SISTEMA DE BIBLIOTECA EM PYTHON

GABRIELLE CRISTINE RIBEIRO DE AZEVEDO
LEONARDO FELIPE DA SILVEIRA CANS BARBOSA
LETÍCIA TINOCO VIEIRA
CURSO: AGRONOMIA
DISCIPLINA: FUNDAMENTOS DA PROGRAMAÇÃO DE COMPUTADORES

INTRODUÇÃO

- **Programa criado em *Python;***
- **Sistema de biblioteca;**
- **Controle do empréstimo e devolução de obras.**

OBJETIVOS

- Realizar a pesquisa, cadastro de usuários,
- Controlar o empréstimo e devolução de livros e periódicos criados e cadastrados em dicionários.

```python
def cadastro():
    #importar função sleep do módulo time, para dar um tempo durante a transição das funções do programa
    from time import sleep
    #imprime uma apresentação do sistema de biblioteca, com linhas
    print('-='*30)
    print('{:^30}'.format('  <<< BEM VINDO AO SISTEMA DE BIBLIOTECA AGRÁRIAS! >>>'))
    print('-='*30)
    #sleep de 1 segundo
    sleep(1)
    print('iniciando o sistema...')
    sleep(1)
    print('-' * 60)
    print('{:^60}'.format('CADASTRO'))
    print('-' * 60)
    #pedir os dados do usuário, somente aluno, professor ou usuários em geral
    dados = str(input('informe se ALUNO/PROFESSOR/USUÁRIO: ')).strip().upper()
    #strip = função que tira espaços da palavra
    #upper = função que deixa as letras da palavra digitada todas em maiúsculo
    #enquanto os dados forem diferentes de um Enter digitado pelo usuário
    while dados != " ":
        #condicionais que correspondem ao tipo de usuário a ser cadastrado
        if dados == 'ALUNO':
            #cadastrar nome e CPF dos ususários
            aluno = str(input('Nome do Aluno: '))
            cpf = int(input('CPF: '))
            print('Cadastro realizado com sucesso!')
        elif dados=='PROFESSOR':
            prof = str(input('Nome do Professor: '))
            cpf = int(input('CPF: '))
            print('Cadastro realizado com sucesso!')
```

```
        elif dados == 'USUÁRIO':
            usuario = str(input('Nome do Usuário: '))
            cpf = int(input('CPF: '))
            print('Cadastro realizado com sucesso!')
        else:
            print('Não é possível o cadastro!')
            break
        #perguntar ao usuário se deseja continuar
        resp = str(input('Continuar? [S/N] ')).upper()[0]
        while resp != 'S':
        #se o usuário digitar outra opção que não seja S ou N, o programa para e retorna à pergunta
            if resp in 'SN':
                break
            print('ERRO! responda apenas S ou N')
        #se a resposta for S, o programa prossegue à próxima função, senão, ele retorna ao cadastro
        if resp == 'S':
            break
    #função retorna os dados
    return dados
#chamar a função
cadastro()
```

```
def pesquisa():
    #importa do módulo time a função sleep
    from time import sleep
    #o programa espera 1 segundo para prossequir com o texto na tela
    sleep(1)
    print('-' * 60)
    print('{:^60}'.format('PESQUISA'))
    print('-' * 60)
    #declaração das variáveis
    resp=" "
    #função len que conta o total de livros e periódicos existentes na biblioteca
    total_livros = len(biblioteca_livros)
    total_periodicos = len(periódicos)
    #foi estabelecido que alguns livros e periódicos não são permitidos para empréstimo
    não_emprestal = 3
    não_emprestap = 3

    #condicionais de pesquisa dos exemplares
    #enquanto a resposta não for S a pergunta de continuar a pesquisa
    while resp != 'S':
        #determinar a totalidade de exemplares e a quantidade que não está disponível para empréstimo

        emprest_livros = total_livros - não_emprestal
        emprest_periodicos = total_periodicos - não_emprestap

        #perguntar o tipo de exemplar
        pesquisa = str(input('Informe tipo de exemplar:[LIVRO/PERIÓDICO] ')).strip().upper()

        #se pesquisa for livro:
        if pesquisa == 'LIVRO':
            livro = input('Digite o título do livro: ').strip().upper()
            autorl = str(input('Digite o autor do livro: '))
```

```
            #percorrer os dicionários ao longo da lista, verificando os livros existentes
            for v in biblioteca_livros:
                print(v)

            print(f'Resultados da pesquisa: {total_livros} livros no total')
            print(f'Resultados da pesquisa: {emprest_livros} livros disponíveis para empréstimo')

        #se pesquisa for periódico:
        elif pesquisa == 'PERIÓDICO':
            periodico = input('Digite o nome do periódico: ').strip().upper()

            #na biblioteca de periódicos, variável contadora p percorre a lista e imprime todos os periódicos
            for p in periódicos:
                print(p)

            print(f'Resultados da pesquisa: {total_periodicos} periódicos no total')
            print(f'Resultados da pesquisa: {emprest_periodicos} periódicos disponíveis para empréstimo')

        #perguntar ao usuário se deseja verificar as condições de empréstimo
        resp = str(input('Verificar condições do empréstimo? [S/N] ')).upper()[0]
        while not resp == 'S':
            #caso responda outra opção, que não seja S ou N, programa pergunta novamente
            if resp in 'SN':
                break
            print('ERRO! responda apenas S ou N')
        #se a resposta for S, programa prossegue para a próxima função
        if resp == 'S':
            break

#chamar a função pesquisa
pesquisa()
```

```python
def emprestimo():
    #importar do módulo time, a função sleep
    from time import sleep
    #o programa aguarda 1 segundo para mostrar texto na tela
    sleep(1)
    print('-' * 60)
    print('{:^60}'.format('EMPRÉSTIMO'))
    print('-' * 60)

    #dentro do módulo datetime, importar a função date, que calcula data e hora em python
    from datetime import date
    from datetime import datetime

    #declaração das variáveis
    futuro=0
    resp=" "

    #perguntar tipo de usuário
    dados = str(input('Informe se ALUNO/PROFESSOR/USUÁRIO: ')).upper()
    emp = str(input('Deseja realizar o empréstimo? [S/N] '))
    #após a pesquisa de exemplares, deverá haver o controle do empréstimo
    livro = int(input('Quantidade de livros a serem emprestados: '))
    per = int(input('Quantidade de periódicos a serem emprestados: '))

    #enquanto a resposta a pergunta for diferente de N
    while emp != 'N':
        #total de exemplares emprestados = livros + periódicos
        tot = livro + per
        print(f'Total de exemplares emprestados: {tot} ')

        #há condições de empréstimo para cada tipo de usuário:
```

```
#há condições de empréstimo para cada tipo de usuário:

if dados == 'ALUNO':
#se o aluno desejar emprestar mais que 2 exemplares, o sistema não permite e dá o aviso
    if tot > 2:
        print('Não é possível o empréstimo superior a 2 exemplares para ALUNO!')
    #cálculo de dias de empréstimo
    #são permitidos até 15 dias de empréstimo para ALUNO
    #data atual: dia em que ocorre o empréstimo
    '''
    Data atual:
    '''

    """
    Converter primeiramente a nossa data em um número ordinal, através do método toordinal(),
    que nos retorna a quantidade de dias passados desde o dia 1/1/1 até a data recebida como argumento.
    Depois disso, basta somar (ou subtrair) a esse número inteiro o número de dias da diferença
    que queremos calcular e então reconverter o inteiro para data, através do método fromordinal().
    Abaixo, obtivemos a data a daqui exatos x dias.
    """
    #imprime a data de devolução pelo aluno, que é a data atual + 15 dias de empréstimo possíveis
    hj = date.today()
    print(hj.toordinal())
    #calcula a data de devolução
    futuro = date.fromordinal(hj.toordinal() + 15)  # data atual + 15 dias
    print(f'Data de devolução pelo ALUNO: {futuro}')

    '''
    Calcular a diferença de dias entre datas:
    '''

    """
    Obter as duas datas entre as quais queremos saber o
```

```
    intervalo de dias e depois usar o operador de subtração (-) para fazer a operação.
    O operador subtração, quando aplicado a datas, retorna um objeto do tipo timedelta,
    contendo a diferença entre as datas. Esse objeto possui um atributo chamado days, que obviamente
    nos dá o número de dias representados pelo delta.

    """
    #imprime a quantidade de dias de empréstimo, que são 15 para o aluno
    hj = date.today()
    print(hj.toordinal())
    futuro = date.fromordinal(hj.toordinal() + 15)  # data atual + 15 dias
    #data daqui 15 dias que subtrai com a atual
    diferenca = futuro - hj
    print(f'Quantidade de dias de empréstimo: {diferenca.days}')

    #se a quantidade de exemplares for 2 ou menos, o programa imprime 'Boa Leitura!'

    if tot <=2:
        print('-' * 30)
        print('Boa Leitura!')
        print('-' * 30)
        print()

#se for professor, as condições são diferentes:

elif dados == 'PROFESSOR':
    #se professor desejar emprestar mais que 3 exemplares, o sistema não permite e dá o aviso
    if tot > 3:
        print('Não é possível o empréstimo superior a 3 exemplares!')

    #cálculo da data de devolução, para PROFESSOR são permitidos até 30 dias
    hj = date.today()
    print(hj.toordinal())
    futuro = date.fromordinal(hj.toordinal() + 30)  # data atual + 30 dias
```

```
            #cálculo da quantidade de dias de empréstimo
            diferenca = futuro - hj
            print(f'Quantidade de dias de empréstimo: {diferenca.days}')

            if tot == 1:
                print('-'*30)
                print('Boa Leitura!')
                print('-' * 30)
                print()

        #pergunta ao usuário se deseja acessar o sistema de devolução
        resp = str(input('Acessar sistema de devolução? [S/N] ')).upper()[0]
        while resp != 'S':
            if resp in 'SN':
                break
            print('ERRO! responda apenas S ou N')
        #se a resposta for S, o programa prossegue para a próxima função
        if resp == 'S':
            break

#chama a função
emprestimo()
```

```
livro1={'Título': 'Programa de melhoramento da mandioca','Autor': 'Perez e Mendonça', 'Ano':'1973',
 'Editora': 'Sul Publicações'},

livro2={'Título':'Projeto Mandioca','Autor': 'Shermann', 'Ano':'1975',
'Editora': 'Agroeco'},

livro3={'Título':'Proposta do Sistema de Gestão da Formação Profissional do Engenheiro Agrônomo da Universidade Federal '
'do Paraná','Autor': 'Silva e Lemos','Editora': 'UFPR', 'Ano':'2011'},

livro4={'Título':'Recomendação de adubação e calagem no RS e SC: passado, presente e futuro : anais',
'Autor': 'André Lopes, Souza Nunes', 'Editora': 'Morales', 'Ano':'1994'},

livro5={'Título':'Resultados de pesquisa apresentados na XII Reunião da Comissão Sulbrasileira de Pesquisa da Aveia : [anais]',
'Autor': 'João Lourenço', 'Editora': 'Manhatan', 'Ano':'1992'},

livro6={'Título':'Resultados de pesquisa em aveia obtidos na Faculdade de Agronomia da Universidade de Passo Fundo, em 1978',
'Autor': 'Cláudio Santos, Mariana Gonçalves', 'Editora': 'Faculdade de Agronomia da Universidade de Passo Fundo',
 'Ano':'1979'},

livro7={'Título':'Resultados de pesquisas da Fundação Faculdade de Agronomia "Luiz Meneghel", 1970-1984 : resumos',
'Autor': 'Luiz Meneghel','Editora': 'Sul Publicações','Ano' : '1986'},

livro8={'Título':'Rotação de culturas','Autor': 'Paolo Azevedo e Moura, Pedro Carlos Lorenzi','Editora': 'Norte',
'Ano':'1968'},

livro9={'Título':'Sanidade e produtividade em búfalos','Autor': 'Estefania Kruschovsky', 'Editora': 'Santana',
'Ano':'1993'}

biblioteca_livros = [livro1, livro2, livro3, livro4, livro5, livro6, livro7, livro8, livro9]
```

```
periódicos = [{'Nome':'Acta scientiarum : agronomy', 'Ano':'2003'},
{'Nome':'Advances in agronomy','Ano':'1949'},
{'Nome':'Agro-Ciencia','Ano':'2000'},
{'Nome':'Agrociência (Montecillo)','Ano':'2015'},
{'Nome':'Agrociencia (Montevideo)','Ano':'1950'},
{'Nome':'Agronomia','Ano':'1941'},
{'Nome':'Agronomia colombiana','Ano':'1983'},
{'Nome':'Agronomia costarricence','Ano':'1977'},
{'Nome':'Agronomia lusitana','Ano':'1939'},
{'Nome':'Agronomia mocambicana','Ano':'1967'}]
```

```
def devolução():
    from time import sleep
    sleep(1)
    print('-' * 60)
    print('{:^60}'.format('DEVOLUÇÃO'))
    print('-' * 60)

    #importa do módulo datetime, a função date
    from datetime import date
    #importa do módulo datetime, a função datetime, que calcula data e hora
    from datetime import datetime

    dados = str(input('Informe se ALUNO/PROFESSOR/USUÁRIO: ')).upper()
    #programa pede ao usuário a data em que foi devolvido o exemplar emprestado
    dev = input('Insira a data de devolução dos exemplares: ')

    #enquanto a data de devolução for diferente de um Enter digitado pelo usuário
    while dev != " ":

        #condicionais da devolução
        #se for aluno:
        if dados == 'ALUNO':

            #calcula e imprime a data que o sistema estipulou para a devolução das obras
            hj = date.today()
            print(hj.toordinal())
            futuro = date.fromordinal(hj.toordinal() + 15)
            print(f'Data prevista pelo sistema para devolução dos exemplares pelo ALUNO: {futuro}')

            #calcula e imprime a quantidade de dias que o sistema estipulou para a devolução das obras
            hj = date.today()
            print(hj.toordinal())
```

```
print(hj.toordinal())
futuro = date.fromordinal(hj.toordinal() + 15)
diferenca = futuro - hj
print(f'Quantidade de dias de empréstimo prevista pelo sistema: {diferenca.days}')

#pergunta se a quantidade de dias de empréstimo foi ultrapassada
dias = str(input('Quantidade de dias de empréstimo superior ao estipulado pelo sistema? [S/N] ')).upper()[0]
#[0] verifica somente a primeira letra digitada, sendo S ou N

#pergunta a quantidade de dias de atraso da devolução
quant = int(input('Número de dias de atraso: '))

#se a resposta for S, deverá ser aplicada multa pelo atraso da devolução das obras
while dias == 'S':
    if quant > 0:
        #se a quantidade de dias for maior que zero, a multa é calculada
        #multa = 5 reais por dia de atraso
        multa = 5 * quant
        #imprime na tela os dias de atraso;
        print(f'Dias de atraso:{quant}')
        #o valor da multa (2 casas decimais, ou flutuantes);
        print(f'Multa por atraso! Pagamento de R${multa:.2f}')
        #o sistema não permite emprestar exemplares se houve multa por atraso
        print('Não é possível o empréstimo de mais exemplares!')
        #decrementar variável quant, para não haver loop infinito
        quant -= diferenca.days
    break
    #programa para
    #pergunta ao usuário se deseja acessar o menu do sistema

menu = str(input('Acessar menu do sistema: [S/N]')).upper()[0]

if menu == 'S':
```

```
            menu = str(input('Acessar menu do sistema: [S/N]')).upper()[0]

            #se a resposta for S, o programa segue para a função menu
            if menu == 'S':
                print('ACESSANDO MENU DO SISTEMA DE BIBLIOTECA. POR FAVOR, AGUARDE!')

        #programa sai do loop existente na função e prossegue para a próxima função, se o usuário desejar
        break

#chama a função devolução
devolução()
```

```python
def menu():
    from time import sleep
    sleep(0.8)
    '''
    Função que contém o menu do programa
    '''

    # Mostra as opções para o usuário na tela
    print('MENU:')
    print('\t1 - cadastro')
    print('\t2 - pesquisa')
    print('\t3 - emprestimo')
    print('\t4 - devolução')
    print('\t5 - Sair')

    # Retorna o que o usuário digitar
    return int(input('Digite a opção desejada: '))

menu()
```

```python
def main():
    '''
    Programa principal
    '''

    # guardaremos os dados em um vetor ou array
    dados = []
    # chama a função menu que mostra o menu e solicita uma opção ao usuário
    opcao = menu()
    # enquanto o usuário não digitar a opção: 5 - Sair
    while opcao != 5:
        # Se for a opção: 1 - CADASTRO
        if opcao == 1:
            # chama a função que insere dados no vetor de dados
            cadastro()
        # Se for a opção: 2 - PESQUISA
        elif opcao == 2:
            # chama a função que pesquisa livros e periódicos na tela
            # passando o vetor que contém esses dados
            pesquisa()
            time.sleep(2)
        # Se for a opção: 3 - EMPRÉSTIMO
        elif opcao == 3:
            # chama a função que realiza o controle de empréstimo de exemplares:
            emprestimo()
            time.sleep(2)
        # Se for a opção: 4 - DEVOLUÇÃO
        elif opcao == 4:
            # chama a função que realiza o controle de devolução de exemplares:
            devolução()
            time.sleep(2)
```

```
        # Se digitou uma opção de menu inválida
        else:
            print("\nOpção inválida! Por favor, informe uma opção válida.\n")
            # Faz o programa esperar 1 segundo e depois continua
            time.sleep(1)

        # Limpa a tela do terminal para melhorar o aspecto visual da apresentação
        os.system('cls' if os.name == 'nt' else 'clear')

        # Mostra ao usuário o menu novamente e aguarda ele escolher uma opção
        opcao = menu()

    # Salva os dados em um arquivo, antes do término do programa
    salva_dados_arquivo(nome_arquivo, dados)



if __name__ == "__main__":
    '''
    Ao ser executado o programa python, esse trecho de código será executado e o programa principal será chamado.
    '''

    # Chama a função main
    main()
```

CONCLUSÃO

UFPR 105 ANOS DE ORGULHO

REFERÊNCIAS

- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS (ABNT). Informação documentação - Trabalhos acadêmicos - Apresentação. **NBR14724**. Rio de Janeiro, 2011.

- AMADEU, M. S. U. et al. **Manual de normalização de documentos científicos de acordo com as normas da ABNT**. Curitiba, 2015.

UFPR 105 ANOS DE ORGULHO