Aula 04–Gabarito dos Exercícios

- Atividade verificadora de aprendizagem:

```
'''
Aula04 - Atividade
Programa completo que cadastre alunos:
(A) Manipulações de Strings:
(i) tratar entradas vazias
(ii) tratar nomes de alunos que contém números
(iii) tratar número de matrículas com caracteres.
(B) Tratamento de exceções:
(i) Tratar fechamento de arquivos que não foram abertos
(ii) tentar escrever em arquivos com permissão para somente leitura
(iii) tentar ler dados de arquivos inexistentes.
'''
#Função para exibir as opções de cadastro
def entradaUsuario():
globa lacao #variável global
print('''1: Cadastrar novo aluno.
2: Listar alunos cadastrados.
3: Procurar aluno pelo nome.
4: SAIR.''')
acao = int(input('O que deseja fazer ? '))

#Função para inserir um novo aluno
def cadastraAluno():
print('Cadastro de aluno. Preencha as informações:')
   # tratamento de strings
ok = 0 #variável sinalizadora (flag)
while(ok == 0):
matricula = input('Matrícula: ')
nome = input('Nome: ')
email = input('E-mail: ')
curso = input('Curso: ')
          # (i) tratar entradas vazias
if (not matricula ornot nome ornotemailornot curso):
print("Entrada vazia ! Redigite.\n")
else:
             #(ii) tratar nomes de alunos que contém números
if (nome.isdigit()):
print("Somente caracteres no nome !!! Redigite.\n")
else:
                #(iii) tratar número de matrículas com caracteres.
if (notmatricula.isdigit()):
print("Somente números na matrícula !!! Redigite.\n")
else:
ok = 1 #tudo certo
else:
aluno = f'{matricula},{nome}, {email}, {curso}\n' #formato
       #(i) Tratar fechamento de arquivos que não foram abertos
with open('alunos.txt', 'a', encoding='UTF-8') as arquivo:
arquivo.write(aluno) #gravar aluno adicionando ao arquivo
print('Cadastrado!!!\n')

#Função para listar os dados dos alunos
def listaAluno():
   #tratamento de exceções
try:
       #(ii) tentar escrever em arquivos com permissão para somente leitura
with open('alunos.txt', 'r', encoding='utf-8') as arquivo:
listaAlunos = arquivo.read().split('\n') #ler o registro
print('\nLista de alunos cadastrados:')
```

```
print('Matrícula,Nome, Email, Curso')
for aluno in listaAlunos:
print(aluno)
   #(iii) tentar ler dados de arquivos inexistentes.
except FileNotFoundError:
print('Arquivo não existe. Cadastre primeiro !!!\n')

#Função para buscar um aluno pelo nome
def procuraAluno():
   #tratamento de exceções
try:
print('Buscar aluno por nome:')
busca = input('Nome: ')
if (not busca):
print ('Entrada vazia !!!')
else:
with open('alunos.txt', 'r', encoding='utf-8') as arquivo:
listaAlunos = arquivo.read().split('\n') #separa os dados
resultado = None
for aluno in listaAlunos:
if (aluno):
                #divide com , e remove espaço no fim
nomeAluno = aluno.split(',')[1].rstrip()
if busca == nomeAluno:
resultado = aluno #encontrou
break
if resultado == None: #não encontrou
print('\nNão foi encontrado nenhum aluno com esse nome.\n')
else:
print(resultado+'\n')
except FileNotFoundError:
print('Arquivo não existe. Cadastre primeiro !!!\n')

#programa principal
#repete até encerrar
while True:
entradaUsuario()

if acao == 1:
cadastraAluno()
elif acao == 2:
listaAluno()
elif acao == 3:
procuraAluno()
elif acao == 4:
break
else:
print('\n::::: Escolha uma das quatro opções! :::::\n')
#Fim do programa
```

- Atividade Autônoma Aura:

  Questão 1)A interpolação de String é uma característica bastante útil quando queremos fazer uma substituição dentro de um texto. O Python trata a interpolação de Strings de modo a facilitar o trabalho do desenvolvedor.

  A respeito da interpolação de Strings no Python 3, selecione a opção correta:

  a) Pode ser feita com @ (arroba), ou com $ (Cifrão)

  b) Podemos aplicar usar$ (Cifrão), ou # (Hashtag)

  c) O modo correto é usando # (Hashtag), ou % (percentual)

d) A forma é usando % (percentual), ou { } (chaves)

e) Podemos utilizar { } (chaves), ou " (Aspas duplas)

Gabarito: A alternativa CORRETA é a letra “D”. O Python suporta várias maneiras de fazer interpolação de strings de texto, entre elas estão %-formatação, sys.format(), string.Template e f-strings.

Tópico de aprendizagem: FUNÇÕES DE MANIPULAÇÃO DE STRINGS.

Referência: Conteúdo digital da disciplina, Tema "Manipulação de Dados", Módulo 2 - "funções de manipulação de strings" e Módulo 2 - "Exceções na manipulação de arquivos e outras operações".

Questão 2) As operações com arquivos são fundamentais na programação, pois os arquivos são um recurso muito útil para manter a persistência de dados. O Python oferece comandos que permitem que os desenvolvedores possam fazer essas manipulações.

A respeito da manipulação de arquivos no Python, selecione a opção CORRETA.

a) Para que um programa que manipula arquivos funcione é necessário fazer o tratamento de exceções.

b) Apesar de não ser obrigatório o tratamento de exceções, ele deve ser usado sempre para evitar problemas ao longo da execução do programa.

c) Por se tratar de um recurso que já faz parte do Python padrão, o tratamento de exceções não é necessário.

d) Para abrir um arquivo texto, por exemplo, sempre é necessário verificar se ele não está corrompido antes, pois, caso contrário, pode gerar inconsistência no programa.

e) Os arquivos podem gerar muitos problemas, portanto são um recurso de uso apenas didático. Aplicações reais são desenvolvidas com bancos de dados

Gabarito: A alternativa CORRETA é a letra “B”. O tratamento de exceções no Python não é obrigatório, porém se trata de uma boa prática de programação que reduz as chances do programa ter um comportamento inconsistente, caso ocorra alguma exceção.

Tópico de aprendizagem: EXCEÇÕES NA MANIPULAÇÃO DE ARQUIVOS E OUTRAS OPERAÇÕES.

Referência: Conteúdo digital da disciplina, Tema "Manipulação de Dados", Módulo 2 - "funções de manipulação de strings" e Módulo 2 - "Exceções na manipulação de arquivos e outras operações".

Fim do gabarito