram estudados neste capítulo. Temos o seguinte código-fonte no arquivo denominado *calculadora.py*:

ATENÇÃO: O código-fonte do arquivo *calculadora.py* encontra-se no Apêndice deste *e-book*.

```
#Calculadora Básica

num1=int(input("Digite um número:"))

num2=int(input("Digite outro número:"))

operacao=input("Escolha a operação: + - * / :")

if (operacao=="+"):

    print("Soma:",num1+num2

if (operacao=="-"):

    print("Subtração:",num1 - num2)

if (operacao=="*"):

    print("Multiplicação:",num1*num2)

if (operacao=="/"):

    print("Divisão:",num1/num2)
```

VAMOS PRATICAR
Crie um novo arquivo e digite o código-fonte de exemplo. Salve o arquivo com o nome de *calculadora.py*. Execute o programa para verificar os seus resultados.

Interatividade: Utilize o *IDLE Editor* para realizar a atividade proposta.

Analisando o código-fonte deste exemplo, temos:

» o símbolo # representa um comentário;
» o comando *input* permite a entrada de dados. Como os dados digitados são armazenados no formato de *strings,* é preciso convertê-los para que possamos utilizá-los como números inteiros. Sendo assim, aplicamos a função de conversão *int;*
» o comando *if* faz uma seleção. No caso, de acordo com o operador escolhido (adição, subtração, multiplicação ou divisão), o comando *print* mostra o resultado da operação.

1.8
UTILIZAÇÃO DE FUNÇÕES MATEMÁTICAS: A BIBLIOTECA *MATH*

A biblioteca *Math* permite a utilização de diversas funções matemáticas, tais como os cálculos de raiz quadrada, seno e cosseno, por exemplo. Para utilizar esta biblioteca, é preciso importá-la, por meio do comando *import math*.

Vamos praticar

Por exemplo, supondo que queiramos calcular a raiz quadrada de um número, podemos utilizar o código-fonte a seguir. Crie um novo arquivo no *IDLE Editor,* digite, salve o arquivo e execute o programa para ver o resultado.

```
import math
numero=input('Digite um número:')
numero=int(numero)
print('A raiz quadrada de ',numero,' é igual a ', math.sqrt(numero))
```

Interatividade: Acesse o *IDLE Editor* para realizar a atividade proposta.

Supondo que o usuário digite o valor 10 na entrada de dados (*input*), será mostrado o seguinte resultado:

» *A raiz quadrada de 10 é igual a 3.1622776601683795*

Podemos formatar a saída do comando *print*, definindo o número de casas (algarismos) antes e depois da vírgula (ponto decimal). Por exemplo:

```
print('A raiz quadrada de ',numero,' é igual a %3.2f.' %math.sqrt(numero))
```

em que *%3.2f.* significa até 3 algarismos antes do ponto decimal, 2 algarismos após o ponto decimal e *f* significa um número do tipo *float*. A impressão do resultado ficará assim:

» *A raiz quadrada de 10 é igual a 3.16.*

Outras funções da biblioteca Math compreendem:

- *math.cos(número):* retorna o cosseno do número em radianos;
- *math.sin(número)*: retorna o seno do número em radianos;
- *math.tan(número)*: retorna a tangente do número em radianos;
- *math.radians(número)*: converte o ângulo definido em número de graus para radianos;
- *math.hypot(x, y)*: retorna a hipotenusa dos números (catetos) fornecidos em *x* e *y*;
- *math.factorial(número)*: retorna o fatorial do número definido;
- *math.log(número)*: retorna o logaritmo do número definido;
- *math.pow(x,y)*: retorna o valor de *x* elevado à potência *y*.

Atividades – Unidade 1

1) Utilizando o *IDLE Editor*, crie programas em *Python*, baseando-se nos comandos estudados nesta unidade:

a) Crie um programa que, a partir da entrada de três variáveis A, B e C, escreva na tela qual foi o maior valor lido, ou seja, qual é a variável que recebeu o maior valor.
b) Crie um programa que mostre na tela os números de 1 a 50.
c) Crie um programa que leia um valor de entrada e calcule o seno e o cosseno.

2

MATRIZES

INTRODUÇÃO

Para iniciar o estudo da Álgebra Linear, abordaremos as matrizes que, embora tenham surgido no século XIX, continuam a proporcionar experiências inovadoras à Matemática. A denominação "matriz" foi utilizada pela primeira vez em 1850, nos trabalhos do matemático inglês James Joseph Sylvester (1814-1897). Uma matriz é caracterizada como sendo uma tabela, sendo que seus elementos estão dispostos em linhas e colunas.

Mesmo que não percebamos, as matrizes estão presentes em vários contextos de nosso cotidiano, desde cálculos efetuados por um computador ou mesmo na construção civil, o que configura a sua aplicação não só à Matemática, mas também em várias áreas, a exemplo da Informática, Economia, Física, Engenharias, entre outras.

Uma representação matricial, ou um sistema matricial, é muito utilizada(o) na resolução de um sistema linear. Como os sistemas lineares também serão estudados neste *e-book*, na Unidade 4, passaremos agora, preliminarmente, ao estudo das matrizes.

Além do estudo das matrizes, vamos desenvolver exemplos práticos de como utilizar esse tipo de estrutura de dados na linguagem de programação *Python*, por meio de exemplos e exercícios desenvolvidos no *IDLE Editor*. Todos os códigos-fonte utilizados nos exemplos encontram-se em apêndice e são comentados no texto, visando a auxiliar na explicação dos conceitos matemáticos aplicados na programação. Para facilitar a leitura por meio de *softwares* leitores de tela, as matrizes não foram representadas na forma de imagens.

2.1
NOTAÇÃO E PROPRIEDADES DAS MATRIZES

Para contextualizar a abordagem dos tópicos relacionados ao estudo das matrizes, vamos tomar como exemplo a situação fictícia elaborada por Fortes et al. (2019). Consideraram-se as médias finais (hipotéticas) de desempenho escolar de três acadêmicos do Curso de Licenciatura em Computação da UFSM/UAB, referentes a quatro disciplinas curriculares, observando-se o aproveitamento de cada aluno por disciplina. Vejamos a Tabela 1.

Tabela 1 - Notas de três acadêmicos em quatro disciplinas

	Lógica Matemática	Informática Básica	Laboratório de Montagem e Manutenção	Fundamentos Filosóficos e Sociológicos da Educação
Acadêmico 1	6,2	7,1	6,8	8,0
Acadêmico 2	5,7	6,4	7,0	7,3
Acadêmico 3	7,2	8,2	6,9	7,8

Fonte: Fortes et al. (2019).

Descrição de Imagem: Tabela 1- A Tabela 1 possui 4 linhas e 5 colunas. Na primeira linha, ficam as definições das colunas, começando pela coluna 2, temos sequencialmente: "Lógica Matemática", "Informática Básica", "Laboratório de Montagem" e "Manutenção e Fundamentos Filosóficos e Sociológicos da Educação". Na primeira coluna, começando pela linha 2, temos as definições das linhas, sequencialmente: "Acadêmico 1", "Acadêmico 2" e "Acadêmico 3". Como valores das notas dispostos na tabela, temos:

Acadêmico 1: Lógica Matemática: 6,2; Informática Básica: 7,1; Laboratório de Montagem e Manutenção: 6,8; Fundamentos Filosóficos e Sociológicos da Educação: 8,0.

Acadêmico 2: Lógica Matemática: 5,7; Informática Básica: 6,4; Laboratório de Montagem e Manutenção: 7,0; Fundamentos Filosóficos e Sociológicos da Educação: 7,3.

Acadêmico 3: Lógica Matemática: 7,2; Informática Básica: 8,2; Laboratório de Montagem e Manutenção: 6,9; Fundamentos Filosóficos e Sociológicos da Educação: 7,8.

Para localizar a nota de um determinado acadêmico em uma das quatro disciplinas, devemos orientar-nos na linha em que consta o nome daquele acadêmico e na coluna onde estão as notas da disciplina analisada. Por exemplo, se desejamos saber a nota do Acadêmico 3 em Informática Básica, devemos observar o local de intersecção da última linha e da coluna onde constam as notas da referida disciplina, e assim encontraremos a nota 8,2.

Se passarmos a considerar as informações da Tabela 1 somente a partir da disposição das notas em linhas e colunas, usando parênteses ou colchetes, teríamos uma matriz que denominaremos de matriz *A*, com 3 linhas e 4 colunas, que pode ser descrita na linguagem de programação *Python* por meio do comando:

A = [[6.2, 7.1, 6.8, 8.0], [5.7, 6.4, 7.0, 7.3], [7.2, 8.2, 6.9, 7.8]]

Atenção: É válido lembrar que, no comando descrito em *Python*, foi adotado o ponto decimal ("." ponto) como separador decimal em substituição à vírgula decimal ("," vírgula), que é o símbolo oficialmente usado no Brasil. Assim, quando estivermos adotando os comandos da linguagem de programação *Python*, há necessidade de se utilizar o ponto como separador decimal.

Este comando cria a matriz *A* e atribui os valores à mesma. Em *Python*, uma matriz é uma estrutura de dados do tipo lista, pertencente à classe *list*.

Em tabelas assim, os números são chamados de elementos. As colunas são enumeradas da esquerda para a direita e as linhas, de cima para baixo. Esse tipo de tabela, disposta com linhas e colunas, é chamada de matriz.

Uma matriz, por convenção, é representada por uma letra maiúscula do alfabeto da língua portuguesa (A, B, C, ... , X, Y, Z), e seus elementos são representados pela respectiva letra minúscula, acompanhada de dois índices, sendo que o primeiro índice representa a linha e o segundo índice representa a coluna onde aquele determinado elemento está posicionado no interior da matriz.

Logo a seguir, temos uma representação genérica de uma matriz *A*, de ordem *mxn*, ou seja, que apresenta *m* linhas e *n* colunas, ou *A=(aij)mxn*, com *1 <= i <= m*, *1 <= j <= n*, sendo que *i* e *j* pertencem ao conjunto dos números naturais. Lê-se: matriz *A*, de elementos *aij*, de ordem *mxn*. Tal matriz pode ser descrita como:

Amxn = [[a11, a12, a13, …, a1j, …, a1n], [a21, a22, a23, …, a2j, …, a2n], [a31, a32, a33, …, a3j, …, a3n], ..., [ai1, ai2, ai3, …, aij, …, ain], …, [am1, am2, am3, …, amj, …, amn]]

onde o nome da matriz é seguido pela quantidade de linhas e colunas (indicação da ordem da matriz) e os três pontos (as reticências) indicam que há sequência no número estabelecido de linhas e colunas da matriz, para assim tornar a representação genérica.

Vamos praticar

Crie um novo arquivo denominado *matriz1.py* e digite o código-fonte abaixo, onde criamos a matriz *A* e mostramos os seus respectivos valores na tela, utilizando o comando *for*. Neste exemplo, estamos utilizando *range* no comando *for*, que identifica o intervalo de iteração (entre 0 e 3, ou seja, as linhas 0, 1 e 2 da matriz). Utilizando *print(A[linha])*, vamos imprimir cada uma das linhas da matriz.

A = [[6.2, 7.1, 6.8, 8.0], [5.7, 6.4, 7.0, 7.3], [7.2, 8.2, 6.9, 7.8]]
for linha in range(3):
    print(A[linha])

Interatividade: Utilize o *IDLE Editor* para desenvolver a atividade proposta.

Atenção: O código-fonte do arquivo *matriz1.py* encontra-se no Apêndice deste *e-book*.

Atenção: Nos exemplos em *Python*, estamos utilizando as variáveis linha e coluna para representar os índices *i* e *j* utilizados nas descrições das matrizes. Entretanto, como os identificadores podem ser definidos livremente pelo programador, você pode utilizar as variáveis *i* e *j*, ao invés de linha e coluna.

Podemos imprimir, também, cada item da matriz, separadamente. Para isso, precisamos das informações referentes às linhas e colunas. Inclua no código-fonte do arquivo *matriz1.py* o trecho abaixo:

for linha in range(3):

    for coluna in range(4):

        print(A[linha][coluna])

Como os valores do intervalo *range* iniciam em zero, as linhas da matriz seriam 0, 1 e 2 e as colunas, 0, 1, 2 e 3. Para facilitar a visualização, na impressão, podemos incrementar os valores da linha e coluna em 1, para que a representação das linhas fique em 1, 2 e 3 e das colunas em 1, 2, 3 e 4. Abaixo, segue um exemplo de código-fonte, já contando com esse incremento no número de linhas e colunas (apenas para melhorar a visualização para o usuário final do nosso programa).

```
A = [[6.2, 7.1, 6.8, 8.0], [5.7, 6.4, 7.0, 7.3], [7.2, 8.2, 6.9, 7.8]]

print("Dados armazenados na matriz A, de acordo com a linha:")

for linha in range(3):

    print(«Linha da Matriz:»,linha+1)

    print(A[linha])

print("Dados armazenados na matriz A:")

for linha in range(3):

    for coluna in range(4):

        print("Linha:", linha+1, "Coluna:", coluna+1, "Valor armazenado:", A[linha]
        [coluna])
```

Os resultados da execução deste código-fonte são os seguintes:

```
 Dados armazenados na matriz A, de acordo com a linha:

Linha da Matriz: 1
[6.2, 7.1, 6.8, 8.0]
Linha da Matriz: 2
[5.7, 6.4, 7.0, 7.3]
Linha da Matriz: 3
[7.2, 8.2, 6.9, 7.8]
Dados armazenados na matriz A:
Linha: 1 Coluna: 1 Valor armazenado: 6.2
Linha: 1 Coluna: 2 Valor armazenado: 7.1
Linha: 1 Coluna: 3 Valor armazenado: 6.8
Linha: 1 Coluna: 4 Valor armazenado: 8.0
Linha: 2 Coluna: 1 Valor armazenado: 5.7
Linha: 2 Coluna: 2 Valor armazenado: 6.4
Linha: 2 Coluna: 3 Valor armazenado: 7.0
Linha: 2 Coluna: 4 Valor armazenado: 7.3
Linha: 3 Coluna: 1 Valor armazenado: 7.2
Linha: 3 Coluna: 2 Valor armazenado: 8.2
Linha: 3 Coluna: 3 Valor armazenado: 6.9
Linha: 3 Coluna: 4 Valor armazenado: 7.8
```

A Figura 13 apresenta os resultados da execução destes exemplos no *IDLE Editor*.

Figura 13 - Resultados da execução dos exemplos com a matriz *A*

```
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:21:23) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==================== RESTART: C:\Users\Sidnei\AppData\Local\Programs\Python\Python38-32\matriz1.py ====================
Dados armazenados na matriz A, de acordo com a linha:
Linha da Matriz: 1
[6.2, 7.1, 6.8, 8.0]
Linha da Matriz: 2
[5.7, 6.4, 7.0, 7.3]
Linha da Matriz: 3
[7.2, 8.2, 6.9, 7.8]
Dados armazenados na matriz A:
Linha: 1 Coluna: 1 Valor armazenado: 6.2
Linha: 1 Coluna: 2 Valor armazenado: 7.1
Linha: 1 Coluna: 3 Valor armazenado: 6.8
Linha: 1 Coluna: 4 Valor armazenado: 8.0
Linha: 2 Coluna: 1 Valor armazenado: 5.7
Linha: 2 Coluna: 2 Valor armazenado: 6.4
Linha: 2 Coluna: 3 Valor armazenado: 7.0
Linha: 2 Coluna: 4 Valor armazenado: 7.3
Linha: 3 Coluna: 1 Valor armazenado: 7.2
Linha: 3 Coluna: 2 Valor armazenado: 8.2
Linha: 3 Coluna: 3 Valor armazenado: 6.9
Linha: 3 Coluna: 4 Valor armazenado: 7.8
>>> |
```

Fonte: Captura de tela do Programa *IDLE Editor* 3.8 (2021).

Descrição de Imagem: Figura 13- captura de tela do *IDLE Editor*, contendo os resultados da execução do código-fonte em que imprimimos, primeiro, os dados contidos em cada uma das linhas da matriz (Linha da matriz: 1 {6.2, 7.1, 6.8, 8.0}; Linha da matriz: 2 {5.7, 6.4, 7.0, 7.3}; Linha da matriz: 3 {7.2, 8.2, 6.9, 7.8} e, abaixo, os dados de cada posição da matriz (indicando a linha e a coluna onde se encontram), conforme apresentado no texto anterior à figura: Linha: 1 Coluna: 1 Valor armazenado: 6.2; Linha: 1 Coluna: 2 Valor armazenado: 7.1; Linha: 1 Coluna: 3 Valor armazenado: 6.8; Linha: 1 Coluna: 4 Valor armazenado: 8.0; Linha: 2 Coluna: 1 Valor armazenado: 5.7; Linha: 2 Coluna: 2 Valor armazenado: 6.4; Linha: 2 Coluna: 3 Valor armazenado: 7.0; Linha: 2 Coluna: 4 Valor armazenado: 7.3; Linha: 3 Coluna: 1 Valor armazenado: 7.2; Linha: 3 Coluna: 2 Valor armazenado: 8.2; Linha: 3 Coluna: 3 Valor armazenado: 6.9 e Linha: 3 Coluna: 4 Valor armazenado: 7.8.

2.2
TIPO E ORDEM DE UMA MATRIZ

As matrizes recebem certo tipo de nome dependendo da quantidade de elementos em suas linhas e colunas ou apenas por características específicas. A ordem de uma matriz refere-se ao número de linhas e colunas que compõem a matriz. A ordem é apresentada na notação *mxn*, onde *m* é o número de linhas e *n* o número de colunas. Lê-se "*m* por *n*" (ANTON; RORRES, 2001). A seguir, apresentamos alguns tipos de matrizes. Em *Python,* podemos utilizar o comando *len* para sabermos o número de itens de uma matriz (no caso, de uma lista em *Python*).

Matriz Linha

Uma matriz linha é uma matriz de ordem *1xn*, ou seja, com uma única linha.

Exemplo: Considerando na Tabela 1 apenas as notas do Acadêmico 1 nas quatro disciplinas, formamos a matriz linha do tipo 1x4:

C = [6.2, 7.1, 6.8, 8.0]

Matriz Coluna

Uma matriz coluna é uma matriz de ordem *nx1*, ou seja, com uma única coluna.

Exemplo: Considerando na Tabela 1 as notas na disciplina de Lógica Matemática para os três acadêmicos, formamos a matriz:

D = [[6.2], [5.7], [7.2]]

que apresenta ordem 3x1.

Matriz Quadrada

Uma matriz quadrada é uma matriz de ordem *nxn*, ou seja, com o mesmo número de linhas e colunas. Neste caso, podemos simplesmente dizer matriz de ordem *n*.

Exemplo: Considerando na Tabela 1 as notas dos três acadêmicos nas disciplinas de Lógica Matemática, Informática Básica e Laboratório de Montagem e Manutenção, formamos a matriz quadrada:

E = [[6.2, 7.1, 6.8], [5.7, 6.4, 7.0], [7.2, 8.2, 6.9]]

de ordem 3x3 ou simplesmente de ordem 3 (pois o número de linhas é igual ao número de colunas).

Em uma matriz quadrada, definimos a diagonal principal e a diagonal secundária. A diagonal principal é formada pelos elementos *aij*, nos quais *i=j* (a linha é igual à coluna), enquanto que a diagonal secundária é formada pelos elementos *aij*, com *i+j=n+1* (a soma da linha e coluna é igual ao valor de *n* mais 1). No exemplo anterior, na matriz que denominamos *E*, os elementos da diagonal principal são: 6.2, 6.4 e 6.9; e os elementos da diagonal secundária são: 7.2, 6.4 e 6.8. Sendo assim, o elemento E22=6.4 (correspondente à linha 2 e coluna 2) tanto está presente na diagonal principal como também na diagonal secundária.

Vamos praticar

Vamos utilizar o exemplo da matriz *E* (matriz quadrada contendo 3 linhas e 3 colunas) para imprimirmos na tela os valores constantes na diagonal principal e na diagonal secundária. Para tanto, crie um novo arquivo no *IDLE Editor*, denominado *matriz2.py*, e digite o seguinte código-fonte:

```
E = [[6.2, 7.1, 6.8], [5.7, 6.4, 7.0], [7.2, 8.2, 6.9]]
print("Valores da diagonal principal:")
for linha in range(3):
    for coluna in range(3):
        if linha==coluna:
            print("Linha:",linha+1,"Coluna:",coluna+1,"Valor
armazenado:",E[linha][coluna])
print("Valores da diagonal secundária:")
for linha in range(3):
    for coluna in range(3):
        if linha+coluna==3-1:
            print("Linha:",linha+1,"Coluna:",coluna+1,"Valor
armazenado:",E[linha][coluna])
```

Como as linhas e colunas em *Python* iniciam no valor zero, incluímos, como fizemos no exemplo anterior, o incremento em mais 1 na visualização (no comando *print*), e, para verificarmos a condição da diagonal secundária, utilizamos o valor de *n* (3) subtraído do valor 1.

Interatividade: Utilize o *IDLE Editor* para realizar esta atividade.

Atenção: O código-fonte do arquivo *matriz2.py* encontra-se no Apêndice deste *e-book*.

Os resultados impressos após a execução deste código-fonte são os seguintes:

Valores da diagonal principal:
Linha: 1 Coluna: 1 Valor armazenado: 6.2
Linha: 2 Coluna: 2 Valor armazenado: 6.4
Linha: 3 Coluna: 3 Valor armazenado: 6.9
Valores da diagonal secundária:
Linha: 1 Coluna: 3 Valor armazenado: 6.8
Linha: 2 Coluna: 2 Valor armazenado: 6.4

Linha: 3 Coluna: 1 Valor armazenado: 7.2

Matriz Nula

Uma matriz nula é uma matriz em que todos os elementos são nulos. Denotamos uma matriz nula de ordem *mxn* por *Omxn*.

Exemplos:

a) Matriz nula com 3 linhas e 2 colunas: *O = [[0, 0], [0, 0], [0, 0]]*

b) Matriz nula de ordem 3: *O = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]*

Matriz Diagonal

Toda matriz quadrada em que os elementos que não pertencem à diagonal principal sejam iguais a zero (nulos) é uma matriz diagonal. Os elementos da diagonal principal podem ser iguais a zero ou não.

Exemplos:

a) *D = [[3, 0, 0], [0, -1, 0], [0, 0, 7]]*

b) *E = [[1, 0], [0, 3]]*

c) *F = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]*. A matriz *F* caracteriza um exemplo de matriz nula e também de matriz diagonal.

VAMOS PRATICAR

Vamos utilizar a matriz *D* como exemplo e criar um programa em *Python* que verifica se esta é uma matriz diagonal. Crie um novo arquivo no *IDLE Editor*, denominado *matriz3.py*, e digite o seguinte código-fonte:

```
D = [[3, 0, 0], [0, -1, 0], [0, 0, 7]]
matrizdiagonal=0
for linha in range(3):
    for coluna in range(3):
        if linha!=coluna:
            if D[linha][coluna]!=0:
                matrizdiagonal=1
if matrizdiagonal==0:
    print("A matriz D é uma matriz diagonal")
else:
    print("A matriz D não é uma matriz diagonal")
```

INTERATIVIDADE: Utilize o *IDLE Editor* para desenvolver a atividade proposta.

ATENÇÃO: O código-fonte do arquivo *matriz3.py* encontra-se no Apêndice deste *e-book*.

Neste exemplo, utilizamos uma *flag* (ou bandeira), representada pela variável *matrizdiagonal*. Inicializamos esta variável com o valor zero e, caso exista um elemento que não faça parte da diagonal principal da matriz quadrada (elementos cujas linha e coluna não são iguais), cujo valor é diferente de zero, fazemos com que esta *flag* receba o valor um. Ao final dos laços de repetição *for* (após percorrermos toda a matriz *D*), se a *flag* ainda mantiver o valor zero, é porque nenhum elemento fora da diagonal principal tem valor diferente de zero, o que indica que essa matriz é uma matriz diagonal.

Matriz Identidade

Uma matriz identidade é uma matriz quadrada em que os elementos que pertencem à diagonal principal são todos iguais a 1 e o restante dos elementos são todos iguais a zero. É representada por *In*, sendo *n* a ordem da matriz (que é quadrada).

Se preferirmos trabalhar com a lei de formação de uma matriz, podemos escrever a matriz identidade na forma:

In = [aij] em que aij = 1 *se i = j e* 0 *se i != j*

Exemplos:

a) *I3 = [[*1, 0, 0], [0, 1, 0], [0, 0, 1*]]*

b) *I1 = [[*1*]]*

c) *I2 = [[*1, 0], [0, 1*]]*

d) *I4 = [[*1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1*]]*

Vamos praticar
Seguindo o exemplo anterior de código-fonte que desenvolvemos para identificar uma matriz diagonal, escreva um programa em *Python* para identificar se uma matriz é uma matriz identidade.

INTERATIVIDADE: Utilize o *IDLE Editor* para realizar a atividade proposta.

Matriz Transposta

Se *A* for uma matriz de ordem *mxn*, a matriz transposta de *A*, representada por *At*, é obtida a partir da matriz *A* trocando-se ordenadamente as linhas por colunas (ou as colunas por linhas). Assim, a matriz *At* terá ordem *nxm*.

Exemplos:

a) *A = [[2, 5], [0, 9], [-2, 7]]*, logo *At = [[2, 0, -2], [5, 9, 7]]*

b) *A = [[-1], [4], [3]]*, logo *At = [[-1, 4, 3]]*

c) *A = [[8, 2, 3], [0, 1, 4], [6, 11, 9]]*, logo *At = [[8, 0, 6], [2, 1, 11], [3, 4, 9]]*

Vamos praticar

Vamos escrever um programa em *Python* para gerar a matriz transposta da matriz *A = [[2, 5], [0, 9], [-2, 7]]*. Crie um novo arquivo no *IDLE Editor*, denominado *matriz4.py*, e digite o seguinte código-fonte:

```
# a matriz A tem 3 linhas e 2 colunas
A = [[2, 5], [0, 9], [-2, 7]]
# a matriz B, onde guardaremos a matriz transposta de A, terá 2 linhas e 3
colunas
# criamos a matriz B com os valores nulos (zeros)
B = [[0, 0, 0],[0, 0, 0]]
# Impressão dos dados da matriz A
print("Valores da Matriz A:")
for linha in range(3):
    for coluna in range(2):
        print("Linha:",linha+1,"Coluna:",coluna+1,"Valor
armazenado:",A[linha][coluna])
# Transposição da matriz A para B
for linha in range(3):
    for coluna in range(2):
        B[coluna][linha]=A[linha][coluna]
# Impressão dos dados da matriz B
print("Valores da Matriz B:")
for linha in range(2):
    for coluna in range(3):
        print("Linha:",linha+1,"Coluna:",coluna+1,"Valor
armazenado:",B[linha][coluna])
```

Interatividade: Utilize o *IDLE Editor* para realizar a atividade proposta.

Atenção: O código-fonte do arquivo *matriz4.py* encontra-se no Apêndice deste *e-book*.

Os resultados da execução deste código-fonte são os seguintes:

Valores da Matriz A:
Linha: 1 Coluna: 1 Valor armazenado: 2
Linha: 1 Coluna: 2 Valor armazenado: 5
Linha: 2 Coluna: 1 Valor armazenado: 0
Linha: 2 Coluna: 2 Valor armazenado: 9
Linha: 3 Coluna: 1 Valor armazenado: -2

Linha: 3 Coluna: 2 Valor armazenado: 7
Valores da Matriz B:
Linha: 1 Coluna: 1 Valor armazenado: 2
Linha: 1 Coluna: 2 Valor armazenado: 0
Linha: 1 Coluna: 3 Valor armazenado: -2
Linha: 2 Coluna: 1 Valor armazenado: 5
Linha: 2 Coluna: 2 Valor armazenado: 9
Linha: 2 Coluna: 3 Valor armazenado: 7

Propriedades da Matriz Transposta:

- Para qualquer matriz *A = (aij)mxn*, tem-se que: *(At)t = A*
- Dadas as matrizes *A = (aij)mxn* e *B = (bij)mxn*, tem-se que *(A+B)t = At+Bt*
- Dadas as matrizes *A = (aij)mxn* e *B = (bij)nxp*, tem-se que *(AB)t = BtAt*
- Para qualquer matriz *A = (aij)mxn* e para qualquer número real não nulo *k*, tem-se que: *(kA)t = kAt*

Matriz Simétrica

Uma matriz simétrica é uma matriz quadrada tal que *A = At*.

Nos exemplos a seguir, será possível constatar que, em uma matriz simétrica, os elementos simetricamente dispostos em relação à diagonal principal são iguais.

Exemplos:

a) *A = [[5, 3], [3, 2]]* é simétrica, pois *A=At*.

b) *B = [[2, 4, -1], [4, 0, 5], [-1, 5, 1]]* é simétrica, pois *B=Bt*.

c) *C = [[3, -2], [2, 0]]* não é simétrica, uma vez que *C != Ct*, sendo *Ct = [[3, 2], [-2, 0]]*. Observe que a desigualdade ocorreu justamente porque o elemento *c12 != c21*. (O símbolo *!=* representa diferente).

Vamos praticar

Crie um programa em *Python* para verificar se uma matriz é simétrica, a partir do exemplo em que criamos uma matriz transposta.

Propriedades da Matriz Simétrica:

- Se A é uma matriz simétrica e k é um número real não nulo, então kA é também uma matriz simétrica.
- Para qualquer matriz quadrada A, tem-se que $A+At$ é uma matriz simétrica.

Matriz Oposta

A matriz oposta de uma matriz A é a matriz $-A$, que é obtida trocando-se o sinal de todos os elementos de A. Assim, a matriz oposta de $A = (aij)mxn$ é a matriz $-A = (-aij)mxn$.

Exemplos:

a) *A = [[2, 6], [-1, 3], [5, -4]]*, logo a matriz oposta de *A* será *-A = [[-2, -6], [1, -3], [-5, 4]]*

b) *B = [[3, -2], [-1, 7]]*, logo a matriz oposta de *B* será *-B = [[-3, 2], [1, -7]]*

Igualdade de Matrizes

Duas matrizes A e B, de mesma ordem mxn, são ditas iguais se e somente se todos os seus elementos correspondentes (que ocupam a mesma posição) são iguais, ou seja:

$A = B$ se e somente se $aij = bij$ para todo $1 <= i <= m$ e $1 <= j <= n$

Exemplos:

a) Dadas *A = [[3, 9], [1, 2], [0, 5]]* e *B = [[3, 9], [1, 2], [0, 5]]*. Neste caso, temos $A = B$.

b) Dadas *C = [[2, 4], [0, 3]]* e *D = [[2, 4], [1, 3]]*. Neste caso, temos *C != D*, uma vez que *c21 != d21*.

VAMOS PRATICAR

Vamos criar um novo programa em *Python*, denominado *matriz5.py*, e verificar se as matrizes *A = [[3, 9], [1, 2], [0, 5]]* e *B = [[3, 9], [1, 2], [0, 5]]* são iguais. Digite e execute o código-fonte abaixo:

```
A = [[3, 9], [1, 2], [0, 5]]
B = [[3, 9], [1, 2], [0, 5]]
matrizesiguais=0
for linha in range(3):
    for coluna in range(2):
        if A[linha][coluna]!=B[linha][coluna]:
            matrizesiguais=1
if matrizesiguais==0:
    print("As matrizes A e B são iguais")
else:
    print("As matrizes A e B não são iguais")
```

ATENÇÃO: O código-fonte do arquivo *matriz5.py* encontra-se no Apêndice deste *e-book*.

INTERATIVIDADE: Utilize o *IDLE Editor* para realizar a atividade proposta.

Utilizamos, neste exemplo, o recurso da *flag* que denominamos *matrizesiguais*. Inicializamos a *flag* com o valor zero e, caso algum dos valores das matrizes, de acordo com as linhas e colunas correspondentes, não sejam iguais, setamos o valor da *flag* para 1. Após percorrer todas as matrizes *A* e *B*, se o valor da *flag* se mantiver zero, é porque todos os valores são iguais.

ATENÇÃO: Sugerimos que sejam efetuadas as atividades numeradas de 1 a 11 deste capítulo, uma vez que tais atividades são referentes aos conteúdos estudados até aqui. As atividades estão no final desta Unidade.

2.3
OPERAÇÕES COM MATRIZES

Nesta subunidade, vamos estudar as operações de adição, subtração e multiplicação de matrizes. Faremos, também, exemplos práticos utilizando a linguagem de programação *Python*, para demonstrar como aplicar essas operações.

2.3.1 Adição de Matrizes

Para adicionarmos duas ou mais matrizes, é preciso que todas elas tenham o mesmo número de linhas e de colunas, ou seja, as matrizes devem ter a mesma ordem. A soma dessas matrizes resultará em uma outra matriz que também terá o mesmo número de linhas e de colunas. Para efetuarmos a soma de matrizes, os termos deverão ser somados aos seus termos correspondentes.

Assim, dadas duas matrizes, *A* e *B*, as duas de ordem *mxn*, então: *A+B = C*, com *C* de ordem *mxn*, onde *aij+bij = cij*.

Exemplo:

Dadas as matrizes *A3x2 = [[1, 8], [0, 2], [5, -1]]* e *B3x2 = [[8, -2], [2, 4], [3, 5]]*, calcule a matriz *C = A B*.

Resolução:

Sendo *A* e *B* matrizes de mesma ordem, para calcularmos a matriz *C*, resultante da soma de *A* e *B*, temos que somar os termos correspondentes, assim:

C = [[1, 8], [0, 2], [5, -1]] + [[8, -2], [2, 4], [3, 5]]

logo

C = [[1+8, 8-2], [0+2, 2+4], [5+3, -1+5]]

ou seja

C = [[9, 6], [2, 6], [8, 4]]

sendo *C* uma matriz de ordem 3x2.

VAMOS PRATICAR

Vamos criar um novo arquivo, denominado *matriz6.py,* e vamos criar a matriz *C*, realizando a adição das matrizes *A = [[1, 8], [0, 2], [5, -1]]* e *B = [[8, -2], [2, 4], [3, 5]]*. Para tanto, digite o código-fonte abaixo e execute-o:

```
A = [[1, 8], [0, 2], [5, -1]]
B = [[8, -2], [2, 4], [3, 5]]
C = [[0, 0],[0, 0],[0, 0]]

# Impressão dos valores da matriz A
print("Valores armazenados na Matriz A:")
```

```
for linha in range(3):
    for coluna in range(2):
        print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",A[linha][coluna])
 # Impressão dos valores da matriz B
print("Valores armazenados na Matriz B:")
for linha in range(3):
    for coluna in range(2):
        print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",B[linha][coluna])
# Soma dos valores das matrizes A e B e atribuição à matriz C
for linha in range(3):
    for coluna in range(2):
        C[linha][coluna]=A[linha][coluna]+B[linha][coluna]
# Impressão dos valores da matriz C
print("Valores armazenados na Matriz C:")
for linha in range(3):
    for coluna in range(2):
        print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",C[linha][coluna])
```

ATENÇÃO: O código-fonte do arquivo *matriz6.py* encontra-se no Apêndice deste *e-book*.

INTERATIVIDADE: Utilize o *IDLE Editor* para realizar a atividade proposta.

O resultado da execução deste programa é o seguinte:

```
Valores armazenados na Matriz A:
    Linha: 1 Coluna: 1 Valor armazenado: 1
    Linha: 1 Coluna: 2 Valor armazenado: 8
    Linha: 2 Coluna: 1 Valor armazenado: 0
    Linha: 2 Coluna: 2 Valor armazenado: 2
    Linha: 3 Coluna: 1 Valor armazenado: 5
    Linha: 3 Coluna: 2 Valor armazenado: -1

Valores armazenados na Matriz B:
    Linha: 1 Coluna: 1 Valor armazenado: 8
    Linha: 1 Coluna: 2 Valor armazenado: -2
    Linha: 2 Coluna: 1 Valor armazenado: 2
    Linha: 2 Coluna: 2 Valor armazenado: 4
    Linha: 3 Coluna: 1 Valor armazenado: 3
    Linha: 3 Coluna: 2 Valor armazenado: 5
```

Valores armazenados na Matriz C:
Linha: 1 Coluna: 1 Valor armazenado: 9
Linha: 1 Coluna: 2 Valor armazenado: 6
Linha: 2 Coluna: 1 Valor armazenado: 2
Linha: 2 Coluna: 2 Valor armazenado: 6
Linha: 3 Coluna: 1 Valor armazenado: 8
Linha: 3 Coluna: 2 Valor armazenado: 4

Propriedades da Soma entre Matrizes

Sejam A, B e C matrizes dadas e O a matriz nula, temos:

Propriedade comutativa: $A+B = B+A$.
Propriedade distributiva: $(A+B)+C = A+(B+C)$.
Propriedade do elemento neutro da soma: $A+O = O+A = A$.
Propriedade do elemento oposto da soma: $A+(-A) = (-A)+A = O$.

2.3.2 Subtração de Matrizes

Para podermos efetuar a subtração de matrizes, as matrizes subtraídas devem ter a mesma ordem (mesmo número de linhas e colunas), sendo que a matriz obtida com a subtração (matriz diferença) terá o mesmo número de linhas e colunas das matrizes subtraídas. As subtrações devem ocorrer com elementos correspondentes.

Assim, dadas duas matrizes, A e B, as duas de ordem mxn, ao subtrairmos a matriz A da matriz B, teremos $A\text{-}B = C$, com C de ordem mxn, onde $aij\text{-}bij = cij$.

Exemplo:
Dadas as matrizes $A3x2 = [[1, 8], [0, 2], [5, -1]]$ e $B3x2 = [[8, -2], [2, 4], [3, 5]]$, calcule a matriz $D = A\text{-}B$.

Resolução:
Sendo A e B matrizes de mesma ordem, para calcularmos a matriz D resultante da diferença entre A e B, temos que subtrair termos correspondentes, assim:
$D = [[1, 8], [0, 2], [5, -1]] - [[8, -2], [2, 4], [3, 5]]$
é equivalente a
$D = [[1\text{-}8, 8\text{+}2], [0\text{-}2, 2\text{-}4], [5\text{-}3, -1\text{-}5]]$
ou seja
$D = [[-7, 10], [-2, -2], [2, -6]]$
sendo D uma matriz de ordem 3x2.

VAMOS PRATICAR

A partir do programa *matriz6.py*, em que calculamos a adição de matrizes, escreva um programa em *Python* para calcular a subtração das matrizes *A* e *B*, sendo *A= [[1, 8], [0, 2], [5, -1]]* e *B = [[8, -2], [2, 4], [3, 5]]*, e calcule a matriz *D = A-B*.

ATENÇÃO: O código-fonte do arquivo *matriz6.py* encontra-se no Apêndice deste *e-book*.

INTERATIVIDADE: Utilize o *IDLE Editor* para realizar a atividade proposta.

2.3.3 Multiplicação de Matrizes

Dadas as matrizes *A = (aij)mxn* e *B = (bij)nxp*, o produto de *A* e *B* é uma matriz *C = (cij)mxp*, em que cada elemento *cij* é dado por
cij = Soma(n, k=1)(aik bkj), com *i = 1, 2, 3, …, m* e *j = 1, 2, 3, …, p*.

Assim, temos que cada elemento da matriz *C* é obtido por meio da soma dos produtos dos elementos correspondentes da *i*-ésima linha de *A* pelos elementos da *j*-ésima coluna de *B*.

Observe que esta multiplicação só será possível se o número de colunas de *A* (denotado aqui por *n*) for exatamente igual ao número de linhas de *B* (por esta razão aqui também denotado por *n*). A matriz *C*, resultante dessa multiplicação, terá ordem *mxp*, que corresponde ao número de linhas de *A* e ao número de colunas de *B*.

Exemplo:

Dadas as matrizes *A2x2 = [[4, 0], [2, 7]]* e *B2x3 = [[1, 0, 2], [7, 5, 3]]*, calcular a matriz *C = AB*.

Resolução:

Vamos efetuar passo a passo a multiplicação das matrizes *A2x2 = [[4, 0], [2, 7]]* e *B2x3 = [[1, 0, 2], [7, 5, 3]]*, para exemplificar a obtenção de cada elemento da matriz *C2x3 = A2x2B2x3*.

Neste exemplo, temos:

C = AB

então

C = [[4, 0], [2, 7]] [[1, 0, 2], [7, 5, 3]] Teremos que determinar os seis elementos da matriz *C*, ou seja:

C = [[c11, c12, c13], [c21, c22, c23]].

1º) Trabalhando com os elementos da 1ª linha de *A* e da 1ª coluna de *B*, obtemos o elemento *c11*:

c11 = (4)(1)+(0)(7) = 4

2º) Trabalhando com os elementos da 1ª linha de *A* e da 2ª coluna de *B*, obtemos o elemento *c12*:

c12 = (4)(0) + (0)(5) = 0

3º) Trabalhando com os elementos da 1ª linha de *A* e *B*, obtemos o elemento *c13*:

c13 = (4)(2) + (0)(3) = 8

4º) Trabalhando com os elementos da 2ª linha de *A* e da 1ª coluna de *B*, obtemos o elemento *c21*:

c21 = (2)(1) + (7)(7) = 51

5º) Trabalhando com os elementos da 2ª linha de *A* e da 2ª coluna de *B*, obtemos o elemento *c22*:

c22 = (2)(0) + (7)(5) = 35

6º) Trabalhando com os elementos da 2ª linha de *A* e da 3ª coluna de *B*, obtemos o elemento *c23*:

c23 = (2)(2) + (7)(3) = 25

Terminadas as multiplicações das linhas de *A* pelas colunas de *B*, concluímos o cálculo da matriz *C*:

C2x3= [[4, 0, 8], [51, 35, 25]]

ATENÇÃO: Em geral, o resultado da multiplicação *AB* é diferente do resultado da multiplicação *BA*.

VAMOS PRATICAR:

Vamos criar um novo programa, denominado *matriz7.py* e realizarmos a multiplicação das matrizes *A=[[4, 0],[2, 7]]* pela matriz *B=[[1, 0, 2],[7, 5, 3]]*, armazenando o resultado na matriz *C*. Digite e execute o código-fonte abaixo:

```
A = [[4, 0], [2, 7]]
B = [[1, 0, 2], [7, 5, 3]]
```

```
C = [[0, 0, 0],[0, 0, 0]]
# Impressão da matriz A
print("Valores da Matriz A:")
for linha in range(2):
    for coluna in range(2):
        print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",A[linha]
[coluna])
# Impressão da matriz B
print("Valores da Matriz B:")
for linha in range(2):
    for coluna in range(3):
        print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",B[linha]
[coluna])
# for linha de acordo com o número de linhas da matriz A (duas linhas)
# for coluna de acordo com o número de colunas da matriz B (três colunas)
# for k de acordo com o número de colunas da matriz A e de linhas da matriz B
for linha in range(2):
    for coluna in range(3):
        for k in range(2):
            C[linha][coluna]=C[linha][coluna]+A[linha][k]*B[k][coluna]
 # Impressão da matriz C
print("Valores da matriz C - multiplicação da matriz A pela matriz B:")
for linha in range(2):
    for coluna in range(3):
        print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",C[linha]
[coluna])
```

INTERATIVIDADE: Utilize o *IDLE Editor* para realizar a atividade proposta.

ATENÇÃO: O código-fonte do arquivo *matriz7.py* encontra-se no Apêndice deste *e-book*.

O resultado da execução deste código-fonte é apresentado abaixo:

```
Valores da Matriz A:
    Linha: 1 Coluna: 1 Valor armazenado: 4
    Linha: 1 Coluna: 2 Valor armazenado: 0
    Linha: 2 Coluna: 1 Valor armazenado: 2
    Linha: 2 Coluna: 2 Valor armazenado: 7
```

Valores da Matriz B:

Linha: 1 Coluna: 1 Valor armazenado: 1
Linha: 1 Coluna: 2 Valor armazenado: 0
Linha: 1 Coluna: 3 Valor armazenado: 2
Linha: 2 Coluna: 1 Valor armazenado: 7
Linha: 2 Coluna: 2 Valor armazenado: 5
Linha: 2 Coluna: 3 Valor armazenado: 3

Valores da matriz C - multiplicação da matriz A pela matriz B:

Linha: 1 Coluna: 1 Valor armazenado: 4
Linha: 1 Coluna: 2 Valor armazenado: 0
Linha: 1 Coluna: 3 Valor armazenado: 8
Linha: 2 Coluna: 1 Valor armazenado: 51
Linha: 2 Coluna: 2 Valor armazenado: 35
Linha: 2 Coluna: 3 Valor armazenado: 25

Propriedades da Multiplicação de uma Matriz por Escalar:

Para *A* e *B* matrizes de mesma ordem, *k* e *l* escalares e O a matriz nula, temos:

- Propriedade associativa da multiplicação: $(kl)A = k(lA)$
- Propriedades distributivas: $k(A+B) = kA+kB$ e $(k+l)A = kA+lA$
- Propriedades de identidade multiplicativa: $1A = A$
- Propriedades multiplicativas do zero: $0A = \mathrm{O}$ (multiplicar a matriz *A* por zero resulta na matriz nula) e $k\mathrm{O} = \mathrm{O}$ (multiplicar a matriz nula por um escalar *k* resulta na matriz nula).

Propriedades da Multiplicação de uma Matriz por outra Matriz:

Sejam *A*, *B* e *C* matrizes dadas, *I* a matriz identidade e O a matriz nula. Sempre que for possível efetuar os produtos indicados, de acordo com a definição do produto de matrizes, teremos:

- Propriedade associativa: $(AB)C = A(BC)$
- Propriedades distributivas: $A(B+C) = AB+AC$ e $(B+C)A = BA+CA$.
- Propriedade do elemento neutro da multiplicação: $IA = A$ e $AI = A$.
- Propriedade do elemento nulo da multiplicação: $\mathrm{O}A = \mathrm{O}$ e $A\mathrm{O} = \mathrm{O}$.
- Multiplicação de matrizes não é comutativa: *AB != BA.*

Exemplo 1:
Dadas as matrizes *A = [[1, 3], [2, 4], [3, 0]]* e *B = [[0, 1, 2], [-1, 2, 0]]*, se *At* é a matriz transposta de *A*, calcule *At-3B*

Resolução*:*
Sendo *A = [[1, 3], [2, 4], [3, 0]]*, então *At = [[1, 2, 3], [3, 4, 0]]*, e como *B = [[0, 1, 2], [-1, 2, 0]]*, então
3B = [[0, 3, 6], [-3, 6, 0]].
Desta forma,
At-3B = [[1, 2, 3], [3, 4, 0]] – [[0, 3, 6], [-3, 6, 0]]
e, portanto,
At-3B = [[1, -1, -3], [6, -2, 0]].

Exemplo 2:
(ENEM 2012): Um aluno registrou as notas bimestrais de algumas de suas disciplinas em uma tabela. Ele observou que as entradas numéricas da tabela formavam uma matriz 4*x*4, e que poderia calcular as médias anuais dessas disciplinas usando produto de matrizes. Todas as provas possuíam o mesmo peso, e a tabela que ele conseguiu é dada a seguir (Tabela 2).

Tabela 2 – Notas por Disciplina e Bimestres

	1º Bimestre	2º Bimestre	3º Bimestre	4º Bimestre
Matemática	5,9	6,2	4,5	5,5
Português	6,6	7,1	6,5	8,4
Geografia	8,6	6,8	7,8	9,0
História	6,2	5,6	5,9	7,7

Fonte: Brasil (2012).

Descrição de Imagem: Tabela 2- Contém cinco linhas e cinco colunas. Em suas colunas, apresenta informações relacionadas a quatro bimestres ("1º Bimestre", "2º Bimestre", "3º Bimestre" e "4º Bimestre") e, em suas linhas, estão elencadas as disciplinas "Matemática", "Português", "Geografia" e "História". Como valores dispostos na tabela, temos:

Matemática: 1º Bimestre: nota 5,9; 2º Bimestre: 6,2; 3º Bimestre: 4,5; 4º Bimestre: 5,5.

Português: 1º Bimestre: 6,6; 2º Bimestre: 7,1; 3º Bimestre: 6,5; 4º Bimestre: 8,4.

Geografia: 1º Bimestre: 8,6; 2º Bimestre: 6,8; 3º Bimestre: 7,8; 4º Bimestre: 9,0.
História: 1º Bimestre: 6,2; 2º Bimestre: 5,6; 3º Bimestre: 5,9; 4º Bimestre: 7,7.

Para obter essas médias, ele deve multiplicar a matriz obtida a partir da Tabela 2 por:

a) [[½, ½, ½, ½]]
b) [[¼, ¼, ¼, ¼]]
c) [[1], [1], [1], [1]]
d) [[½], [½], [½], [½]]
e) [[¼], [¼], [¼], [¼]]

Resolução:

Para resolver este problema, precisamos fazer um cálculo de multiplicação de matrizes, uma vez que a nota anual de cada disciplina deve ser obtida pela soma das 4 notas bimestrais da disciplina, com consequente divisão desta soma por 4, ou então multiplicar cada uma das notas por ¼ e depois somá-las (média aritmética de quatro valores). Escrevendo a matriz em linguagem *Python* (lembrando de usar o ponto como separador decimal), temos:

M = [[5.9, 6.2, 4.5, 5.5], [6.6, 7.1, 6.5, 8.4], [8.6, 6.8, 7.8, 9.0], [6.2, 5.6, 5.9, 7.7]]

Assim, chegamos a uma matriz de ordem *4x4,* e para proporcionar uma soma dos elementos de cada linha divididos por 4 (média aritmética anual de cada disciplina), temos que multiplicar a matriz *M* por uma matriz de ordem *4x1*. Sendo assim, a matriz procurada está na alternativa "e".

Multiplicando a matriz *M* pela matriz dada na alternativa "e", obtemos:

[[5.9, 6.2, 4.5, 5.5], [6.6, 7.1, 6.5, 8.4], [8.6, 6.8, 7.8, 9.0], [6.2, 5.6, 5.9, 7.7]] [[¼], [¼], [¼], [¼]] = [[(5.9 + 6.2 + 4.5 + 5.5)/4], [(6.6 + 7.1 + 6.5 + 8.4)/4], [(8.6 + 6.8 + 7.8 + 9.0)/4], [(6.2 + 5.6 + 5.9 + 7.7)/4]] = [[5.5], [7.1], [8.0], [6.3]]

Sendo assim, as médias anuais foram: Matemática = 5,5; Português = 7,1; Geografia = 8,0 e História = 6,3 (na resposta, usamos a vírgula como separador decimal).

Exemplo 3:

Em um campeonato de futebol, obteve-se o seguinte resultado referente aos quatro times que disputaram a competição, como mostra a Tabela 3:

Tabela 3 – Resultados da Competição

	Vitória	Empate	Derrota
Time A	2	0	1
Time B	0	1	2
Time C	1	1	1
Time D	1	2	0

Fonte: Fortes et al. (2019).

Descrição de Imagem: Tabela 3- A Tabela 3 possui cinco linhas e quatro colunas. Nas colunas, estão as denominações: "vitória", "empate" e "derrota", enquanto que suas linhas são correspondentes aos times "A", "B", "C" e "D". Como valores dispostos na tabela, temos:

Time A: Vitória 2, Empate 0, Derrota 1

Time B: Vitória 0, Empate 1, Derrota 2

Time C: Vitória 1, Empate 1, Derrota 1

Time D: Vitória 1, Empate 2, Derrota 0.

Pelo regulamento do campeonato, vale a seguinte tabela de pontuação, conforme Tabela 4:

TABELA 4 - Pontuação

Vitória	3 pontos
Empate	1 ponto
Derrota	0 pontos

FONTE: Fortes et al. (2019).

Descrição de Imagem: Tabela 4- A Tabela 4 apresenta três linhas e duas colunas, sendo que a primeira coluna apresenta os resultados possíveis ("vitória", "empate" e "derrota") e a segunda coluna apresenta os pontos respectivos. Como valores dispostos na tabela, temos:

Vitória: 3 pontos.
Empate: 1 ponto.
Derrota: 0 pontos.

Qual foi a classificação de cada um dos quatro times no final do campeonato?

Resolução:

Para solucionar este problema, devemos realizar uma operação de multiplicação de matrizes:

[[2, 0, 1], [0, 1, 2], [1, 1, 1], [1, 2, 0]] [[3], [1], [0]] = [[6], [1], [4], [5]]

Assim, a classificação final foi: em 1º lugar o Time A, com 6 pontos; em 2º lugar o Time D, com 5 pontos; em 3º lugar o Time C, com 4 pontos; e em 4º lugar o Time B, com apenas 1 ponto.

ATENÇÃO: Sugerimos que sejam efetuadas as atividades numeradas de 12 a 23 desta Unidade, uma vez que tais atividades são referentes às operações com matrizes.

2.4
MATRIZ INVERSA

Dada uma matriz quadrada *A*, se existe outra matriz *B* que verifique *(A)(B)=(B)(A) = I* (onde *I* é a matriz identidade), então dizemos que *B* é a matriz inversa de *A*. Vamos aqui adotar a representação da matriz inversa de *A* por *Ai*. Sendo assim, se *Ai* é a inversa da matriz quadrada *A*, teremos *(A)(Ai) = (Ai)(A) = I*.

Se a matriz *A* é invertível, sua inversa é única. Se *A* não admite inversa, diz-se que *A* é singular.

Exemplo:

Dadas as matrizes *A = [[1, 3], [2, 4]]* e *B = [[-2, 3/2], [1, -1/2]]*, verifique se uma é inversa da outra.

Resolução:

Para realizar a verificação, temos que multiplicar as matrizes e verificar se o resultado consiste em uma matriz identidade.

(A)(B) = [[1, 3], [2, 4]] [[-2, 3/2], [1, -1/2]] = [[1(-2)+3(1), 1(3/2)+3(-1/2)], [2(-2)+4(1), 2(3/2)+4(-1/2)]] = [[1, 0], [0, 1]] = I2.

Portanto, neste caso, a matriz *B* é inversa da matriz *A*. Podemos então denotar *B = Ai*.

Ainda, por definição, podemos verificar que a matriz *A* é inversa da matriz *B*, uma vez que

(B)(A) = [[-2, 3/2], [1, -1/2]] [[1, 3], [2, 4]] = [[-2(1)+3/2(2), -2(3)+3/2(4)], [1(1)+(-1/2)(2), 1(3)+(-1/2)(4)]] = [[1, 0], [0, 1]] = I2.

VAMOS PRATICAR

Vamos criar um novo programa, denominado *matriz8.py*, para verificarmos se as matrizes *A = [[1, 3], [2, 4]]* e *B = [[-2, 3/2], [1, -1/2]]* são inversas uma da outra. Vamos calcular a multiplicação das matrizes (seguindo o exemplo que fizemos anteriormente no código-fonte *matriz8.py*) e, posteriormente, verificar se o resultado da multiplicação é uma matriz identidade, sendo que uma matriz identidade é uma matriz quadrada, em que os elementos que pertencem à diagonal principal são todos iguais a 1 e o restante dos elementos são todos iguais a zero.

```
A = [[1, 3], [2, 4]]
B = [[-2, 3/2], [1, -1/2]]
C = [[0, 0],[0, 0]]
 # Impressão da matriz A
print("Valores da Matriz A:")
```

```
for linha in range(2):
      for coluna in range(2):
           print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",A[linha]
[coluna])
 # Impressão da matriz B
print("Valores da Matriz B:")
for linha in range(2):
      for coluna in range(2):
           print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",B[linha]
[coluna])
 # for linha de acordo com o número de linhas da matriz A (duas linhas)
# for coluna de acordo com o número de colunas da matriz B (três colunas)
# for k de acordo com o número de colunas da matriz A e de linhas da matriz B
for linha in range(2):
    for coluna in range(2):
        for k in range(2):
            C[linha][coluna]=C[linha][coluna]+A[linha][k]*B[k][coluna]
 # Impressão da matriz C
#Verificação da matriz C - se é matriz identidade
matrizidentidade=0
print("Valores da matriz C - multiplicação da matriz A pela matriz B:")
for linha in range(2):
     for coluna in range(2):
          print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",C[linha]
[coluna])
          if linha==coluna:
             if C[linha][coluna]!=1:
matrizidentidade=1
else:
if C[linha][coluna]!=0:
matrizidentidade=1
if matrizidentidade==0:
      print("A matriz C é uma matriz identidade, então as matrizes A e B são
inversas uma da outra")
else:
print("A matriz C não é uma matriz identidade, então as matrizes A e B não são
inversas uma da outra")
```

Atenção: O código-fonte do arquivo *matriz8.py* encontra-se no Apêndice deste *e-book*.

INTERATIVIDADE: Utilize o *IDLE Editor* para realizar a atividade proposta.

Neste exemplo, utilizamos a *flag matrizidentidade* para verificar se a matriz resultante da multiplicação (a matriz *C*) é identidade, sendo que os valores da diagonal principal precisam ser iguais a 1 e os demais valores da matriz precisam ser iguais a zero.

O resultado da execução deste programa pode ser visto a seguir:

Valores da Matriz A:

Linha: 1 Coluna: 1 Valor armazenado: 1
Linha: 1 Coluna: 2 Valor armazenado: 3
Linha: 2 Coluna: 1 Valor armazenado: 2
Linha: 2 Coluna: 2 Valor armazenado: 4

Valores da Matriz B:

Linha: 1 Coluna: 1 Valor armazenado: -2
Linha: 1 Coluna: 2 Valor armazenado: 1.5
Linha: 2 Coluna: 1 Valor armazenado: 1
Linha: 2 Coluna: 2 Valor armazenado: -0.5

Valores da matriz C - multiplicação da matriz A pela matriz B:

Linha: 1 Coluna: 1 Valor armazenado: 1
Linha: 1 Coluna: 2 Valor armazenado: 0.0
Linha: 2 Coluna: 1 Valor armazenado: 0
Linha: 2 Coluna: 2 Valor armazenado: 1.0
A matriz C é uma matriz identidade, então as matrizes A e B são inversas uma da outra

2.4.1 Cálculo de uma Matriz Inversa

De acordo com a definição, para determinar a matriz inversa de uma matriz quadrada *A* de ordem *n*, basta descobrir uma matriz *B* tal que a multiplicação entre elas tenha como resultado uma matriz identidade de ordem *n*.

Uma maneira de determinar a inversa de uma matriz é por meio da multiplicação da matriz dada por uma matriz genérica, promovendo ainda a igualdade desta multiplicação a uma matriz identidade.

Exemplo:
Calcule a inversa da matriz *A=[[3, 2], [5, 4]]*.

Resolução:

Procuramos *Ai=[[a, b], [c, d]]* tal que *(A)(Ai) =I2*. Substituindo as expressões de cada uma das três matrizes, obtemos

[[3, 2], [5, 4]] [[a, b], [c, d]] = [[1, 0], [0, 1]].

Realizando a multiplicação das matrizes do lado esquerdo, ficamos com
[[3a+2c, 3b+2d], [5a+4c, 5b+4d]] = [[1, 0], [0, 1]].
Por igualdade de matrizes, chegamos a dois sistemas lineares:
3a+2c=1; 5a+4c=0
e
3b+2d=0; 5b+4d=1

cujas soluções resultam em: *a=2, b=-1, c=-5/2* e *d=3/2*. Portanto, concluímos que:
Ai =[[2, -1], [-5/2, 3/2]].

2.4.2 Dispositivo Prático para Cálculo de Matriz Inversa de Ordem *2x2*

Seja *A* uma matriz de ordem *2 x 2*, ou seja, *A=[[a11, a12], [a21, a22]]*. Se *a11a22 - a21a12 != 0*, então a matriz *A* é invertível, e nesse caso:

Ai =(1/(a11a22 - a21a12))[[a22, -a12], [-a21, a11]]
Se *a11a22 - a21a12 = 0,* então a matriz *A* não é invertível.

Exemplo:
Calcule a matriz inversa da matriz *A=[[2, 3], [2, 5]]*.

Resolução:
Temos *a11a22 - a21a12 = (2)(5) - (2)(3) = 10 - 6 = 4 != 0*. Portanto, a matriz *A* é invertível. Pelo dispositivo prático

Ai =(1/(a11a22 - a21a12))[[a22, -a12], [-a21, a11]]

sendo assim

Ai =(1/((2)(5) – (2)(3)))[[5, -3], [-2, 2]]

ou seja

Ai =(1/4)[[5, -3], [-2, 2]]

e finalmente obtemos

Ai =[[5/4, -3/4], [-1/2, 1/2]]

Para ter certeza de que a matriz obtida corresponde à matriz inversa de *A*, temos que verificar se o produto das duas matrizes resulta na matriz identidade. De fato:

(A)(Ai) = [[2, 3], [2, 5]] [[5/4, -3/4], [-1/2, 1/2]] = [[2(5/4)+3(-1/2), 2(-3/4)+3(1/2)], [2(5/4)+5(-1/2), 2(-3/4)+5(1/2)]] = [[1, 0], [0, 1]] = I2

ATENÇÃO: Sugerimos que sejam efetuadas as atividades numeradas de 24 a 29 desta Unidade, uma vez que tais atividades são voltadas ao cálculo de matrizes inversas.

Atividades – Unidade 2

1) Formular a matriz quadrada *A*, de ordem 2, cujos elementos que compõem a referida matriz são definidos pela lei de formação *aij = 3i-j*.

2) Seja a matriz *A* de ordem *3x4* tal que *aij = i+j*, se *i = j* e *aij = i**2*, se *i != j*, calcule a soma dos elementos *a13*, *a22* e *a34*.

3) Determine a soma dos elementos da diagonal principal com os elementos da diagonal secundária da matriz quadrada *A*, de ordem *3*, em que *aij = 3+3i-j*. Escreva um programa em *Python* para realizar essa soma.

4) Determine a soma dos elementos da matriz linha de ordem *1x5*, que obedece a lei de formação dada por *aij = 2(i**3)-4j*.

5) Se uma matriz quadrada *A* é tal que *At = -A*, ela é chamada matriz antissimétrica. Sabe-se que *M* é uma matriz antissimétrica e:
M = [[4+x, a12, a13], [x, y+2, a23], [y, z, 2z-8]].
Assim, os termos *a12, a13* e *a23* da matriz *M* devem valer respectivamente:

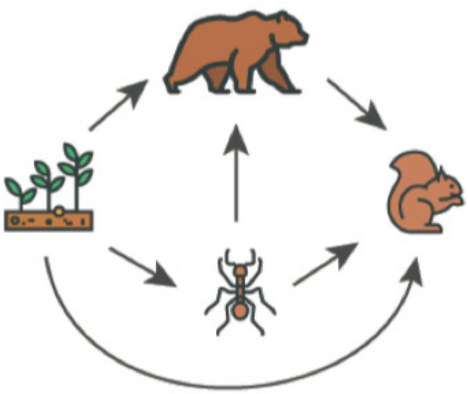
a) *-4*, *-2* e *4*
b) *4*, *2* e *-4*
c) *4*, *-2* e *-4*
d) *2*, *-4* e *2*
e) *2*, *2* e *4*

6) Escreva a matriz diagonal:

a) de ordem *3*, em que *aij = 2i+j* para *i = j*
b) de ordem *4*, em que *aij = i+j* para *i = j*

7) (UFSM, 2011)

Figura 14 - Cadeia Alimentar Simplificada



Fonte: CTE/UFSM.

Descrição de Imagem: Figura 14- Grafo com quatro nodos: Urso, Esquilo, Inseto e Planta. Os nodos estão conectados entre si por meio de setas que representam as arestas no grafo. Obs.: a Planta serve de comida para o Urso, para o Inseto e para o Esquilo; o Inseto serve de comida para o Urso e para o Esquilo; e o Esquilo serve de comida para o Urso.

A Figura 14 representa a cadeia alimentar (simplificada) de um determinado ecossistema. As setas indicam a espécie de que a outra espécie se alimenta. Atribuindo valor 1, quando uma espécie se alimenta de outra, e zero, quando ocorre o contrário, tem-se a seguinte tabela (Tabela 5):

Tabela 5 – Cadeia Alimentar

	Urso	Esquilo	Inseto	Planta
Urso	0	1	1	1
Esquilo	0	0	1	1
Inseto	0	0	0	1
Planta	0	0	0	0

Fonte: UFSM (2011).

Descrição de Imagem: Tabela 5- A tabela 5 apresenta cinco linhas e cinco colunas. Em suas linhas e colunas, os nomes dos animais e planta que estão representados na Figura 14. Nas linhas, temos: "Urso", "Esquilo", "Inseto" e "Planta". Nas colunas, também temos: "Urso", "Esquilo", "Inseto" e "Planta". Como valores dispostos na tabela, temos:

Na linha Urso, temos: 0 (Urso), 1 (Esquilo), 1 (Inseto) e 1 (Planta).

Na linha Esquilo, temos: 0 (Urso), 0 (Esquilo), 1 (Inseto), 1 (Planta).

Na linha Inseto, temos: 0 (Urso), 0 (Esquilo), 0 (Inseto) e 1 (Planta).

Na linha Planta, temos todos os valores preenchidos com 0.

Obs.: O valor dentro de cada item da tabela (ou matriz) corresponde a 1 se o referido animal se alimenta do animal (ou de planta), de acordo com a intersecção entre as linhas e colunas.

A matriz *A=(aij)4x4*, associada à Tabela 5, possui a seguinte lei de formação:

a) *aij = 0* se *i >= j* e *1* se *i < j*

b) *aij = 0* se *i = j* e *1* se *i != j*

c) *aij = 0* se *i <= j* e *1* se *i > j*

d) *aij = 0* se *i != j* e *1* se *i = j*

8) Determinar *a* e *b* para que as matrizes *[[6a+4b, 4], [4, 6a-6b]]* e *[[14, 4], [4, -6]]* sejam iguais.

9) Seja *M* uma matriz quadrada, de ordem *2,* cujos elementos são tais que *mij=2i+2j.* Determine *a, b, c* e *d* para que se tenha *[[a+2b, 3c+d], [3a-2b, 2c+4d]] = M.*

10) Uma empresa é formada por 4 lojas de jogos educacionais, numeradas de 1 a 4. A matriz a seguir apresenta o faturamento (em reais) de cada loja nos 3 primeiros dias de março:

[[1850, 2020, 1700], [1400, 1650, 1620], [2010, 2100, 2300], [2100, 2120, 2000]]

Cada elemento *aij* dessa matriz é o faturamento da loja *i* no dia *j.*

a) Qual foi o faturamento da loja 2 no dia 3?
b) Qual foi o faturamento de todas as lojas no dia 2?
c) Qual a maior fatura? Em que dia? E qual a loja?
d) Qual a menor fatura? Em que dia? E qual a loja?

11) Três meninas foram identificadas por meio de números, sendo 1 = Adriana, 2 = Bianca e 3 = Carla. Elas falam muito ao telefone entre si. A matriz *M* apresenta cada elemento representando o número de telefonemas que a menina de número *i* fez para a menina de número *j,* em um determinado mês.

M = [[0, 13, 10], [18, 0, 6], [9, 12, 0]]

Quem mais telefonou e quem mais recebeu ligações?

12) Dadas as matrizes *A = [[2, -1], [0, 1]]*, *B = [[5, 0, 1], [2, -1, -3]]* e *C = [[4, -8], [1, 0]]*, determine, se possível:

a) *2A +C*
b) *AB*
c) *B+C*
d) *BC*
e) *(A-3C)B*
f) *A**2*

13) Dadas as matrizes: *A = [[0, 1, 1], [1, 1, 0], [1, 1, 1]]* e *B = [[0, -2, 1], [2, -1, 0], [0, 1, 0]],* calcule a matriz *2AB+B**2.*

14) Em uma padaria, são preparados três diferentes tipos de salgados, sendo que na preparação destes produtos são usadas porções de quatro ingredientes, conforme indicado na Tabela 6:

Tabela 6 - Salgados e Ingredientes

	Ovos	Farinha	Açúcar	Carne
Pastel	3	6	1	3
Empada	4	4	2	2
Quibe	1	1	1	6

Fonte: Fortes et al. (2019).

Descrição de Imagem: Tabela 6- Apresenta quatro linhas e cinco colunas. Nas colunas, estão representados os ingredientes, sendo: "ovos", "farinha", "açúcar" e "carne", respectivamente; nas linhas, estão elencados os produtos, sendo: "pastel", "empada" e "quibe". A intersecção entre as linhas e colunas representa a quantidade de cada ingrediente para cada produto. Como valores dispostos na tabela, temos:

Pastel: 3 (ovos), 6 (farinha), 1 (açúcar) e 3 (carne).
Empada: 4 (ovos), 4 (farinha), 2 (açúcar) e 2 (carne).
Quibe: 1 (ovo), 1 (farinha), 1 (açúcar) e 6 (carne).

Os Ingredientes e os preços dos ingredientes constam na Tabela 7.

Tabela 7 – Ingredientes e Preços dos Ingredientes

Ingredientes	Preço da Porção (R$)
Ovo	0,20
Farinha	0,30
Açúcar	0,50
Carne	0,80

Fonte: Fortes et al. (2019).

Descrição de Imagem: Tabela 7- A Tabela 7 apresenta cinco linhas e duas colunas. Na primeira coluna, constam os ingredientes ("ovo"," farinha", "açúcar" e "carne") e, na segunda, o preço de cada porção (em reais). Na linha 1, temos o cabeçalho, sendo que na coluna 1 temos "Ingredientes" e na coluna 2, "Preço da Porção (R$)". Como valores dispostos na tabela, temos:
Ovo (R$ 0,20) Farinha (R$ 0,30) Açúcar (R$ 0,50) e Carne (R$ 0,80)

Qual, então, deve ser o preço base de venda de cada salgado?

15) Uma matriz quadrada *A* é dita idempotente se *A**2 = A*. Verifique se a matriz *A=[[-2, 2], [-3, 3]]* é idempotente.

16) Considere as matrizes *A = [[-1, 1, 3], [8, 3, -5], [1, -2, 7]]* e *At = [[-1, 8, 1], [1, 3, -2], [3, -5, 7]]*. Qual é o tipo correspondente à matriz *A+At*?

17) Uma empresa fabrica *notebooks* de modelos A, B e C, conforme consta na Tabela 8, e os custos e lucros de cada *notebook* estão representados na Tabela 9.

Tabela 8 – Modelos de *Notebooks*

	Modelo A	Modelo B	Modelo C
Janeiro	2	1	4
Fevereiro	0	0	1
Março	4	5	2

Fonte: Fortes et al. (2019).

Descrição de Imagem: Tabela 8- A Tabela 8 apresenta quatro linhas e quatro colunas, sendo que as colunas apresentam modelos ("A", "B" e "C") e as linhas apresentam meses ("Janeiro", "Fevereiro" e "Março"). A intersecção formada entre as linhas e as colunas representa a quantidade de cada modelo fabricado de acordo com o mês. Como valores dispostos na tabela, temos:

Janeiro: Modelo A: 2; Modelo B: 1; Modelo C: 4
Fevereiro: Modelo A: 0; Modelo B: 0; Modelo C: 1
Março: Modelo A: 4; Modelo B: 5; Modelo C: 2

Tabela 9 – Custo e Lucro

Modelo de Notebook	Custo (R$)	Lucro (R$)
A	1.000	0
B	2.000	1.000
C	3.000	2.000

Fonte: Fortes et al. (2019).

Descrição de Imagem: Tabela 9- A Tabela 9 apresenta quatro linhas e três colunas, sendo que as linhas apresentam os modelos de *notebooks* produzidos ("A", "B" e "C") e as colunas, respectivamente, o "Custo" e o "Lucro" na produção de cada tipo de equipamento. A intersecção formada pelas linhas e colunas traz o valor do custo (em reais) e do lucro (em reais). Como valores dispostos na tabela, temos:

Modelo notebook A: Custo: 1.000; Lucro: 0
Modelo notebook B: Custo: 2.000; Lucro: 1.000
Modelo notebook C: Custo: 3.000; Lucro: 2.000

Com base nessas tabelas, qual foi o custo e o respectivo lucro para cada mês?

18) Uma empresa de motos é composta pelas lojas A e B. Ao realizar uma pesquisa de aceitação de dois novos modelos de motos, nos quatro primeiros dias do mês de dezembro, foram obtidos os seguintes resultados:

A = [[1, 3, 2, 4], [3, 2, 0, 5]]
B = [[2, 1, 2, 1], [0, 3, 2, 3]]

Sendo que a matriz A descreve o desempenho da loja A, de modo que cada elemento aij é o número de unidades vendidas do modelo i no dia j; a matriz B descreve o desempenho da loja B, de modo que cada elemento bij é o número de unidades vendidas do modelo i no dia j;

1ª) Qual loja vendeu um número maior de motos de um determinado modelo? Qual o modelo? Quantas?

2ª) Como poderíamos representar, matricialmente, a quantidade vendida desses dois modelos, nas duas lojas, nos quatro primeiros dias de dezembro?

3ª) Como poderíamos representar, matricialmente, a diferença de desempenho da loja A em relação à loja B?

19) Ao comprar os produtos necessários para fazer uma feijoada, uma pessoa resolveu pesquisar preços em três supermercados. A matriz P dos preços está representada a seguir; a primeira linha contém os preços por kg do supermercado A; a segunda, do supermercado B; a terceira, do supermercado C. Esses preços são relativos, respectivamente, aos produtos feijão, linguiça, tomate e cebola.

P = [[2.05, 9.89, 2.48, 1.78], [1.93, 11.02, 2.00, 1.60], [1.70, 10.80, 2.40, 1.20]]
Q = [[5], [3], [2], [3]]

Sabendo que a matriz Q representa as quantidades necessárias, respectivamente, de feijão, linguiça, tomate e cebola, em qual dos supermercados se economizará mais ao se efetuar tais compras?

20) Considere as informações das Tabelas 10 e 11, que descrevem a produção de grãos em dois anos consecutivos (CANDIDO, [2018?]).

Tabela 10 – Produção de grãos (em milhares de toneladas) durante o primeiro ano

	Soja	Feijão	Arroz	Milho
Região A	3.000	200	400	600
Região B	700	350	700	100
Região C	1.000	100	500	800

Fonte: Candido ([2018?]).

Descrição de Imagem: Tabela 10- A Tabela 10 apresenta quatro linhas e cinco colunas. Nas linhas, constam informações sobre três regiões (Regiões "A", "B" e "C") e, em suas colunas, informações sobre produtos agrícolas ("Soja", "Feijão", "Arroz" e "Milho"). A intersecção entre as linhas e colunas traz o valor de toneladas produzidas por região, de cada um dos produtos, no primeiro ano. Como valores dispostos na tabela, temos:

Região A: Soja: 3.000; Feijão: 200; Arroz: 400; Milho: 600
Região B: Soja: 700; Feijão: 350; Arroz: 700; Milho: 100
Região C: Soja: 1.000; Feijão: 100; Arroz: 500; Milho: 800

Tabela 11 – Produção de grãos (em milhares de toneladas) durante o segundo ano

	Soja	Feijão	Arroz	Milho
Região *A*	5.000	50	200	0
Região *B*	2.000	100	300	300
Região *C*	2.000	100	600	600

Fonte: Candido ([2018?]).

Descrição de Imagem: Tabela 11- A Tabela 11 apresenta quatro linhas e cinco colunas. Nas linhas, as denominadas Regiões "A", "B" e "C" e, nas colunas, as denominações dos produtos "Soja", "Feijão", "Arroz" e "Milho". A intersecção entre as linhas e colunas traz o valor de toneladas produzidas por região, de cada um dos quatro produtos, no segundo ano. Como valores dispostos na tabela, temos:

Região A: Soja: 5.000; Feijão: 50; Arroz: 200; Milho: 0
Região B: Soja: 2.000; Feijão: 100; Arroz: 300; Milho: 300
Região C: Soja: 2.000; Feijão: 100; Arroz: 600; Milho: 600

Elabore uma matriz que estabeleça a produção por produto e por região nos dois anos conjuntamente.

21) Considere as mesmas informações da Questão 20, que descrevem a produção de grãos de três regiões em dois anos consecutivos, e a seguinte situação:

Existem muitos incentivos para se incrementar a produção, condições climáticas favoráveis, etc., de tal forma que a previsão para a safra do terceiro ano será o dobro da produção do primeiro mais a produção do segundo ano. Obtenha a matriz de estimativa da produção de grãos do terceiro ano.

22) Em um torneio, composto por apenas três times, cada um joga apenas uma vez contra os outros dois. Ao final, será declarada campeã a equipe que obtiver o maior número de pontos. Caso duas ou mais equipes cheguem ao final com o mesmo número de pontos, serão considerados os seguintes critérios de desempate: saldo de gols, maior número de gols pró e sorteio.

Os times foram numerados da seguinte maneira: Vasco: 1, Flamengo: 2 e Botafogo: 3.

Os resultados dos jogos foram tabulados na matriz quadrada *A*, de ordem 3, dada a seguir, onde *aij* é igual ao número de gols que a equipe *i* marcou na equipe *j*. Se houver gol contra, este será creditado para a outra equipe (FREITAS; SOUZA, 2012).

A=[[0, 5, 4], [1, 0, 0], [2, 6, 0]]

a) Com os dados da matriz, complete a Tabela 12:

Tabela 12 – Resultados dos Jogos

	Resultado
Vasco e Flamengo	
Vasco e Botafogo	
Flamengo e Botafogo	

Fonte: Freitas e Souza (2012).

Descrição de Imagem: Tabela 12- A Tabela 12 possui quatro linhas e duas colunas. A partir da segunda linha, os nomes dos times correspondentes às partidas disputadas: "Vasco e Flamengo", "Vasco e Botafogo" e "Flamengo e Botafogo”, respectivamente. Na segunda coluna, espaços em branco para que sejam armazenados os resultados de cada um destes três jogos.

b) No referido torneio, foi instituída a seguinte regra: "O número de pontos que cada time ganha por partida é igual ao quadrado do número de gols marcados pelo time nesta partida". Sabendo dos resultados dados na matriz *A*, complete a Tabela 13:

Tabela 13 – Pontos e Colocação de Cada Time

	Pontos no torneio	Colocação
Botafogo		
Flamengo		
Vasco		

Fonte: Freitas e Souza (2012).

Descrição de Imagem: Tabela 13- A tabela 13 apresenta quatro linhas e três colunas. As linhas correspondem aos nomes dos três times: "Botafogo", "Flamengo" e "Vasco". Há duas colunas denominadas: "Pontos no Torneio" e "Colocação". Os pontos no torneio e colocação (intersecções entre os nomes dos times, pontos e respectiva colocação) estão em branco, para serem preenchidos com o resultado do exercício.

23) Aplica-se pesticida nas plantas para eliminar os insetos daninhos, e certa parte destes pesticidas são absorvidos pela planta. Depois, os pesticidas acabam também sendo absorvidos pelos herbívoros que comem essas plantas. Suponha que temos três tipos de pesticidas e quatro tipos de plantas. Denotando por *aij* a quantidade do pesticida *i* (em miligramas) que foi absorvida pela planta *j*, essa informação pode ser representada como segue (CANDIDO, [2018?]):

Tabela 14 – Pesticidas e Plantas

	Planta 1	Planta 2	Planta 3	Planta 4
Pesticida 1	2	3	4	3
Pesticida 2	3	2	2	5
Pesticida 3	4	1	6	4

Fonte: Candido ([2018?]).

Descrição de Imagem: Tabela 14- A Tabela 14 apresenta quatro linhas e cinco colunas. Nas colunas, as denominações: "Planta 1", "Planta 2", "Planta 3" e "Planta 4"; nas linhas: "Pesticida 1", "Pesticida 2" e "Pesticida 3". A intersecção entre as linhas e colunas representa a quantidade, em miligramas, de cada pesticida em cada uma das plantas. Como valores dispostos na tabela, temos:

Pesticida 1: Planta 1: 2; Planta 2: 3; Planta 3: 4; Planta 4: 3.

Pesticida 2: Planta 1: 3; Planta 2: 2; Planta 3: 2; Planta 4: 5.

Pesticida 3: Planta 1: 4; Planta 2: 1; Planta 3: 6; Planta 4: 4.

Suponha, agora, que temos três herbívoros. Denotando por *bij* o número de plantas do tipo *i* que um herbívoro do tipo *j* come por mês, esta informação pode ser representada como mostra a Tabela 15:

Tabela 15 – Herbívoros e Plantas

	Herbívoro 1	Herbívoro 2	Herbívoro 3
Planta 1	20	12	8
Planta 2	28	15	15
Planta 3	30	12	10
Planta 4	40	16	20

Fonte: Candido ([2018?]).

Descrição de Imagem: Tabela 15- A Tabela 15 apresenta cinco linhas e 4 colunas. Nas linhas, as denominações: "Planta 1", Planta 2", "Planta 3" e "Planta 4"; nas colunas: "Herbívoro 1", "Herbívoro 2" e "Herbívoro 3". Nas linhas da tabela e nas colunas, temos a quantidade de plantas que cada herbívoro come em um mês. Assim, como valores dispostos na tabela, temos:

Planta 1: Herbívoro 1: 20; Herbívoro 2: 12; Herbívoro 3: 8.

Planta 2: Herbívoro 1: 28; Herbívoro 2: 15; Herbívoro 3: 15.

Planta 3: Herbívoro 1: 30; Herbívoro 2: 12; Herbívoro 3: 10.

Planta 4: Herbívoro 1: 40; Herbívoro 2: 16; Herbívoro 3: 20.

Por meio da multiplicação das matrizes que representam esses dados, determine a quantidade de cada tipo de pesticida absorvida por cada tipo de herbívoro.

Vamos praticar

Vamos escrever o código-fonte deste exercício em *Python:*

```
Pesticidas=[[2, 3, 4, 3], [3, 2, 2, 5], [4, 1, 6, 4]]
Herbivoros=[[20, 12, 8], [28, 15, 15], [30, 12, 10], [40, 16, 20]]
Qtde=[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
# Impressão da matriz/tabela Pesticidas
print("Valores da Tabela de Pesticidas")
for linha in range(3):
    for coluna in range(4):
        print("Linha:",linha+1,"Coluna:",coluna+1,\
       "Valor armazenado:",Pesticidas[linha][coluna])
# Impressão da matriz/tabela herbívoros
print("Valores da Tabela de Herbívoros:")
for linha in range(4):
    for coluna in range(3):
        print("Linha:", linha+1,"Coluna:", coluna+1, "Valor:", Herbivoros [linha]
[coluna])
# for linha de acordo com o número de linhas da matriz/tabela Pesticidas (três
linhas)
# for coluna de acordo com o número de colunas da matriz/tabela herbívoros
B (três colunas)
# for k de acordo com o número de colunas da matriz/tabela Pesticidas e de
linhas da matriz/tabela herbivoros
for linha in range(3):
    for coluna in range(3):
        for k in range(4):
                    Qtde[linha][coluna]=Qtde[linha][coluna]+Pesticidas[linha]
[k]*Herbivoros[k][coluna]
# Impressão da matriz/tabela com as quantidades de pesticida por herbívoros
print("Valores da Tabela de Quantidades de Pesticida por Herbívoros:")
for linha in range(3):
    for coluna in range(3):
        print("Linha:", linha+1, "Coluna:", coluna+1, "Valor armazenado:", Qtde
[linha][coluna])
```

Interatividade: Utilize o *IDLE Editor* para realizar a atividade proposta.

Atenção: O código-fonte deste exercício encontra-se no Apêndice deste *e-book*.

24) Verifique se a matriz *A = [[2, -5], [-1, 3]]* é a matriz inversa da matriz *B = [[3, 5], [1, 2]]*.

25) Dadas as matrizes *A = [[2, 2], [1, 4]]* e *B = [[1, 2], [1, 1]]*, calcule a expressão *AB - 6Ai*.

26) Quais dos itens seguintes apresentam matrizes inversas entre si?

a) *A = [[2, 3], [1, 4]]* e *B = [[4/3, -1], [2, 5]]*

b) *A = [[3, 4], [1, 0]]* e *B = [[0, 1], [1/4, -3/4]]*

c) *A = [[1, 0, 0], [1, 3, 1], [1, 2, 0]]* e *B = [[1, 0, 0], [-1/2, 0, 1/2], [1/2, 1, -3/2]]*

27) Para que valores de *x* a matriz *A = [[1, x], [1, 4]]* admite inversa?

28) Calcule a inversa da matriz *B = [[2, 5], [1, 4]]*.

29) Dadas as matrizes *A = [[-2, 3], [1, 5]]* e *B = [[5, 3], [-2, -1]]*, estabeleça a matriz resultante do cálculo *2A+2ABi*.

30) Considere a Tabela 16, em que cada letra do alfabeto é representada por um número. Com esta tabela, pode-se elaborar uma matriz *M*, que corresponde numericamente a uma mensagem que pode ser criptografada.

TABELA 16 - Alfabeto com respectivos valores numéricos

A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*
1	2	3	4	5	6	7	8	9
J	K	L	M	N	O	P	Q	R
10	11	12	13	14	15	16	17	18
S	T	U	V	W	X	Y	Z	Espaço
19	20	21	22	23	24	25	26	27

FONTE: Autores.

Descrição de Imagem: Tabela 16- A Tabela 16 apresenta seis linhas e nove colunas, contendo as letras do alfabeto (de A até Z) e, para cada letra, um valor numérico, iniciando com A igual a 1. O espaço em branco é representado pelo valor 27. Como valores dispostos na tabela, temos:

A: 1 B: 2 C: 3 D: 4 E: 5 F: 6 G: 7 H: 8 I: 9

J: 10 K: 11 L: 12 M: 13 N: 14 O: 15 P: 16 Q: 17 R: 18

S: 19 T: 20 U: 21 V: 22 W: 23 X: 24 Y: 25 Z: 26 Espaço: 27

Se você fosse o receptor da mensagem criptografada, sabendo que a matriz codificadora é

A=[[3, 4], [2, 3]]
e sabendo também que

AM=[[117, 23, 156, 7, 89, 63, 92, 87, 28, 35] , [87, 17, 113, 5, 66, 45, 68, 65, 20, 25]]

Então, calcule a matriz inversa de *A* (denominada matriz *Ai*) e, por meio da multiplicação *(Ai)(AM),* encontre a mensagem original utilizando a Tabela 16.

Respostas das Atividades

1) *A= [[2, 1], [5, 4]]*

2) A soma dos elementos *a13, a22* e *a34* resulta em *14*

3) Soma dos elementos *a11*, *a22*, *a33*, *a31*, *a22* e *a13* resulta em *42*

4) Soma dos elementos de *A* resulta em *50*

5) Alternativa "b"

6) a) *[[3, 0, 0], [0, 6, 0], [0, 0, 9]]*
b) *[[2, 0, 0, 0], [0, 4, 0, 0], [0, 0, 6, 0], [0, 0, 0, 8]]*

7) Alternativa "a"

8) *a=1* e *b=2*

9) *a=5/2*, *b=3/4*, *c=8/5* e *d=6/5*

10) a) *R$ 1.620,00*
b) loja 1: *R$ 2.020,00*; loja 2: *R$ 1.650,00*; loja 3: *R$ 2.100,00* e loja 4: *R$ 2.120,00*
c) *R$ 2.300,00*; terceiro dia; loja 3
d) *R$ 1.400,00*; primeiro dia; loja 2

11) Bianca fez o maior número de ligações (24 ligações), enquanto Adriana recebeu o maior número de ligações (27 ligações)

12) a) *2A+C = [[8, -10], [1, 2]]*
b) *AB = [[8, 1, 5], [2, -1, -3]]*
c) Não é possível, pois *B* e *C* não têm a mesma ordem.
d) Não é possível, pois o número de colunas de *B* não é igual ao número de linhas de *C*.
e) *(A*-3C)B = [[-4, -23, -79], [-13, -1, -6]]
f) A**2 = *[[4, -3], [0, 1]]*

13) *[[0, 2, 0], [2, -5, 4], [6, -5, 2]]*

14) A multiplicação das duas matrizes nos dará o preço base (custo) de cada salgado. Assim, temos:
[[3, 6, 1, 3], [4, 4, 2, 2], [1, 1, 1, 6]] [[0.20], [0.30], [0.50], [0.80]]=[[5.30], [4.60], [5.80]]
Portanto, o preço base (sem prejuízo) de cada salgado deverá ser: Pastel: *R$ 5,30*; Empada: *R$ 4,60* e Quibe: *R$ 5,80*.

15*) A**2=[[-2, 2], [-3, 3]] [[-2, 2], [-3, 3]]*
*A**2=[[(-2)(-2)+(2)(-3), (-2).(2)+(2)(3)], [(-3)(-2)+(3)(-3), (-3)(2)+(3)(3)]]*
*A**2= [[-2, 2], [-3, 3]= A* e, portanto, a matriz *A* é idempotente.

16) Temos que a matriz *A+At* é simétrica.

17) A primeira coluna corresponde ao custo e a segunda coluna, ao lucr*o:*
[[16000, 9000], [3000, 2000], [20000, 9000]].

18) 1ª) Loja A; modelo 2; 5 unidades
2ª) *A+B = [[3, 4, 4, 5], [3, 5, 2, 8]]*
3ª) *A-B = [[-1, 2, 0, 3], [3, -1, -2, 2]]*
19) Haverá mais economia se forem efetuadas compras no supermercado C, cujo valor a ser pago será R$ 49,30.

20) *[[8000, 250, 600, 600], [2700, 450, 1000, 400], [3000, 200, 1100, 600]].*

21) *[[11000, 450, 1000, 1200], [3400, 800, 1700, 500], [4000, 300, 1600, 2200]]*

22) a) As respostas da primeira parte do exercício 22 estão apresentadas na Tabela 17.

Tabela 17 – Respostas da Primeira Parte do Exercício 22

Jogo	Resultado
Vasco e Flamengo	*5 x 1*
Vasco e Botafogo	*4 x 2*
Flamengo e Botafogo	*0 x 6*

Fonte: Autores.

Descrição de Imagem: Tabela 17- A Tabela 17 possui quatro linhas e duas colunas. Nas linhas, constam informações correspondentes às partidas de jogos disputados entre: "Vasco e Flamengo", "Vasco e Botafogo" e "Flamengo e Botafogo", respectivamente. Como valores dispostos na tabela, temos:

Jogo: Vasco e Flamengo. Resultado: 5x1
Jogo: Vasco e Botafogo. Resultado: 4x2
Jogo: Flamengo e Botafogo. Resultado: 0x6

b) As respostas da segunda parte do exercício 22 são apresentadas na Tabela 18.

Tabela 18 – Respostas da Segunda Parte do Exercício 22.

Time	Pontos no Torneio	Colocação
Botafogo	*40*	*2º*
Flamengo	*1*	*3º*
Vasco	*41*	*1º*

Fonte: Autores.

Descrição de Imagem: Tabela 18- A Tabela 18 apresenta quatro linhas e três colunas. Nas linhas, constam os nomes dos times: "Botafogo", "Flamengo" e "Vasco". Nas colunas da tabela, temos as denominações: "Pontos no Torneio" e "Colocação". Como valores dispostos na tabela, temos:

Time: Botafogo Pontos: 40 Colocação: 2º
Time: Flamengo Pontos: 1 Colocação: 3º
Time: Vasco Pontos: 41 Colocação: 1º

23) *[[364, 165, 161], [376, 170, 174], [448, 199, 187]]*

24) Sendo que $BA = I2$, temos que a matriz B é a inversa da matriz A.

25) *AB - 6Ai =[[0, 8], [6, 4]]*

26) "b" e "c".

27) *x != 4.*

28) *Bi =[[4/3, -5/3], [-1/3, 2/3]].*

29) *2A+2ABi =[[12, 48], [20, 54]]*

30) CAPACIDADE E ATITUDE

3

DETERMINANTES

INTRODUÇÃO

No estudo das matrizes, definimos que uma matriz quadrada é aquela que apresenta o mesmo número de linhas e de colunas, ou seja, que é uma matriz de ordem *nxn*. Sendo assim, estudaremos que, a toda matriz quadrada, está associado um número denominado "determinante", sendo que o principal objetivo desta unidade é o de conhecer as regras matemáticas voltadas ao cálculo dos determinantes de matrizes de ordem um, dois, três (Regra de *Sarrus*) e também de matrizes de ordem maior que três (Teorema de *Laplace*).

Há regras e propriedades que especificam as formas de cálculo de um determinante, e estes tópicos serão abordados ao longo desta unidade, juntamente com algumas aplicações dos determinantes, que abordam, por exemplo, o cálculo da área de um triângulo situado no plano cartesiano, quando são conhecidas as coordenadas dos seus vértices.

Faremos, também, exemplos práticos utilizando a linguagem de programação *Python*. Os códigos-fonte em *Python* serão explicados de forma detalhada e inseridos em Apêndice.

3.1 CÁLCULO DE UM DETERMINANTE

Na área de Matemática, um determinante de uma matriz é uma função matricial que associa a cada matriz quadrada um número escalar, ou seja, é uma função que transforma uma matriz quadrada em um número real (CAMPAGNER, 2021). Vamos, então, aprender a calcular os determinantes de matrizes quadradas de ordem um, dois, três (Regra de *Sarrus*) e também de matrizes de ordem maior que três (Teorema de *Laplace*).

3.1.1 Determinante de uma Matriz de Ordem 1

Dada uma matriz quadrada de ordem 1, ou seja, que apresenta uma linha e uma coluna, o seu determinante é o próprio elemento $a11$. Assim, se $A = (a11)$ então $det(A) = a11$.

Exemplos:

$A = (2)$ então $det(A) = 2$.
$B = (-5)$ então $det(B) = -5$.

3.1.2 Determinante de uma Matriz de Ordem 2

Dada uma matriz de ordem 2, o seu determinante é o número real obtido pela diferença entre o produto dos elementos da diagonal principal e o produto dos elementos da diagonal secundária, ou seja, se

$$A = [[a11, a12], [a21, a22]]$$

então

$$det(A) = det[[a11, a12], [a21, a22]] = a11a22 - a21a12$$

sendo que a diagonal principal é dada pelos elementos $a11$ e $a22$ e a diagonal secundária é dada pelos elementos $a21$ e $a12$.

Exemplo:
Calcule o determinante da matriz A de ordem 2, cujos elementos devem ser calculados a partir da lei de formação: $aij = 2i+j$.

Resolução:

Primeiramente, devemos calcular cada um dos elementos que irão compor a matriz *A*, adotando a lei de formação dada no exercício: *aij = 2i+j*

a11=2(1)+1=3

a12=2(1)+2=4

a21=2(2)+1=5

a22=2(2)+2=6

Assim, temos a matriz *A = [[3, 4], [5, 6]]*. Logo, *det(A) = det[[3, 4], [5, 6]] = (3)(6) - (5)(4) = -2.*

Vamos praticar

Vamos criar um novo arquivo no *IDLE Editor*, denominado *determinante1.py*, para fazermos o exemplo anterior em *Python*, por meio do seguinte código-fonte:

```
# inicialização das variáveis
A=[[3, 4],[5, 6]]
produtodiagonalprincipal=1
produtodiagonalsecundaria=1
determinante=0
# Impressão da matriz A
print("Valores armazenados na matriz A:")
for i in range(2):
    for j in range(2):
        print("Linha:",i+1,"Coluna:",j+1,"Valor armazenado:",A[i][j])
# Cálculo do produto dos elementos da diagonal principal e da diagonal se-
cundária
for i in range(2):
    for j in range(2):
        if i==j:
            produtodiagonalprincipal=produtodiagonalprincipal*A[i][j]
        else:
            produtodiagonalsecundaria=produtodiagonalsecundaria*
A[i][j]
# Cálculo do determinante
determinante=produtodiagonalprincipal-
produtodiagonalsecundaria
print("O determinante da matriz A é:",determinante)
```

INTERATIVIDADE: Utilize o *IDLE Editor* para realizar a atividade proposta.

ATENÇÃO: O código-fonte do arquivo *determinante1.py* encontra-se no Apêndice deste *e-book*.

No exemplo, criamos duas variáveis para armazenar o produto dos elementos da diagonal principal e da diagonal secundária, respectivamente. Como estas variáveis são acumuladores e irão acumular o resultado de produtos (multiplicações), as mesmas foram inicializadas com o valor 1. Se tivéssemos inicializado essas variáveis com o valor zero, o resultado dos produtos seria zero.

O resultado da execução deste código-fonte é o seguinte:
Valores armazenados na matriz A:
Linha: 1 Coluna: 1 Valor armazenado: 3
Linha: 1 Coluna: 2 Valor armazenado: 4
Linha: 2 Coluna: 1 Valor armazenado: 5
Linha: 2 Coluna: 2 Valor armazenado: 6
O determinante da matriz A é: -2

Exemplo:
Resolva a equação *det[[2, x], [3, -3]]=det[[1, x], [x, -2]]*.

Resolução:
Desenvolvendo cada determinante, obtemos:

(2)(-3) - (3)(x) = (1)(-2)-(x)(x)
*-6-3x=-2-x**2*
*x**2-3x-4 = 0*

Resolvendo a equação anterior pela fórmula de Bhaskara, obtemos:

$x = 4$ ou $x = -1$
$S = \{-1 , 4\}$, sendo S o conjunto solução.

INTERATIVIDADE: Procure recordar a forma de se efetuar o cálculo das raízes de uma equação de segundo grau. Dica de site com explicações deste conteúdo (SILVA, 2021): https://brasilescola.uol.com.br/matematica/raiz-uma-equacao-2-grau-1.htm

3.1.3 Determinante de uma Matriz de Ordem 3 (Regra de *Sarrus*)

Dada uma matriz de ordem 3, o seu determinante é obtido utilizando uma regra prática denominada *regra de Sarrus*.

INTERATIVIDADE: Para acompanhar explicações e exemplos de aplicação da Regra de *Sarrus*, consulte o site (RIBEIRO, 2021): https://brasilescola.uol.com.br/matematica/regra-sarrus.htm

Seja a matriz *A = [[a11, a12, a13], [a21, a22, a23], [a31, a32, a33]]*.

Para aplicação da regra de *Sarrus,* devemos primeiramente repetir à direita da matriz a primeira coluna, constituída pelos elementos *a11*, *a21* e *a31,* e também repetir a segunda coluna, constituída pelos elementos *a12*, *a22* e *a32*. Na sequência, multiplicamos entre si os elementos que estão dispostos nas três diagonais principais da matriz ampliada *(a11, a22* e *a33)*, *(a12, a23* e *a31)* e *(a13, a21* e *a32),* conservando os sinais destes três produtos. Depois, devemos multiplicar entre si os elementos que estão dispostos nas três diagonais secundárias da matriz ampliada *(a31, a22* e *a13)*, *(a32, a23* e *a11)* e *(a33, a21* e *a12),* invertendo os sinais desses três produtos. Resumidamente:

$$det(A) = (a11)(a22)(a33) + (a12)(a23)(a31) + (a13)(a21)(a32) - (a31)(a22)(a13) - (a32)(a23)(a11) - (a33)(a21)(a12)$$

Exemplo:

Calcule o determinante da matriz *A = [[3, 9, 7], [-2, 4, 5], [-1, 8, 10]]*.

Resolução:

Pela regra de *Sarrus*, repetindo as duas primeiras colunas à direita da matriz *A,* promovendo a multiplicação entre os elementos de cada uma das três diagonais principais da matriz ampliada (conservando os sinais destes três produtos) e multiplicando entre si os elementos que estão dispostos nas três diagonais secundárias da matriz ampliada (invertendo os sinais desses três produtos), obtemos:

$$det(A) = (3)(4)(10) + (9)(5)(-1) + (7)(8)(-2) - (-1)(4)(7) - (8)(5)(3) - (10)(9)(-2)$$
$$det(A) = 120 - 45 - 112 + 28 - 120 + 180$$
$$det(A) = 51$$

Vamos praticar

Vamos criar um novo programa, denominado *determinante2.py*, e digitarmos o código-fonte abaixo, de acordo com o exemplo, para calcularmos o determinante da matriz *A = [[3, 9, 7], [-2, 4, 5], [-1, 8, 10]]* pela regra de *Sarrus*:

```
# Inicialização da Matriz A (3 linhas e 3 colunas)
A = [[3, 9, 7], [-2, 4, 5], [-1, 8, 10]]
# Inicialização da Matriz ampliada - Matriz B
# 3 linhas e 5 colunas - copiando as duas primeiras colunas de A à direita de B
B = [[3, 9, 7, 3, 9],[-2, 4, 5, -2, 4],[-1, 8, 10, -1, 8]]
# o determinante deve ser calculado de acordo com a fórmula
# det(B) = (B11)(B22)(B33) + (B12)(B23)(B31) + (B13)(B21)(B32) - (B31)(B22)(B13)-(B32)(B23)(B11)- (B33)(B21)(B12)
# como os índices em Python iniciam em zero, o elemento B11 é, na verdade, B00, B22 é B11 e assim por diante
# Cálculo do determinante
det=((B[0][0]*B[1][1]*B[2][2])+(B[0][1]*B[1][2]*B[2][3])+(B[0][2]*B[1][3]*B[2][4]))-\
((B[2][0]*B[1][1]*B[0][2]))-((B[2][1]*B[1][2]*B[0][3]))-((B[2][2]*B[1][3]*B[0][4]))
print("O determinante da matriz A é:",det)
```

Atenção: O código-fonte do arquivo *determinante2.py* encontra-se no Apêndice deste *e-book*.

Interatividade: Utilize o *IDLE Editor* para realizar a atividade proposta.

Como os índices das matrizes (listas em *Python*) começam em zero, é preciso verificar os índices corretamente para serem utilizados no cálculo do determinante. Por exemplo, a posição *A11*, na matriz em *Python*, é equivalente a *A00*. Por isso, os índices utilizados no cálculo do determinante, em nosso exemplo de código-fonte, estão diferentes. Criamos a matriz ampliada *B* para utilizarmos no cálculo do determinante. Além disso, utilizamos no final da primeira linha da fórmula, para o cálculo do determinante (armazenado na variável *det*), uma barra invertida (\), para que pudéssemos quebrar a linha de código.

O resultado da execução desse código-fonte é apresentado abaixo:

O determinante da matriz A é: 51

Exemplo:

Qual o valor de *x* que satisfaz a equação *det[[1, 2, 3], [x, 1, x], [1, 0, 2]] = 11*?

Resolução:

Repetindo a primeira e a segunda coluna à direita da matriz dada e efetuando o cálculo do determinante pela regra de *Sarrus*, temos a equação:

$$(1)(1)(2) + (2)(x)(1) + (3)(0)(x) - (1)(1)(3) - (0)(x)(1) - (2)(2)(x) = 11$$
$$2 + 2x - 3 - 4x = 11$$
$$x = -6$$
$$S = \{-6\}$$

3.1.4 Determinante de uma Matriz de Ordem Maior que Três (Teorema de *Laplace*)

Antes de estudarmos o Teorema de *Laplace*, usado para calcular o determinante de uma matriz de ordem maior que três, vamos definir o que é um cofator.

> ***Cofator:*** Dada uma matriz *A = (aij)*, quadrada de ordem *n*, sendo *n* pertencente ao conjunto dos números naturais e *n >= 2*, denominamos cofator de *aij* o produto de *(-1)**(i+j)* pelo determinante *Dij*, proveniente da matriz que se obtém quando eliminamos da matriz *A* a *i*-ésima linha e a *j*-ésima coluna. O cofator do elemento *aij* será indicado por *Cij*.
> *Cij = (-1)**(i+j)Dij*

Por exemplo, considerando a matriz *A = [[a11, a12, a13], [a21, a22, a23], [a31, a32, a33]]*, de ordem 3, teremos o cofator *C21* na forma:

*C21 = (-1)**(2+1)D21*

sendo *D21* o determinante da matriz resultante da eliminação da segunda linha (dada pelos elementos *a21*, *a22* e *a23)* e da primeira coluna (dada pelos elementos *a11*, *a21* e *a31)* da matriz *A*. Assim:

*C21 = (-1)**(2+1)det[[a12, a13], [a32, a33]]*

ou seja

*C21 = (-1)**(3)((a12)(a33) - (a32)(a13)).*

Exemplo:

Dada a matriz $A = [[3 ,9, 7], [-2, 4, 5], [-1, 8, 10]]$. Calcule os cofatores $C11$, $C22$ e $C32$.

Resolução*:*

Cálculo do cofator *C11*:

$$C11 = (-1)**(1+1)det[[4, 5], [8, 10]]$$
$$C11 = (-1)**(2)((4)(10) - (8)(5))$$
$$C11 = 0$$

Cálculo do cofator *C22*:

$$C22 = (-1)**(2+2)det[[3, 7], [-1, 10]]$$
$$C22 = (-1)**(4)((3)(10) - (-1)(7))$$
$$C22 = 37$$

Cálculo do cofator *C32*:

$$C32 = (-1)**(3+2)det[[3, 7], [-2, 5]]$$
$$C32 = (-1)**(5)((3)(5) - (-2)(7))$$
$$C32 = -29$$

Teorema de *Laplace*: Considere a matriz $A = (aij)$ quadrada de ordem n, sendo n pertencente ao conjunto dos números naturais e $n >= 2$. O determinante dessa matriz é igual à soma dos produtos dos elementos de uma linha (ou de uma coluna) qualquer pelos respectivos cofatores.

INTERATIVIDADE: Para conhecer melhor a forma de se calcular determinantes de matrizes quadradas de ordem maior ou igual a três, consulte o site (OLIVEIRA, G., 2021): https://brasilescola.uol.com.br/matematica/teorema-laplace.htm

O teorema de *Laplace* pode ser utilizado para cálculo de determinante de matrizes (quadradas) de qualquer ordem $n >= 2$, diferentemente da regra de *Sarrus* que somente pode ser utilizada no cálculo de determinantes de matrizes de ordem 3.

Veja como calcular o determinante de uma matriz $A = [[a11, a12, a13], [a21, a22, a23], [a31, a32, a33]]$ pelo teorema de *Laplace*, escolhendo a primeira linha:

det(A) = a11C11 + a12C12 + a13C13
det(A) = a11(-1)**(1+1)D11 + a12(-1)**(1+2)D12 + a13(-1)**(1+3)D13
det(A) = a11(-1)**(2)det[[a22, a23], [a32, a33]] + a12(-1)**(3)det[[a21, a23], [a31, a33]] + a13(–1)**(4)det[[a21, a22], [a31, a32]].

Exemplo:

Calcule, usando o teorema de *Laplace*, o determinante das matrizes A = [[3, 9, 7], [-2, 4, 5], [-1, 8, 10]] e B = [[1, -1, 0, 2], [0, 0, 1, 4], [2, 1, 5, 3], [3, 2, 6, 8]].

Resolução:

Para cálculo do determinante da matriz *A*, vamos escolher a terceira linha e, assim, teremos:

det(A) = a31(-1)**(3+1)det[[a12, a13], [a22, a23]] + a32(-1)**(3+2)det[[a11, a13], [a21, a23]] + a33(-1)**(3+3)det[[a11, a12], [a21, a22]]
det(A) = (-1) (-1)**(4)det[[9, 7] [4, 5]] + (8)(-1)**(5)det[[3, 7] [-2, 5]] + (10)(-1)**(6) det[[3, 9] [-2, 4]]
det(A) = (-1) (1)(45 - 28) + (8)(-1)(15 + 14) + (10)(1)(12 + 18)
det(A) = - 17 - 232 + 300
det(A) = 51.

No caso da matriz *B*, o cálculo do determinante se tornará mais rápido se escolhermos a segunda linha para aplicação do teorema de *Laplace*, uma vez que dois de seus elementos são nulos *(a21 = 0* e *a22 = 0*) e, assim, tornam nulas duas multiplicações do cálculo do determinante.

det(B) = a21(-1)**(2+1)det[[a12, a13, a14], [a32, a33, a34], [a42, a43, a44]] + a22(-1)**(2+2)det[[a11, a13, a14], [a31, a33, a34], [a41, a43, a44]] + a23(-1)**(2+3)det[[a11, a12, a14], [a31, a32, a34], [a41, a42, a44]] + a24(–1)**(2+4) det[[a11, a12, a13], [a31, a32, a33], [a41, a42, a43]]

Assim, teremos:
det(B) = 0 + 0 + (1)(–1)**(5)det[[1, -1, 2], [2, 1, 3], [3, 2, 8]] + (4)(–1)**(6)det [[1, -1, 0], [2, 1, 5], [3, 2, 6]]

Para finalizar o cálculo do determinante da matriz *B*, temos que ainda calcular dois determinantes de ordem 3, o que pode ser efetuado, por exemplo, pela regra de *Sarrus*, e assim

det(B) = (–1)(8 – 9 + 8 – 6 – 6 + 16) + (4)(6 – 15 – 0 – 0 – 10 + 12)
det(B) = – 11 – 28 = –39.

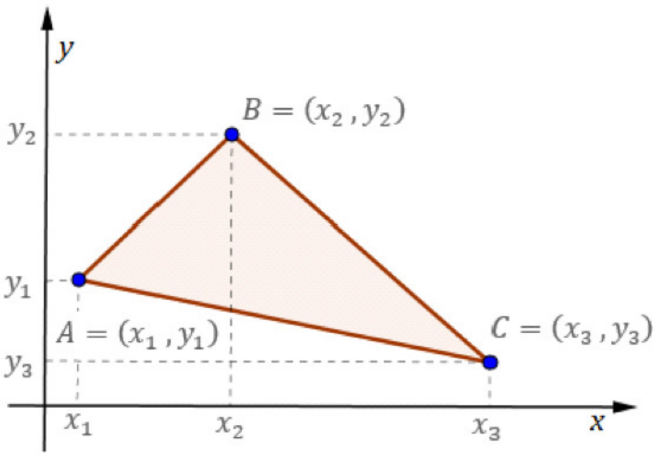
3.2
APLICAÇÃO DO CÁLCULO DE DETERMINANTE

Se pretendemos calcular a área de um triângulo, podemos utilizar apenas as coordenadas cartesianas dos vértices que definem esta região triangular. Se denotarmos os três vértices do triângulo por $A = (x1, y1)$, $B = (x2, y2)$ e $C = (x3, y3)$, então o cálculo da área do triângulo *ABC* pode ser obtido pela expressão:

$S= |D|/2$ sendo $D = det[[x1, y1, 1], [x2, y2, 1], [x3, y3, 1]]$.

Na fórmula anterior, o parâmetro *D* corresponde ao determinante da matriz que contém as coordenadas dos vértices do triângulo *ABC*. Ainda, na expressão para o cálculo da área do triângulo *ABC*, o parâmetro D está em módulo, ou seja, usaremos o seu valor absoluto por se tratar de um cálculo de área, que deve resultar em um valor positivo ou nulo (o valor correspondente à área será nulo quando os três vértices estiverem alinhados). Na Figura 15, temos um esboço de um triângulo com vértices em $A = (x1, y1)$, $B = (x2, y2)$ e $C = (x3, y3)$.

FIGURA 15 - Triângulo *ABC*



FONTE: Fortes et al. (2019).

Descrição de Imagem: Figura 15- A imagem possui dois eixos coordenados representados por linhas contínuas com setas na ponta. O eixo *x* é representado horizontalmente na parte de baixo da imagem e possui as marcações das abcissas dos pontos dos vértices do triângulo: *x1*, *x2*, *x3* e também a notação *x*. O eixo *y* é representado verticalmente no lado esquerdo da imagem e possui as marcações das ordenadas dos pontos dos vértices do triângulo: *y3*, *y1* e *y2* e também a notação *y*. O vértice *A* é a ligação de linhas perpendiculares tracejadas entre *x1* e *y1*, o vértice *B* é a ligação de linhas perpendiculares tracejadas entre *x2* e *y2* e o vértice

C é a ligação de linhas perpendiculares tracejadas entre *x3* e *y3*. Os vértices são ligados entre si por segmentos de reta que constituem os lados do triângulo.

Exemplo*:*

Determine a área do triângulo definido pelos vértices *A = (1, 2)*, *B = (3, 5)* e *C = (5, 1)*.

Resolução:

Para realizar este cálculo, teremos que determinar primeiramente o parâmetro *D*, sendo que, na sequência, vamos avaliar

$$S = |D|/2 \text{ sendo } D = det[[x1, y1, 1], [x2, y2, 1], [x3, y3, 1]].$$

Então, substituindo os valores das coordenadas dos vértices *A = (1, 2)*, *B = (3, 5)* e *C = (5, 1)* temos:

$$D = \det[[1, 2, 1], [3, 5, 1], [5, 1, 1]]$$

e utilizando a regra de *Sarrus*:

$$D = 5 + 10 + 3 - 25 - 1 - 6 = -14$$

Portanto:

$$S = |-14|/2 = (14)/2 = 7\ u.a.$$

Ou seja, o valor correspondente à área da região triangular situada entre os vértices *A = (1, 2)*, *B = (3, 5)* e *C = (5, 1)* é *7 u.a.* (sendo *u.a.* a abreviação para unidades de área).

3.3
PROPRIEDADES DOS DETERMINANTES

1ª propriedade: Se todos os elementos de uma linha ou coluna de uma matriz quadrada forem nulos, o seu determinante será zero.

Exemplo:

Se *M = [[0, 9, 7], [0, 4, 5], [0, 8, 10]]*, então *det(M) = det[[0, 9, 7], [0, 4, 5], [0, 8, 10]] = 0*

Vamos praticar

Vamos criar um novo arquivo, denominado *determinante3.py*, e verificar se uma matriz tem o determinante igual a zero, seguindo a regra de que, se todos os elementos de uma linha ou coluna de uma matriz quadrada forem nulos, o seu determinante será zero. Vamos fazer o teste para a matriz *M = [[0, 9, 7], [0, 4, 5], [0, 8, 10]]* utilizando o código-fonte abaixo:

```
M = [[0, 9, 7], [0, 4, 5], [0, 8, 10]]
# Impressão da matriz M
print("Valores da Matriz M:")
for linha in range(3):
    for coluna in range(3):
        print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",M[linha]
[coluna])
# Verificação de valores nulos
# percorrer as linhas da matriz - se pelo menos uma linha tiver todos os valores
nulos
# então o determinante da matriz será zero
det=0
matrizdetzero=0
for linha in range(3):
    for coluna in range(3):
        if M[linha][coluna]!=0:
            det=1
                if det==0:
        print("A matriz M tem determinante igual a zero!")
```

```
            matrizdetzero=1
            break
           det=0
     # vamos percorrer as colunas da matriz M se ainda não foram encontradas
linhas com valores todos nulos
if matrizdetzero==0:
   det=0
   for coluna in range(3):
         for linha in range(3):
              if M[linha][coluna]!=0:
                det=1
         if det==0:
             print("A matriz M tem determinante igual a zero!")
             matrizdetzero=1
            break
           det=0
if matrizdetzero==0:
   print("A matriz M não tem determinante igual a zero!")
```

Atenção: O código-fonte do arquivo *determinante3.py* encontra-se no Apêndice deste *e-book*.

Interatividade: Utilize o *IDLE Editor* para desenvolver esta atividade.

O resultado da execução deste código-fonte é o seguinte:

Valores da Matriz M:
Linha: 1 Coluna: 1 Valor armazenado: 0
Linha: 1 Coluna: 2 Valor armazenado: 9
Linha: 1 Coluna: 3 Valor armazenado: 7
Linha: 2 Coluna: 1 Valor armazenado: 0
Linha: 2 Coluna: 2 Valor armazenado: 4
Linha: 2 Coluna: 3 Valor armazenado: 5
Linha: 3 Coluna: 1 Valor armazenado: 0
Linha: 3 Coluna: 2 Valor armazenado: 8
Linha: 3 Coluna: 3 Valor armazenado: 10
A matriz M tem determinante igual a zero!

Vamos praticar

Altere os valores da matriz *M*, execute o programa novamente e verifique o resultado. Teste com os seguintes valores para a matriz *M*: *M = [[1, 0, 7], [2, 0, 5], [3, 0, 10]]* e *M = [[1, 1, 7], [2, 0, 5], [3, 0, 10]]*.

Interatividade: Utilize o *IDLE Editor* para desenvolver esta atividade.

2ª propriedade: Se os elementos correspondentes de duas linhas (ou de duas colunas) de uma matriz quadrada forem iguais, seu determinante será nulo.

Exemplo:

Se *E = [[3, 9, 7], [3, 9, 7], [-1, 8, 10]]* então *det(E) = det[[3, 9, 7], [3, 9, 7], [-1, 8, 10]] = 0*.

3ª propriedade: Se uma linha (ou coluna) de uma matriz quadrada for combinação linear de outras linhas (ou colunas), seu determinante será nulo.

Exemplo:

Dada a matriz *N = [[3, 2, 3], [-3, 5, -2], [0, 7, 1]]*, em que a terceira linha é a soma da primeira com a segunda linha, ou seja, a terceira linha é uma combinação linear da primeira com a segunda linha, logo

det(N) = det[[3, 2, 3], [-3, 5, -2], [0, 7, 1]] = 0.

4ª propriedade: Se uma matriz quadrada possui duas linhas (ou colunas) proporcionais, seu determinante será nulo.

Exemplo:

Se *B = [[5, 4, 1], [2, 7, 3], [10, 8, 2]]*, então *det(B) = det[[5, 4, 1], [2, 7, 3], [10, 8, 2]] = 0*, uma vez que a terceira linha da matriz *B* é proporcional à primeira linha (mais precisamente, os elementos da terceira linha são iguais ao dobro dos respectivos elementos da primeira linha).

5ª propriedade: O determinante de uma matriz quadrada é igual ao determinante de sua transposta.

Exemplo:

Sendo *A = [[1, 2, 3], [4, 3, 2], [2, 3, 5]]*, temos *det(A) = -5*, e para *At = [[1, 4, 2], [2, 3, 3], [3, 2, 5]]*, que é a transposta da matriz *A*, também teremos *det(At) = -5*.

6ª propriedade: Se trocarmos de posição entre si duas linhas (ou duas colunas) de uma matriz quadrada, o determinante da nova matriz é o oposto do determinante da primeira matriz.

Exemplo:

Dada a matriz *X = [[3, 1, 2], [1, 3, 5], [4, 3, 1]]*, temos que *det(X) = 9+20+6-24-45-1 = -35*. Agora, trocando as posições da primeira e terceira linhas de *X*, obtemos a matriz *Y = [[4, 3, 1], [1, 3, 5], [3, 1, 2]]*, sendo que *det(Y) = 24+45+1-9-20-6 = 35*, que é o oposto do *det(X)*.

7ª propriedade: Se multiplicarmos todos os elementos de uma linha (ou de uma coluna) por um número real *k*, então o determinante da nova matriz é o produto de *k* pelo determinante da matriz inicial.

Exemplo:

A = *[[2, -3, 1], [0, 5, 2], [2, 0, 2]]*, *det(A) = -2* e *B = [[6, -9, 3], [0, 5, 2], [2, 0, 2]]*, sendo que a matriz *B* foi obtida da matriz *A* multiplicando a primeira linha da matriz *A* por *3*, então *det(B) = 3(-2) = -6*.

8ª propriedade: Se todos os elementos de uma matriz quadrada situados de um mesmo lado da diagonal principal forem nulos, o determinante da matriz será igual ao produto dos elementos da diagonal principal.

Exemplo:

Dada a matriz quadrada *N = [[-2, 3, 5], [0, 9, 7], [0, 0, 8]]*, temos que o determinante da matriz *N* (denominada matriz triangular superior) corresponde à multiplicação dos termos de sua diagonal principal: *det(N) = (–2)(9)(8) = –144*.

9^a propriedade: Se uma matriz quadrada M de ordem n é multiplicada por um número real k, o seu determinante fica multiplicado por k^{n}, isto é:

$$det(kM) = k^{n}det(M)$$

Exemplo:

Seja $A = [[2, 5], [3, -4]]$, temos que $det(A) = -23$. Multiplicando a matriz A (de ordem 2) pelo número real 3, obtemos:

$$3A = [[(3)(2), (3)(5)], [(3)(3), (3)(-4)]] = [[6, 15], [9, -12]]$$

Assim, pela 9^a propriedade dos determinantes podemos antecipar o resultado deste cálculo, pois $det(3A)$ será igual ao $det(A)$ multiplicado por 3^{2}, ou seja: $det(3A) = 3^{2}(-23) = -207$.

E, de fato, por conferência, temos:

$$det(3A) = (6)(-12) - (9)(15) = -72 - 135 = -207.$$

10^a propriedade (Teorema de *Binet*): Sendo A e B duas matrizes quadradas de mesma ordem e AB a matriz-produto de A por B, então $det(AB) = det(A)det(B)$.

Exemplo:

Sejam $A = [[-4, 2], [-8, 5]]$ com $det(A) = -4$ e $B = [[2, 3], [7, 4]]$ com $det(B) = -13$. Temos que:

$$AB = [[-4, 2], [-8, 5]], [[2, 3], [7, 4]] = [[6, -4], [19, -4]]$$

Logo,

$$det(AB) = (6)(-4) - (19)(-4) = 52$$

Contudo, este mesmo resultado podemos encontrar usando a 10^a propriedade dos determinantes, ou seja: $det(AB) = det(A)det(B) = (-4)(-13) = 52$.

3.4 DETERMINANTE DE UMA MATRIZ INVERSA

No estudo das matrizes, vimos que $(A)(Ai) = (Ai)(A) = I$, sendo Ai a matriz inversa de A e I a matriz identidade, todas quadradas de mesma ordem.

Assim, da relação $(A)(Ai) = I$ podemos escrever $det((A)(Ai)) = det(I)$. Mas como $det(I) = 1$ para todo n pertencente ao conjunto dos números naturais, escrevemos então que $det((A)(Ai)) = 1$. Aplicando o teorema de *Binet* (10ª propriedade dos determinantes), escrevemos:

$$det(A)det(Ai) = 1$$

e supondo que $det(A)$ seja diferente de zero, podemos concluir:

$$det(Ai) = 1/(det(A))$$

Agora, se $det(A) = 0$, então não existirá matriz inversa.

Atividades - Unidade 3

1) Calcule o valor dos determinantes:

a) *det[[5, -5], [2, -3]]*

b) *det[[math.sqrt(5), math.sqrt(2)], [-math.sqrt(2), math.sqrt(5)]]*

c) *det [[3, 2, 5], [4, 1, 3], [2, 3, 4]]*

d) *det [[2, -3, 3], [4, -2, 4], [8, -5, 2]]*

e) *det [[0, 5, -2, 3], [0, 4, -3, 2], [0, 2, -5, 4], [1, 3, -4, 3]]*

f) *det [[2, 8, 0, 0], [1, 2, -3, 2], [6, 2, -6, 2], [4, 5, -4, 3]]*

2) Na equação a seguir, envolvendo determinantes, encontre os valores reais de *x:*

det[[2, 0, 0], [1, -1, 0], [2, 7, x]] + det[[0, x, 1], [3, 3x, 0], [-3, x, 2]] = 14

3) Sendo *AB* o produto das matrizes *A = [[x, 3], [2, 1]]* e *B = [[1, 5], [2, 0]]*, calcule o valor (ou valores) de *x* (pertencente ao conjunto dos números reais) para que o determinante de *AB* seja nulo.

4) Determine o conjunto solução das equações, aplicando a regra de *Sarrus*:

a) *det[[5, x, 3], [3, -3, 2], [2, 7, 1]] = 0*

b) *det [[-3, 4, x], [6, 0, 0], [4, 2, 3]] = 0*

5) Resolva as equações:

a) *det[[3(x-1), -(x-3)], [x+3, x+1]] = 10*

b) *det[[4/5, 2, 0], [2/3, x, x], [0, 3, 5/4]] = 0*

6) Calcule os determinantes, aplicando as propriedades. Indique a propriedade utilizada.

a) *det[[3, 9, 0], [2, 7, 0], [1, 4, 0]]*

b) *det[[2, 2, 2], [7, 9, 8], [2, 2, 2]]*

c) *det[[0, 0, 0, 0], [5, 4, 3, 2], [7, 8, 3, 5], [9, 2, 4, 3]]*

d) *det[[3, 5, 6], [-3, -4, 2], [0, 1, 8]]*

e) *det[[1, 2, 4, 8], [5, 6, 7, 9], [-3, -2, 2, 3], [8, 8, 5, 6]]*

7) Obtenha a matriz *A* de ordem 3 definida por elementos, tal que:

aij = i+j se i > j;

aij = i**3 se i = j;

aij = j-i se i < j.

Calcule o determinante desta referida matriz.

Respostas das Atividades

1) a) *-5* b) *7* c) *15* d) *-52* e) *28* f) *-136*

2) $x = 7/2$

3) $x = 6$

4) a) S={4} b) S={6}

5) a) S={-*math.sqrt(11/2)* , *math.sqrt(11/2)*} b) S= { *-25/21*}

6) a) 0 (1ª propriedade) b) 0 (2ª propriedade) c) 0 (1ª propriedade)
d) 0 (3ª propriedade) e) 0 (3ª propriedade)

7) *A = [[1, 1, 2,], [3, 8, 1], [4, 5, 27]]* e *det(A) = 100*

4

SISTEMAS LINEARES

INTRODUÇÃO

Nesta unidade, faremos um estudo introdutório dos sistemas de equações lineares, iniciando com definições, exemplos e técnicas de resolução.

O fato de sabermos solucionar um sistema de equações lineares torna-se relevante em função das inúmeras aplicações práticas que estão associadas a este conteúdo de Álgebra Linear. Atualmente, tais aplicações estão presentes nas mais variadas áreas, como na Astronomia, Biologia, Computação, Demografia, Física, Economia, Engenharia, Geografia, Navegação, Cartografia, Aviação e em muitas outras.

Estudaremos diferentes métodos para resolvermos um sistema de equações lineares – tais como os métodos da substituição, da adição – bem como a aplicação da biblioteca *NumPy* em *Python*. Também, abordaremos os sistemas lineares equivalentes e a classificação dos sistemas lineares.

4.1
EQUAÇÃO LINEAR

Toda a equação da forma $a1X1 + a2X2 + a3X3 + ... +anXn = b$ é denominada equação linear. Nesta equação, temos que:

- ► $a1, a2 , ... , an$ são números reais chamados coeficientes
- ► $X1, X2, X3, ... , Xn$ são as incógnitas
- ► b é o termo independente

Exemplos:

$3x – 2y + 7z = 8$ é uma equação linear, sendo que: 3, $–2$ e 7 são os coeficientes, x, y e z são as incógnitas e 8 é o termo independente.

$2x + 4y**2 = 3$ não é uma equação linear, pois a incógnita y apresenta grau 2.

$–3x + 4y = 0$ é uma equação linear, sendo que: -3 e 4 são os coeficientes, x e y são as incógnitas e o termo independente é nulo.

$3z = 2x – 7y + 9$ é uma equação linear que não está na "forma padrão". Podemos escrevê-la na forma $2x – 7y – 3z = –9$, sendo que: 2, $–7$ e $– 3$ são os coeficientes, x, y e z são as incógnitas e -9 é o termo independente.

$(1/math.sqrt(2))x = 5$ é uma equação linear com apenas a incógnita x, com coeficiente $1/math.sqrt(2)$ e com termo independente 5.

A solução de uma equação linear com n incógnitas é a sequência de números reais ou *ênupla* ($\alpha1$, $\alpha2$, $\alpha3$, ... , αn) que, colocados respectivamente no lugar de $X1, X2, X3, ... , Xn$, tornam verdadeira a igualdade dada.

Quando o termo independente b for igual a zero, a equação linear denomina-se equação linear homogênea.

Pela forma como é definida, uma equação é denominada linear quando não contém produtos, radicais ou outras funções não lineares em suas incógnitas. Em outras palavras, uma equação linear é uma equação composta exclusivamente de adições e subtrações de termos que são constantes ou são o produto de uma constante pela primeira potência de uma incógnita.

4.2 SISTEMAS LINEARES

Na área de Matemática, um sistema de equações lineares (ou abreviadamente sistema linear) é um conjunto finito de equações lineares aplicadas em um mesmo conjunto, igualmente finito, de incógnitas. A palavra "sistema" indica que as equações devem ser consideradas em conjunto, e não de forma individual (POOLE, 2004). Neste *e-book*, vamos partir da análise da situação-problema para iniciar o estudo dos sistemas de equações lineares.

Situação-problema: Na lanchonete de uma universidade, os pastéis (de tamanho médio) têm preço único e os sucos naturais (copo de 300 ml) também. Um aluno do curso de Sistemas de Informação pagou R$16,00 por 2 pastéis e 3 copos de suco natural, e seu colega pagou R$29,00 por 4 pastéis e 5 copos de suco natural. Qual o preço do pastel e do copo de suco natural?

Resolução:

Para equacionar esse problema, vamos chamar de:
x: preço de cada pastel.
y: preço do copo de suco natural.

Assim, conforme especificado, no caso do primeiro aluno, sabemos que ele comprou 2 pastéis e 3 copos de suco natural e que pagou R$16,00. Portanto, se x é o preço de cada pastel e y é o preço de cada copo de suco, podemos estabelecer a equação linear: $2x + 3y = 16$.

No caso do segundo aluno, sabemos que ele comprou 4 pastéis e 5 copos de suco natural e que pagou R$ 29,00. Então, considerando a especificação dos preços, podemos escrever: $4x + 5y = 29$.

O preço respectivo de cada pastel (preço denominado x) e de cada copo de suco natural (preço denominado y) não se altera de uma situação para outra, e assim estas equações passam a compor um sistema de equações, devendo ser consideradas em conjunto. Então, passamos a escrever o sistema linear composto pelas equações: $2x + 3y = 16$ e $4x + 5y = 29$.

A notação matemática correspondente a um sistema linear considera que cada equação deve ocupar uma linha em separado, e há um abre chaves compreendendo todas as equações que constituem o sistema. Para fins de representação, usaremos cada equação separada em uma linha terminada por ponto e vírgula. Por exemplo, podemos representar o sistema do problema anterior na forma:

$$2x + 3y = 16;$$
$$4x + 5y = 29;$$

Temos, assim, um sistema de duas equações lineares com duas incógnitas. Observe que o sistema linear obtido também pode ser conceituado como um sistema de equações do primeiro grau. Em outras palavras, em um sistema linear, não há potência diferente de um ou zero e tampouco pode haver multiplicação entre incógnitas.

4.3
SOLUÇÃO ALGÉBRICA DE UM SISTEMA LINEAR

Nesta subunidade, estudaremos duas formas de solução de um sistema linear: o método da substituição e o método da adição. Além destes métodos que remetem a cálculos algébricos, também faremos interpretações geométricas a respeito dos sistemas lineares, especificamente analisando aqueles que apresentam duas ou três incógnitas.

4.3.1 Método da Substituição

O método da substituição consiste em isolar uma das incógnitas em uma equação, substituindo em outra equação. Assim, obtemos uma nova equação com apenas uma incógnita.

Exemplo:

Vejamos como este método pode ser aplicado na resolução do sistema linear da situação-problema citada anteriormente:

$$2x + 3y = 16;$$
$$4x + 5y = 29;$$

Resolução:

Isolando a incógnita x na primeira equação do sistema dado, obtemos: $x=(-16-3y)/2$, e substituindo esta incógnita na segunda equação, obtemos:

$$4((16-3y)/2)+5y = 29$$

Resolvendo a equação anterior:

$$32 - 6y + 5y = 29$$
$$-y = 29 - 32$$
$$-y = -3$$
$$y = 3$$

Tendo obtido $y=3$, substituímos este valor na equação $x=(16-3y)/2$ e, assim, encontramos: $x=(16-3(3))/2$, ou seja, obtemos $x=3,5$.

Portanto, podemos concluir que o pastel custa R$ 3,50 e que o copo de suco natural custa R$ 3,00.

O par ordenado (3,50; 3,00) é a solução do sistema linear.

VAMOS PRATICAR

Vamos criar um novo arquivo no *IDLE Editor* e digitarmos o código-fonte abaixo, denominado *linear1.py*, sabendo que x=(16-3y)/2 e que 32 – 6y + 5y = 29:

```
for y in range(100):
  if 32-6*y+5*y==29:
    print("O valor de y é:",y)
    break
x=(16-3*y)/2
print("O valor de x é:",x)
```

INTERATIVIDADE: Utilize o *IDLE Editor* para desenvolver esta atividade.

ATENÇÃO: O código-fonte do arquivo *linear1.py* encontra-se no Apêndice deste *e-book*.

Neste exemplo, utilizamos um laço *for* para testar e validar o valor de *y*, em um intervalo de 0 a 99. Se, substituindo o valor de *y* na equação *32-6*y+5*y*, encontramos o resultado igual a 29, significa que encontramos o valor correto de *y* e podemos encerrar o laço de repetição (comando *break*). Sabendo-se o valor de *y*, podemos aplicá-lo na outra equação e encontrarmos o valor de *x*. O resultado da execução deste programa é o seguinte:

O valor de y é: 3
O valor de x é: 3.5

VAMOS PRATICAR

Por meio da biblioteca *NumPy* (NUMPY, 2021), vamos realizar um exemplo em *Python* para solucionarmos este sistema linear. Primeiro, é preciso instalar esta biblioteca. Abra o *prompt de comando* por meio do menu Iniciar do *Windows*. Na janela de *prompt de comando*, localize a pasta onde está instalado o *Python* em seu computador e digite o comando *python -m pip install --user numpy*.

Após instalar a biblioteca, para utilizá-la no código-fonte dos programas em *Python*, é preciso utilizar o comando *import numpy*.

INTERATIVIDADE: Acesse a biblioteca *NumPy* em: [HTTPS://numpy.org/devdocs/user/building.html](https://numpy.org/devdocs/user/building.html)

Vamos praticar

Vamos digitar o código-fonte abaixo como exemplo de utilização desta biblioteca. Criamos um novo arquivo, denominado *linear2.py*:

```
import numpy as np
lista=[[2, 3],[4, 5]]
A=np.array(lista)
inv_A=np.linalg.inv(A)
print("Matriz inversa de A:")
print(inv_A)
B=np.array([16,29])
X=np.linalg.inv(A).dot(B)
print("Valores das incógnitas X e Y:")
print(X)
```

Interatividade: Utilize o *IDLE Editor* para desenvolver esta atividade.

Atenção: O código-fonte do arquivo *linear2.py* encontra-se no Apêndice deste *e-book*.

Neste exemplo, importamos a biblioteca *NumPy* e a denominamos como *np*. Iniciamos criando a variável *lista*, contendo os coeficientes do sistema linear (sendo os valores 2, 3, 4 e 5). Pelo método *linalg.inv,* criamos a inversa desta matriz e, por meio do método *dot*, realizamos a multiplicação da inversa de *A* pela matriz *B*. O resultado da solução do sistema linear é armazenado na matriz *X*. Vejamos o resultado da execução deste código-fonte:

```
Matriz inversa de A:
[[-2.5  1.5]
 [ 2.  -1. ]]
Valores das incógnitas X e Y:
[3.5 3. ]
```

Atenção: A biblioteca *NumPy* pode ser utilizada para calcular a matriz inversa (método *inv*) e o produto entre as matrizes (método *dot*), como uma forma mais rápida e fácil de implementar os exemplos do capítulo anterior deste *e-book*.

4.3.2 Método da Adição

O método da adição consiste em somar duas equações, membro a membro, de tal forma que se elimine uma das incógnitas**:**

$$2x + 3y = 16;$$
$$4x + 5y = 29;$$

Multiplicando todos os termos da primeira equação do sistema anterior por (-2), obtemos um novo sistema linear:

$$-4x - 6y = -32;$$
$$4x + 5y = 29;$$

Somando as equações deste último sistema linear (soma de termos semelhantes, ou seja, que apresentam a mesma incógnita), obtemos:

$$-4x+4x-6y+5y=-32+29$$
$$-y = -3$$
ou seja
$$y = 3.$$

Agora, substituindo $y=3$ na primeira equação:

$$2x + 3y = 16$$
$$2x + 3(3)= 16$$
$$2x = 16 - 9$$
$$2x = 7$$
$$x = 7/2$$
ou ainda
$$x= 3{,}5$$

Logo, o pastel custa R$ 3,50 e o copo de suco custa R$ 3,00. O par ordenado (3,50, 3,00) é a solução do sistema linear.

Atenção: O conjunto de todas as soluções de um sistema é chamado de conjunto solução (S) ou conjunto verdade (V). No exemplo anterior, S={(3,5 , 3)}.

De uma maneira geral, denomina-se sistema linear de *m* equações nas *n* incógnitas $X1, X2, X3, \ldots, Xn$ a todo sistema da forma:

$$a11\ X1 + a12\ X2 + \ldots + a1nXn = b1;$$
$$a21X1 + a22X2 + \ldots + a2nXn = b2;$$

$$\dots$$

$$am1X1 + am2X2 + \dots + amnXn = bm;$$

em que $a11, a12, \dots, amn$, $b1, b2, \dots, bm$ são números reais.

Se o conjunto ordenado de números reais $(\alpha 1, \alpha 2, \alpha 3, \dots, \alpha n)$ satisfaz todas as equações do sistema, é denominado solução do sistema linear.

Se o termo independente de todas as equações do sistema for nulo, isto é, $b1= b2= \dots = bm= 0$, o sistema linear será dito homogêneo.

Por exemplo, uma solução do sistema linear homogêneo

$$2x - 5y + 3z = 0;$$
$$6x - 2y + 4z = 0;$$
$$4x + 3y - 2z = 0;$$
$$5x - 3y + 7z = 0;$$

é $S=\{(0, 0, 0)\}$. Essa solução chama-se solução trivial do sistema homogêneo. Se o sistema homogêneo admitir outra solução em que as incógnitas não são todas nulas, a solução será chamada solução não trivial.

4.4
SISTEMAS LINEARES EQUIVALENTES

Dois sistemas lineares que admitem o mesmo conjunto solução são ditos equivalentes.

Exemplo:

Verifique se os sistemas a seguir são equivalentes.

$$x - y = 1;$$
$$2x + y = 5;$$
e
$$3x - 4y = 2;$$
$$x + 2y = 4;$$

Resolução:

Resolvendo o primeiro sistema: Isolando na primeira equação $x = 1 + y$ e substituindo na segunda equação, tem-se:

$$2(1 + y) + y = 5$$
$$2 + 2y + y = 5$$
$$3y = 5 - 2$$
$$3y = 3$$
$$y = 3/3$$
$$y = 1$$

Substituindo o valor de y na equação $x = 1 + y$, obtém-se $x = 2$. Logo, a solução do primeiro sistema é S={(2, 1)}.

Resolvendo o segundo sistema: Isolando x na segunda equação $x = 4 - 2y$ e substituindo na primeira equação, tem-se:

$$3(4 - 2y) - 4y = 2$$
$$12 - 6y - 4y = 2$$
$$-10y = 2 - 12$$
$$-10y = -10$$
$$y = -10/(-10)$$
$$y=1$$

Substituindo o valor de y na equação $x = 4 - 2(y)$, obtém-se $x = 4 - 2$, ou seja, $x = 2$. Portanto, a solução do segundo sistema linear é S={(2, 1)}.

E, como os dois sistemas apresentam a mesma solução, eles são ditos sistemas lineares equivalentes.

4.5
CLASSIFICAÇÃO DE UM SISTEMA LINEAR

Os sistemas lineares são classificados quanto ao número de soluções. A seguir, apresentamos um resumo relacionado à classificação de um sistema de equações lineares:

- *Sistema Linear Possível*: quando admite solução. Neste caso, o sistema pode ainda ser classificado como sendo Determinado (quando admite uma única solução; sigla de classificação: SPD) ou Indeterminado (quando admite infinitas soluções; sigla de classificação: SPI).

- *Sistema Linear Impossível*: quando não admite solução (sigla de classificação: SI).

Exemplos:

1) Um caminhão pode levar, no máximo, 60 caixas de livros do tipo L ou P, de mesmo tamanho. Elas têm, respectivamente, 35 kg e 65 kg. A carga máxima para esse caminhão é de 3 toneladas em cada viagem.

a) Descreva as equações que ilustram o problema, estando o caminhão com a capacidade máxima ocupada.

b) Estando o caminhão com a capacidade máxima ocupada, quantas caixas de cada tipo são transportadas por esse caminhão?

Resolução:

a) Denotando por x o número de caixas de livros do tipo L e por y o número de caixas de livros do tipo P, escrevemos, de acordo com os dados do problema, duas equações lineares:

$$x + y = 60;$$
$$35x + 65y = 3000;$$

b) Vamos resolver esse sistema usando o método da substituição. Isolando x na primeira equação do sistema ($x = 60 - y$) e substituindo na segunda equação, obtemos:

$$35(60 - y) + 65y = 3000$$
$$2100 - 35y + 65y = 3000$$
$$30y = 3000 - 2100$$
$$30y = 900$$
$$y = 900/30$$
$$y = 30.$$

Substituindo $y = 30$ na primeira equação:

$$x + 30 = 60$$
$$x = 30.$$

Logo, são transportadas 30 caixas do tipo *L* e 30 caixas do tipo *P*. Como esse sistema tem uma única solução, isto é, $S=\{(30, 30)\}$, ele é classificado como sendo um sistema possível e determinado (SPD).

2) Resolva o sistema linear de duas equações e duas incógnitas:

$$3x - y = 6;$$
$$6x - 2y = 12;$$

Resolução:
Isolando y na primeira equação $(y = 3x – 6)$ e substituindo na segunda equação, temos:

$$6x - 2(3x - 6) = 12$$
$$6x - 6x + 12 = 12$$
$$0x = 12 - 12$$
$$0x = 0$$

Observe que qualquer número real colocado no lugar de x torna a sentença verdadeira. Isto significa que o sistema apresenta infinitas soluções, ou seja, trata-se de um sistema possível e indeterminado (SPI). Cada uma das infinitas soluções é um par ordenado, cujo primeiro elemento é um número real qualquer e cujo segundo elemento é o triplo do primeiro menos seis.

3) Resolva o sistema linear de três equações e três incógnitas:
$x + y + z = 14;$
$2x + 3y - 5z = -11;$
$3x - 2y + 38z = 9;$

Resolução:
Numerando as equações do sistema linear dado:
$x + y + z = 14;$ *(equação 1)*
$2x + 3y - 5z = -11;$ *(equação 2)*
$3x - 2y + 38z = 9;$ *(equação 3)*

Para resolver este sistema usando o método da substituição, vamos, inicialmente, isolar a incógnita x na equação 1 $(x = 14 - y - z)$ e substituir este resultado na equação 2. Assim:

$2(14 - y - z) + 3y - 5z = -11$

que é equivalente a

$28 - 2y - 2z + 3y - 5z = -11$

que é equivalente a

$y - 7z = -39$ *(equação 4)*

Agora, substituindo $x = 14 - y - z$ na equação 2, obtemos:

$3(14 - y - z) - 2y + 38z = 9$

é equivalente a

$42 - 3y - 3z - 2y + 38z = 9$

é equivalente a

$-5y + 35z = -33$ *(equação 5)*

Resumidamente, temos agora o sistema formado pelas equações 4 e 5:

$y - 7z = -39;$

$-5y + 35z = -33;$

Isolando y na equação 4 $(y = -39 + 7z)$ e substituindo na equação 5, obtemos:

$-5(-39 + 7z) + 35z = -33$

$195 - 35z + 35z = -33$

$0z = -228$

Observe que não há valor real de z que torne a sentença verdadeira (pois um valor real para z, ao ser multiplicado por zero, não resultará em -228). Por isso, dizemos que o sistema é impossível (SI), isto é, o conjunto solução é vazio: $S = \{\}$.

4.6
MATRIZES ASSOCIADAS A UM SISTEMA LINEAR

Considere o sistema linear de *m* equações com *n* incógnitas:

$$a11X1 + a12X2 + \ldots + a1nXn = b1;$$
$$a21X1 + a22X2 + \ldots + a2nXn = b2;$$
$$\ldots$$
$$am1X1 + am2X2 + \ldots + amnXn = bm;$$

em que $a11, a12, \ldots, amn, b1, b2, \ldots, bm$ são números reais.

Desse sistema, destacamos as seguintes matrizes:

Matriz completa do sistema: $[[a11, a12, \ldots, a1n, b1], [a21, a22, \ldots, a2n, b2], \ldots, [am1, am2, \ldots, amn, bm]]$
Matriz incompleta do sistema: $[[a11, a12, \ldots, a1n], [a21, a22, \ldots, a2n], \ldots, [am1, am2, \ldots, amn]]$

Consideremos, ainda, as seguintes matrizes colunas associadas ao sistema: $X = [[X1], [X2], \ldots, [Xm]]$ e $B = [[b1], [b2], \ldots, [bm]]$

Multiplicando-se a matriz incompleta pela matriz das incógnitas, obtemos a matriz dos termos independentes:
$[[a11, a12, \ldots, a1n], [a21, a22, \ldots, a2n], \ldots, [am1, am2, \ldots, amn]]\ [[X1], [X2], \ldots, [Xm]] = [[b1], [b2], \ldots, [bm]]$

Dizemos que essa é a **forma matricial** do sistema linear. Em notação simplificada: $AX=B$.

4.7
REGRA DE *CRAMER*

A regra de *Cramer* é empregada para resolver um sistema linear em que o número de equações é igual ao número de incógnitas. Seja o sistema de três equações e três incógnitas:

$a11X + a12Y + a13Z = b1;$
$a21X + a22Y + a23Z = b2;$
$a31X + a32Y + a33Z = b3;$

Saiba mais: Gabriel Cramer foi matemático e astrônomo suíço. Viveu de 1704 a 1752.

Considere:

- O determinante da matriz incompleta: $Da = [[a11, a12, a13], [a21, a22, a23], [a31, a32, a33]]$

- Os determinantes Dx, Dy e Dz, que se obtêm de Da substituindo, respectivamente, a primeira coluna (dos coeficientes de X), a segunda coluna (dos coeficientes de Y) e a terceira coluna (dos coeficientes de Z) pela coluna dos termos independentes.

$$Dx = [[b1, a12, a13], [b2, a22, a23], [b3, a32, a33]]$$
$$Dy = [[a11, b1, a13], [a21, b2, a23], [a31, b3, a33]]$$
$$Dz = [[a11, a12, b1], [a21, a22, b2], [a31, a32, b3]]$$

Se o determinante da matriz incompleta for não nulo, $Da \mathrel{!}= 0$, então o sistema é possível e determinado e os valores das incógnitas são dados por:

$$X = Dx/Da$$
$$Y = Dy/Da$$
$$Z = Dz/Da$$

De um modo geral, um sistema linear de m equações com incógnitas $X1, X2, X3, \dots , Xm$, cujo determinante Da da matriz incompleta é diferente de zero, é possível e determinado (SPD). O conjunto solução desse sistema é $S = \{(D1/Da, D2/Da, \dots, Dm/Da)\}$, em que Di é o determinante que se obtém de Da substituindo a i-ésima coluna (dos coeficientes de Xi) pela coluna dos termos independentes (GIOVANNI; BONJORNO; GIOVANNI JR., 2002).

Exemplo:
Uma universidade oferta o curso de Licenciatura em Computação. Neste curso,

há 100 acadêmicos matriculados no primeiro e segundo semestres, 62 no segundo e terceiro semestres e 84 no primeiro e terceiro semestres. Qual é o número de acadêmicos matriculados em cada um dos três semestres do referido curso?

Para resolver esse problema, vamos considerar:

x: acadêmicos matriculados no primeiro semestre
y: acadêmicos matriculados no segundo semestre
z: acadêmicos matriculados no terceiro semestre

Temos, assim, o sistema de três equações e três incógnitas:

$$x + y = 100;$$
$$y + z = 62;$$
$$x + z = 84;$$

Resolvendo pela regra de *Cramer,* obtemos:

$$Da = [[1, 1, 0], [0, 1, 1], [1, 0, 1]] = 2$$
$$Dx = [[100, 1, 0], [62, 1, 1], [84, 0, 1]] = 122$$
$$Dy = [[1, 100, 0], [0, 62, 1], [1, 84, 1]] = 78$$
$$Dz = [[1, 1, 100], [0, 1, 62], [1, 0, 84]] = 46$$

Logo:

$$x = Dx/Da = 122/2 =61$$
$$y = Dy/Da = 78/2 = 39$$
$$z = Dz/Da = 46/2 = 23$$

Portanto, há 61 acadêmicos matriculados no primeiro semestre, 39 acadêmicos matriculados no segundo semestre e 29 acadêmicos matriculados no terceiro semestre do curso de Licenciatura em Computação.

Saiba mais: Você pode acompanhar outros exemplos do emprego da Regra de *Cramer* em (OLIVEIRA, 2021b): https://brasilescola.uol.com.br/matematica/regra-cramer.htm

Atividades - Unidade 4

1) Determine m para que (-1, 2, 2) seja solução da equação $mx + y - 2z = 8$.

2) Considere o sistema linear de três equações e três incógnitas:

$$2x + 3y - z = 0;$$
$$x - 2y + 2z = 6;$$
$$x + y - z = 0;$$

a) Verifique se (2, -1, 1) é a solução do sistema dado.
b) Verifique se (0, 0, 0) é a solução do sistema dado.

3) Verifique se os dois sistemas dados a seguir são equivalentes:
$2x - y = 5;$
$2x + y = 7;$
e
$-x + 6y = 10;$
$3x - y = 21;$

4) Classifique os sistemas em SPD (sistema possível e determinado), SPI (sistema possível e indeterminado) ou SI (sistema impossível):
a) $x - 4y = -3;$
$3x + 2y = 5;$

b) $4a + 8b = 4;$
$3a + 6b = 6;$

5) Ache dois números reais cuja soma é 11 e cuja diferença é 31.

6) Expresse matricialmente os sistemas, resolva-os e classifique a solução:
a) $2x - 3y = 5;$
$5x + 5y = 25;$

b) $a + 3b - c = 1;$
$a + b + c = 4;$
$-3a - 3b - 4c = -12;$

c) $5x + 10y = 30;$
$-15x - 30y = -60;$

d) $2a - 3b = 10;$
$4a - 6b = 20;$

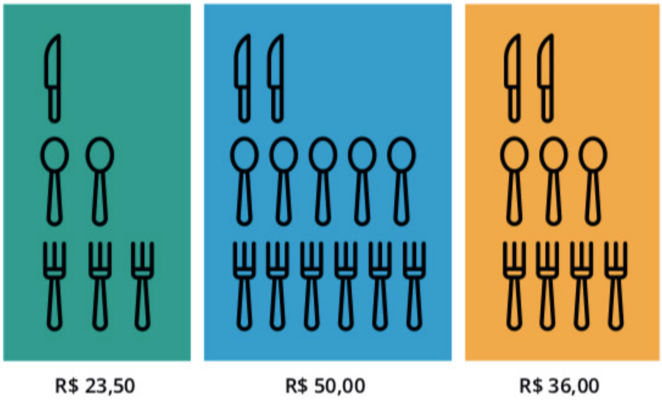
7) Uma empresa deve enlatar uma mistura de amendoim, castanha de caju e castanha-do-pará. Sabe-se que o quilo de amendoim custa R$ 5,00, que o quilo da castanha de caju custa R$ 20,00 e que o quilo de castanha-do-pará custa R$ 16,00. Cada lata deve conter meio quilo da mistura e o custo total dos ingredientes de cada lata deve ser de R$ 5,75. Além disso, a quantidade de castanha de caju em cada lata deve ser igual a um terço da soma das outras duas. Tem-se, então, o seguinte sistema linear:
$x + y + z = 0,5;$
$5x + 20y + 16z = 5,75;$
$y = (x + z)/3;$
Resolver o referido sistema, determinando as quantidades, em gramas, de cada ingrediente por lata.

8) Em torno do prédio de uma prisão, foi construído um fosso, circundado por muros, conforme a planta abaixo, e há uma ponte de largura L para atravessá-lo. Durante a ronda diurna, os guardas da prisão dão uma volta completa no muro externo, atravessam a ponte e dão uma volta completa no muro interno. Esse trajeto pode ser completado em 5.320 metros. Na ronda noturna, os guardas dão duas voltas completas no muro externo, atravessam a ponte e dão uma volta completa no muro interno, completando esse outro trajeto em 8.120 metros. Elabore um sistema linear condizente com este problema e determine a largura L (em metros) da ponte.

Dica: Como se quer descobrir o valor de L e não os perímetros correspondentes aos comprimentos dos muros (interno e externo), pode-se então agrupar tais perímetros como se fossem uma única variável. Assim, será obtido um sistema na forma:
$4x + 9L = 5.320;$
$6x + 17L = 8.120;$

9) Por meio da informação ilustrativa da Figura 16, conclua qual deve ser o valor correspondente a cada faca, colher e garfo.

Figura 16 - Garfos, Facas e Colheres



Fonte: CTE/UFSM.

Descrição de Imagem: Figura 16- Três retângulos, um ao lado do outro, em sentido vertical, sendo que cada um deles apresenta três linhas. O primeiro retângulo, na cor verde, apresenta os seguintes desenhos: na primeira linha, 1 faca; na segunda linha, 2 colheres; e na terceira linha, 3 garfos, sendo que abaixo destes três desenhos (do primeiro retângulo) há a indicação de que o valor a ser pago por esses itens deve ser R$ 23,50. O segundo retângulo, na cor azul, apresenta os seguintes desenhos: na primeira linha, 2 facas; na segunda linha, 5 colheres; e na terceira linha, 6 garfos, sendo que abaixo deste segundo retângulo há uma indicação de que o valor a ser pago por esses itens deve ser R$ 50,00. O terceiro retângulo, na cor amarela, apresenta os seguintes desenhos: na primeira linha, 2 facas; na segunda linha, 3 colheres; e na terceira linha, 4 garfos, com indicação de que o valor a ser pago por eles deve ser R$ 36,00.

10) Em um curso de capacitação, há 107 alunos matriculados na 1ª e 2ª turmas, 74 alunos na 2ª e 3ª turmas e 91 alunos na 1ª e 3ª turmas. Qual o total de alunos desse curso?

11) Um grupo de amigos resolveu fazer uma doação de 120 brinquedos para um orfanato. O referido grupo conseguiu arrecadar um valor de R$ 1.480,00 para efetuar as compras da quantidade total de brinquedos para meninos e meninas. A direção do orfanato sugeriu que fossem comprados carrinhos, bonecas e jogos didáticos, sendo que a quantidade de jogos deve ser igual à soma do número de carrinhos e bonecas. Se cada carrinho custa R$ 8,00, cada boneca custa R$ 12,00 e cada jogo didático custa R$ 14,00, quais são as quantidades a serem compradas de cada brinquedo?

12) Classifique e resolva os sistemas lineares dados pelas equações a seguir:

a) $-x+2y = 4;$
$2x+4y = 8;$

b) $x+y = 4;$
$3x+3y = 3;$

13) Um ourives cobrou R$ 150,00 para cunhar medalhas de ouro com 3 gramas cada uma, de prata com 5 gramas cada e de bronze com 7 gramas cada uma, ao preço unitário de R$ 30,00, R$ 10,00 e R$ 5,00, respectivamente. Sabendo que foram confeccionadas 15 medalhas, com massa total de 87 gramas, determine o número de medalhas de ouro, prata e bronze confeccionadas pelo ourives.

Respostas das Atividades

1) $m = -10$

2) a) sim b) não

3) Os sistemas não são equivalentes, pois não apresentam o mesmo conjunto solução. O primeiro sistema apresenta solução *S={(3, 1)}* e o segundo sistema apresenta solução *S={(8, 3)}*.

4) a) SPD, *S={(1, 1)}*
b) SI, *S={ }*.

5) *S={(21, -10)}*.

6) a) *[[2, -3], [5, 5]] [[x], [y]]=[[5], [25]], S = {(4, 1)}*, SPD.
b) *[[1, 3, -1], [1, 1, 1], [-3, -3, -4]] [[a], [b], [c]]=[[1], [4], [-12]], S={(11/2, -3/2, 0)}*, SPD.
c) *[[5, 10], [-15, -30][[x], [y]]=[[30], [-60]], S={ }, SI.*
d) *[[2, -3], [4, -6]] [[a], [b]]=[[10], [20]]*, SPI (apresenta infinitas soluções).

7) A empresa deve enlatar uma mistura de 250 gramas de amendoim, 125 gramas de castanha de caju e 125 gramas de castanha-do-pará.

8) *L= 40* metros.

9) O preço da faca é R$ 5,50, o preço da colher é R$ 3,00 e o preço do garfo é R$ 4,00.

10) Serão 62 + 45 + 29 = 136 alunos matriculados.

11) Deverão ser comprados 20 carrinhos, 40 bonecas e 60 jogos didáticos.
12) a*) S = {(0, 2)}*, SPD.

b) *S={ }, SI.*
13) S = {2 ouro, 5 prata, 8 bronze}

Considerações finais

Este *e-book* apresentou conceitos, exemplos e atividades práticas ligadas ao estudo introdutório da Álgebra Linear, abordando conteúdos relacionados a matrizes, determinantes e sistemas lineares. Como diferencial, a proposta compreendeu o desenvolvimento de exemplos práticos, utilizando a linguagem de programação *Python*, além da descrição das figuras e tabelas, visando a atender pessoas com deficiência visual.

Os exemplos desenvolvidos na linguagem de programação *Python* foram executados e testados no *IDLE Editor*, a fim de verificar se o resultado do código-fonte era correspondente às atividades propostas.

Sendo assim, esperamos que este *e-book* seja útil nos processos de ensino e de aprendizagem de Álgebra Linear, em especial para cursos da área de Computação.

Referências

ANTON, H.; RORRES, C. **Álgebra Linear com Aplicações**. 8. ed. São Paulo: Bookman, 2001.

BOLDRINI, J. L.; COSTA, S. I. R.; FIGUEIREDO, V. L.; WETZLER, H. G. **Álgebra Linear**. 3. ed. São Paulo, Harbra, 1980.

BRASIL. Ministério da Educação. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP). Exame Nacional do Ensino Médio (ENEM): Prova de redação e de linguagens, códigos e suas tecnologias – Prova de matemática e suas tecnologias. **Provas e gabaritos**, Brasília, 2012. Disponível em: <https://download.inep.gov.br/educacao_basica/enem/provas/2012/caderno_enem2012_dom_amarelo.pdf>. Acesso em: 09 set. 2021.

CALLIOLI, C. A; DOMINGUES, H. H.; COSTA, R.C.F. **Álgebra Linear e Aplicações**. 7. ed. São Paulo: Atual, 2006.

CAMPAGNER, C. A. Determinante – número representa matriz. **UOL**, São Paulo, 2021. Disponível em: <https://educacao.uol.com.br/disciplinas/matematica/determinante-numero-representa-matriz.htm>. Acesso em: 10 set. 2021.

CANDIDO, L. Geometria Analítica e Álgebra Linear. **Páginas pessoais**, UTFPR, [Campo Mourão], [2018?]. Disponível em: <http://paginapessoal.utfpr.edu.br/lcandido/ensino/geometria-analitica-e-algebra-linear/matrizes-e-determinantes/notas-de-aula/matrizes.pdf>. Acesso em: 10 set. 2021.

FORTES, P. R. et al. **Matemática II** [recurso eletrônico]. 1. ed. Santa Maria, RS: UFSM/ NTE/ UAB, 2019. Disponível em: <https://www.ufsm.br/app/uploads/sites/358/2019/07/livro-matematica-final.pdf>. Acesso em: 09 set. 2021.

FREITAS, D. B.; SOUZA, K. C. **Plano de Trabalho sobre Matrizes e Determinantes**, [Rio de Janeiro], 2012. Disponível em: <https://docplayer.com.br/105257277-Plano-de-trabalho-sobre-matrizes-e-determinantes.html>. Acesso em: 10 set. 2021.

GIOVANNI, J. R.; BONJORNO, J. R.; GIOVANNI JR, J. R. **Matemática Fundamental**: Uma nova abordagem. São Paulo: FTD, 2002.

NOVAES, J. C. Sistemas Lineares: regra de Cramer e escalonamento. **Matemática Básica**, [S.l.], 2021. Disponível em: <https://matematicabasica.net/sistemas-lineares>. Acesso em: 22 nov. 2021.

MENEZES, P. B. **Matemática Discreta para Computação e Informática**. Porto Alegre: Bookman, 2010. Série Livros Didáticos – Informática UFRGS.

NUMPY. **User guide**. [S.l.], 2021. Disponível em: <https://numpy.org/devdocs/user/building.html>. Acesso em: 28 abr. 2021.

OLIVEIRA, R. R. de. Sistemas lineares. **Brasil Escola**, São Paulo, 2021a. Disponível em: <https://brasilescola.uol.com.br/matematica/sistemas-lineares.htm>. Acesso em: 8 nov. 2021.

OLIVEIRA, R. R. de. Regra de Cramer. **Brasil Escola**, São Paulo, 2021b. Disponível em: <https://brasilescola.uol.com.br/matematica/regra-cramer.htm>. Acesso em: 18 nov. 2021.

OLIVEIRA, G. A. de. Teorema de Laplace. **Brasil Escola**, São Paulo, 2021. Disponível em: <https://brasilescola.uol.com.br/matematica/teorema-laplace.htm>. Acesso em: 16 nov. de 2021.

PARREIRA, F. J.; SILVEIRA, S. R.; BERTOLINI, C.; SEVERO, R. **Introdução a Algoritmos**. Santa Maria: UAB/NTE/UFSM, 2017. Disponível em: <https://repositorio.ufsm.br/bitstream/handle/1/15820/Licenciatura_Computacao_introducaoalgoritmos.pdf?sequence=1&isAllowed=y>. Acesso em: 21 out. 2019.

PYSCIENCE-BRASIL. **Como instalar o NumPy**, [S.l.], [20--]. Disponível em: <http://pyscience-brasil.wikidot.com/install:como-instalar-o-numpy>. Acesso em: 28 abr. 2021.

PYTHON. **Welcome to Python.org**, [S.l.], 2021. Disponível em: <http://python.org>. Acesso em: 21 out. 2019.

POOLE, D. **Álgebra Linear**. São Paulo: Pioneira Thomson Learning, 2004.

RIBEIRO, A. G. Regra de Sarrus. **Brasil Escola**, São Paulo, 2021. Disponível em: <https://brasilescola.uol.com.br/matematica/regra-sarrus.htm>. Acesso em: 12 nov. 2021.

ROSSUM, G. Tutorial Python. **Python Brasil**, [S.l.], 2010. Disponível em: <https://wiki.python.org.br/Tutorial_Python>. Acesso em: 22 nov. 2021.

SANTOS, N. M. **Vetores e Matrizes**: uma introdução à álgebra linear. São Paulo: Thomson, 2007.

SHOKEEN, M. Módulos Matemáticos em Python: Math e CMath. **Envatotuts+**, [S.l.], 2017. Disponível em: <https://code.tutsplus.com/pt/tutorials/mathematical-modules-in-python-math-and-cmath--cms-26913>. Acesso em: 22 nov. 2021.

SILVA, M. N. P. da. Raiz de uma Equação do 2º Grau. **Brasil Escola**, São Paulo, 2021. Disponível em: <https://brasilescola.uol.com.br/matematica/raiz-uma-equacao-2-grau-1.htm>. Acesso em: 8 nov. de 2021.

STACKABUSE. Solving Systems of Linear Equations with Python's Numpy. **Articles**, [S.l.], 2020. Disponível em: <https://stackabuse.com/solving-systems-of--linear-equations-with-pythons-numpy/>. Acesso em: 28 abr. 2021.

STEINBRUCH, A.; WINTERLE, P. **Álgebra Linear**. São Paulo: Pearson Makron Books, 2009.

UNIVERSIDADE FEDERAL DE SANTA MARIA. Comissão Permanente de Vestibular (COPERVES). Processo Seletivo Seriado 2 – Vestibular 2011. **Concursos Anteriores**, Santa Maria, 2011. Disponível em: <https://www.coperves.ufsm.br/concursos/vestibular_2011/arquivos/Prova_PS2.pdf>. Acesso em: 26 nov. 2021.

WIKILIVROS. **Matemática elementar/Sistemas lineares**, [S.l.], 2018. Disponível em: <https://pt.wikibooks.org/wiki/Matem%C3%A1tica_elementar/Sistemas_lineares>. Acesso em: 22 nov. 2021.

Apêndice

1) Código-fonte dos exemplos da Figura 7, Unidade 1:

```
valor=10
Valor=15
print(valor)
print(Valor)
print(Valor+valor)
nome="José"
print(nome)
sobrenome="Silva"
print(nome+sobrenome)
```

2) Código-fonte dos exemplos da Figura 8, Unidade 1:

```
2+2
3-4
5*6
2.0*5
3.0*-2
10/3
print(2+2)
print(2.0*5)
print("Resultado:",2*4+5)
print("Resultado:"2*(4+5))
print("Soma:",2+2);print("Multiplicação:",2*2)
```

3) Código-fonte dos exemplos da Figura 9, Unidade 1:

```
#Exemplo de seleção simples
a=5
if a>=5:
   print("o valor de a é maior ou igual a 5")
#Exemplo de seleção composta
a=3
if a>=5:
    print("o valor de a é maior ou igual a 5")
else:
       print("o valor de a é menor que 5")
#Exemplo de seleção de múltipla escolha
a=3
b=5
if a>b:
      print("o valor de a é maior que o valor de b")
```

```
elif a<b:
      print("o valor de a é menor que o valor de b")
elif a>=b:
      print("o valor de a é maior ou igual ao valor de b")
elif a<=b:
      print("o valor de a é menor ou igual ao valor de b")
```

4) Código-fonte dos exemplos da Figura 10, Unidade 1:

```
nomes=['Ana','João','José','Maria']
for n in nomes:
       print(n)
nomes=['Ana','João','José','Maria']
for n in nomes:
      print(n)
else:
      print('A lista de nomes foi impressa com sucesso!')
for caracteres in 'Python':
       print('Letra:', caracteres)
```

5) Código-fonte dos exemplos da Figura 11, Unidade 1:

```
str(12)
a=3
str(a)
soma=a+5
print(soma)
a=str(a)
print(a)
soma+5
```

6) Código-fonte do arquivo *calculadora.py*, Unidade 1:

```
#Calculadora Básica
num1=int(input("Digite um número:"))
num2=int(input("Digite outro número:"))
operacao=input("Escolha a operação: + - * / :")

if (operacao=="+"):
     print("Soma:",num1 + num2)
if (operacao=="-"):
     print("Subtração:",num1 - num2)
if (operacao=="*"):
     print("Multiplicação:",num1 * num2)
if (operacao=="/"):
     print("Divisão:",num1 / num2)
```

7) Código-fonte do arquivo *matriz1.py*, Unidade 2:

```
A = [[6.2, 7.1, 6.8, 8.0], [5.7, 6.4, 7.0, 7.3], [7.2, 8.2, 6.9, 7.8]]
print("Dados armazenados na matriz A, de acordo com a linha:")
for linha in range(3):
      print("Linha da Matriz:",linha+1)
      print(A[linha])
print("Dados armazenados na matriz A:")
for linha in range(3):
     for coluna in range(4):
           print("Linha:", linha+1, "Coluna:", coluna+1, "Valor armazenado:",
  A[linha][coluna])
```

8) Código-fonte do arquivo *matriz2.py*, Unidade 2:

```
E = [[6.2, 7.1, 6.8], [5.7, 6.4, 7.0], [7.2, 8.2, 6.9]]
print("Valores da diagonal principal:")
for linha in range(3):
      for coluna in range(3):
            if linha==coluna:
                print("Linha:",linha+1,"Coluna:",coluna+1,"Valor
armazenado:",E[linha][coluna])
print("Valores da diagonal secundária:")
for linha in range(3):
      for coluna in range(3):
            if linha+coluna==3-1:
                print("Linha:",linha+1,"Coluna:",coluna+1,"Valor
armazenado:",E[linha][coluna])
```

9) Código-fonte do arquivo *matriz3.py*, Unidade 2:

```
D = [[3, 0, 0], [0, -1, 0], [0, 0, 7]]
matrizdiagonal=0
for linha in range(3):
     for coluna in range(3):
           if linha!=coluna:
                if D[linha][coluna]!=0:
                      matrizdiagonal=1
if matrizdiagonal==0:
      print("A matriz D é uma matriz diagonal")
else:
      print("A matriz D não é uma matriz diagonal")
```

10) Código-fonte do arquivo *matriz4.py*, Unidade 2:

```
a matriz A tem 3 linhas e 2 colunas
A = [[2, 5], [0, 9], [-2, 7]]
# a matriz B, onde guardaremos a matriz transposta de A, terá 2 linhas e 3
colunas
# criamos a matriz B com os valores nulos (zeros)
B = [[0,0,0],[0,0,0]]
# Impressão dos dados da matriz A
print("Valores da Matriz A:")
for linha in range(3):
    for coluna in range(2):
        print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",A[linha]
[coluna])
# Transposição da matriz A para B
for linha in range(3):
    for coluna in range(2):
        B[coluna][linha]=A[linha][coluna]
# Impressão dos dados da matriz B
print("Valores da Matriz B:")
for linha in range(2):
    for coluna in range(3):
        print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",B[linha]
[coluna])
```

11) Código-fonte do arquivo *matriz5.py*, Unidade 2:

```
A = [[3, 9], [1, 2], [0, 5]]
B = [[3, 9], [1, 2], [0, 5]]
matrizesiguais=0
for linha in range(3):
    for coluna in range(2):
        if A[linha][coluna]!=B[linha][coluna]:
            matrizesiguais=1
if matrizesiguais==0:
    print("As matrizes A e B são iguais")
else:
    print("As matrizes A e B não são iguais")
```

12) Código-fonte do arquivo *matriz6.py*, Unidade 2:

```
A = [[1, 8], [0, 2], [5, -1]]
B = [[8, -2], [2, 4], [3, 5]]
C = [[0,0],[0,0],[0,0]]

# Impressão dos valores da matriz A
```

```
print("Valores armazenados na Matriz A:")
for linha in range(3):
    for coluna in range(2):
        print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",A[linha]
[coluna])
# Impressão dos valores da matriz B
print("Valores armazenados na Matriz B:")
for linha in range(3):
    for coluna in range(2):
        print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",B[linha]
[coluna])
# Soma dos valores das matrizes A e B e atribuição à matriz C
for linha in range(3):
    for coluna in range(2):
        C[linha][coluna]=A[linha][coluna]+B[linha][coluna]
# Impressão dos valores da matriz C
print("Valores armazenados na Matriz C:")
for linha in range(3):
    for coluna in range(2):
        print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",C[linha]
[coluna])
```

13) Código-fonte do arquivo *matriz7.py*, Unidade 2:

```
A = [[4, 0], [2, 7]]
B = [[1, 0, 2], [7, 5, 3]]
C = [[0,0,0],[0,0,0]]
# Impressão da matriz A
print("Valores da Matriz A:")
for linha in range(2):
    for coluna in range(2):
        print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",A[linha]
[coluna])
# Impressão da matriz B
print("Valores da Matriz B:")
for linha in range(2):
    for coluna in range(3):
        print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",B[linha]
[coluna])
# for linha de acordo com o número de linhas da matriz A (duas linhas)
# for coluna de acordo com o número de colunas da matriz B (três colunas)
# for k de acordo com o número de colunas da matriz A e de linhas da matriz B
for linha in range(2):
    for coluna in range(3):
        for k in range(2):
            C[linha][coluna]=C[linha][coluna]+A[linha][k]*B[k][coluna]
```

```
# Impressão da matriz C
print("Valores da matriz C - multiplicação da matriz A pela matriz B:")
for linha in range(2):
      for coluna in range(3):
       print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",C[linha]
[coluna])
```

14) Código-fonte do arquivo *matriz8.py*, Unidade 2:

```
A = [[1, 3], [2, 4]]
B = [[-2, 3/2], [1, -1/2]]
C = [[0,0],[0,0]]
# Impressão da matriz A
print("Valores da Matriz A:")
for linha in range(2):
     for coluna in range(2):
            print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",A[linha]
[coluna])
# Impressão da matriz B
print("Valores da Matriz B:")
for linha in range(2):
     for coluna in range(2):
            print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",B[linha]
[coluna])
# for linha de acordo com o número de linhas da matriz A (duas linhas)
# for coluna de acordo com o número de colunas da matriz B (três colunas)
# for k de acordo com o número de colunas da matriz A e de linhas da matriz B
for linha in range(2):
      for coluna in range(2):
             for k in range(2):
                    C[linha][coluna]=C[linha][coluna]+A[linha][k]*B[k][coluna]
# Impressão da matriz C
#Verificação da matriz C - se é matriz identidade
matrizidentidade=0
print("Valores da matriz C - multiplicação da matriz A pela matriz B:")
for linha in range(2):
     for coluna in range(2):
            print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",C[linha]
[coluna])
            if linha==coluna:
                 if C[linha][coluna]!=1:
                      matrizidentidade=1
            else:
                  if C[linha][coluna]!=0:
                        matrizidentidade=1
if matrizidentidade==0:
```

```
        print("A matriz C é uma matriz identidade, então as matrizes A e B são
inversas uma da outra")
else:
        print("A matriz C não é uma matriz identidade, então as matrizes A e B
não são inversas uma da outra")
```

15) Código-fonte do Exercício 23, Unidade 2:

```
Pesticidas=[[2,3,4,3],[3,2,2,5],[4,1,6,4]]
Herbivoros=[[20,12,8],[28,15,15],[30,12,10],[40,16,20]]
QtdePesticidaHerbivoro=[[0,0,0],[0,0,0],[0,0,0]]
# Impressão da matriz/tabela Pesticidas
print("Valores da Tabela de Pesticidas")
for linha in range(3):
     for coluna in range(4):
          print("Linha:",linha+1,"Coluna:",coluna+1,"Valor
armazenado:",Pesticidas[linha][coluna])
# Impressão da matriz/tabela herbivoros
print("Valores da Tabela de Herbívoros:")
for linha in range(4):
     for coluna in range(3):
          print("Linha:",linha+1,"Coluna:",coluna+1,"Valor
armazenado:",Herbivoros[linha][coluna])
# for linha de acordo com o número de linhas da matriz/tabela Pesticidas (três
linhas)
# for coluna de acordo com o número de colunas da matriz/tabela herbívoros
B (três colunas)
# for k de acordo com o número de colunas da matriz/tabela Pesticidas e de
linhas da matriz/tabela herbivoros
for linha in range(3):
     for coluna in range(3):
          for k in range(4):
               QtdePesticidaHerbivoro[linha][coluna]=QtdePesticidaHerbivoro[l
inha][coluna]+
               Pesticidas[linha][k]*Herbivoros[k][coluna]
# Impressão da matriz/tabela com as quantidades de pesticida por herbívoros
print("Valores da Tabela de Quantidades de Pesticida por Herbívoros:")
for linha in range(3):
     for coluna in range(3):
          print("Linha:",linha+1,"Coluna:",coluna+1,"Valor armazenado:",Qtde
PesticidaHerbivoro[linha][coluna])
```

16) Código-fonte do arquivo *determinante1.py*, Unidade 3:

```
# inicialização das variáveis
A=[[3,4],[5,6]]
```

```
produtodiagonalprincipal=1
produtodiagonalsecundaria=1
determinante=0
# Impressão da matriz A
print("Valores armazenados na matriz A:")
for i in range(2):
    for j in range(2):
        print("Linha:",i+1,"Coluna:",j+1,"Valor armazenado:",A[i][j])
# Cálculo do produto dos elementos da diagonal principal e da diagonal secundária
for i in range(2):
    for j in range(2):
        if i==j:
            produtodiagonalprincipal=produtodiagonalprincipal*A[i][j]
        else:
            produtodiagonalsecundaria=produtodiagonalsecundaria*A[i][j]
# Cálculo do determinante
determinante=produtodiagonalprincipal-produtodiagonalsecundaria
print("O determinante da matriz A é:",determinante)
```

17) Código-fonte do arquivo *determinante2.py*, Unidade 3:

```
# Inicialização da Matriz A (3 linhas e 3 colunas)
A = [[3, 9, 7], [-2, 4, 5], [-1, 8, 10]]
# Inicialização da Matriz ampliada - Matriz B
# 3 linhas e 5 colunas - copiando as duas primeiras colunas de A à direita de B
B = [[3, 9, 7, 3, 9],[-2, 4, 5, -2, 4],[-1, 8, 10, -1, 8]]
# o determinante deve ser calculado de acordo com a fórmula
# det(B) = (B11)(B22)(B33) + (B12)(B23)(B31) + (B13)(B21)(B32) - (B31)(B22)(B13)-(B32)(B23)(B11)- (B33)(B21)(B12)
# como os índices em Python iniciam em zero, o elemento B11 é, na verdade, B00, B22 é B11 e assim por diante
# Cálculo do determinante
det=((B[0][0]*B[1][1]*B[2][2])+(B[0][1]*B[1][2]*B[2][3])+(B[0][2]*B[1][3]*B[2][4]))-\
((B[2][0]*B[1][1]*B[0][2]))-((B[2][1]*B[1][2]*B[0][3]))-((B[2][2]*B[1][3]*B[0][4]))
print("O determinante da matriz A é:",det)
```

18) Código-fonte do arquivo *determinante3.py*, Unidade 3:

```
M = [[0, 9, 7], [0, 4, 5], [0, 8, 10]]
# Impressão da matriz M
print("Valores da Matriz M:")
for linha in range(3):
```

```
    for coluna in range(3):
            print("Linha:",linha+1,"Coluna:",coluna+1,"Valor
armazenado:",M[linha][coluna])
# Verificação de valores nulos
# percorrer as linhas da matriz - se pelo menos uma linha tiver todos os valores
nulos
# então o determinante da matriz será zero
det=0
matrizdetzero=0
for linha in range(3):
    for coluna in range(3):
        if M[linha][coluna]!=0:
            det=1
    if det==0:
        print("A matriz M tem determinante igual a zero!")
        matrizdetzero=1
        break
    det=0
# vamos percorrer as colunas da matriz M se ainda não foram encontradas
linhas com valores todos nulos
if matrizdetzero==0:
    det=0
    for coluna in range(3):
        for linha in range(3):
            if M[linha][coluna]!=0:
                det=1
        if det==0:
            print("A matriz M tem determinante igual a zero!")
            matrizdetzero=1
            break
        det=0
if matrizdetzero==0:
    print("A matriz M não tem determinante igual a zero!")
```

19) Código-fonte do arquivo *linear1.py*, Unidade 4:

```
for y in range(100):
    if 32-6*y+5*y==29:
        print("O valor de y é:",y)
        break
x=(16-3*y)/2
print("O valor de x é:",x)
```

20) Código-fonte do arquivo *linear2.py*, Unidade 4

```
import numpy as np
lista=[[2, 3],[4, 5]]
A=np.array(lista)
inv_A=np.linalg.inv(A)
print("Matriz inversa de A:")
print(inv_A)
B=np.array([16,29])
X=np.linalg.inv(A).dot(B)
print("Valores das incógnitas X e Y:")
print(X)
```

Apresentação dos autores

Mariza Camargo

Professora Associada do Departamento de Engenharia e Tecnologia Ambiental da UFSM (Universidade Federal de Santa Maria) – Campus Frederico Westphalen/RS. Doutora em Engenharia Mecânica e Mestre em Matemática Aplicada pela UFRGS (Universidade Federal do Rio Grande do Sul) e Licenciada em Matemática pela UFSM. mariza@ufsm.br

Patricia Rodrigues Fortes

Professora Associada do Departamento de Engenharia e Tecnologia Ambiental da UFSM (Universidade Federal de Santa Maria) – Campus Frederico Westphalen/RS. Doutora em Engenharia Mecânica e Mestre em Matemática Aplicada pela UFRGS (Universidade Federal do Rio Grande do Sul) e Licenciada em Matemática pela UFSM. patricia@ufsm.br

Cristiano Bertolini

Professor Adjunto do Departamento de Tecnologia da Informação da UFSM (Universidade Federal de Santa Maria) – Campus Frederico Westphalen/RS. Doutor em Ciência da Computação pela UFPE (Universidade Federal de Pernambuco). Mestre em Ciência da Computação pela PUC-RS (Pontifícia Universidade Católica do Rio Grande do Sul). Bacharel em Ciência da Computação pela UPF (Universidade de Passo Fundo). cristiano.bertolini@ufsm.br

Sidnei Renato Silveira

Professor Associado do Departamento de Tecnologia da Informação da UFSM (Universidade Federal de Santa Maria) – Campus Frederico Westphalen/RS. Doutor e Mestre em Ciência da Computação pela UFRGS (Universidade Federal do Rio Grande do Sul). Especialista em Gestão Educacional pelo SENAC (Serviço Nacional de Aprendizagem Comercial). Especialista em Administração e Planejamento para Docentes e Bacharel em Informática pela ULBRA (Universidade Luterana do Brasil). sidnei.silveira@ufsm.br

Guilherme Bernardino da Cunha

Professor Associado do Departamento de Tecnologia da Informação da UFSM (Universidade Federal de Santa Maria) – Campus Frederico Westphalen/RS. Mestre e Doutor em Engenharia Elétrica pela UFU (Universidade Federal de Uberlândia). Especialista em Produção de Material Didático EaD pela UFAM (Universidade Federal do Amazonas). Bacharel em Ciência da Computação pelo UNITRI (Centro Universitário do Triângulo). guilherme@ufsm.br