



SISTEMA DE BIBLIOTECA EM PYTHON

Ana Maria Deminciano Lima
Juliana Sardi Fernandes
Thais Trajano da Silva

Universidade Federal do Paraná
Prof. Jackson Antonio do Prado Lima



Introdução

- Python é uma linguagem de programação interpretada, de código-fonte aberto e disponível para vários sistemas operacionais.
- A problematização que será apresentada é de um programa de automatização de um sistema de biblioteca, através da linguagem python.

Objetivo do Programa

- Automatizar o sistema de biblioteca, facilitando a consulta do usuário, aos livros e aos empréstimos.
- Permitir e facilitar a atualização o estoque da biblioteca.

Utilização do Programa

- Usuário que tenha acesso liberado ao sistema (bibliotecário).
- O bibliotecário coloca os dados do aluno no sistema.
- Realização da busca por livros, onde o bibliotecário procura por meio de características que o livro contenha, como código do livro, nome do autor, ano, editora e quantidade no estoque.
- Quando for realizado um empréstimo, este dado, ficará armazenado em uma matriz, contendo o usuário, nome do livro, código deste livro e sua data de empréstimo, essa matriz, proporcionará o controle dos empréstimos.

Bibliotecas utilizadas

```
import uuid  
import hashlib  
import datetime
```

- O **import**, **uuid** e **hashlib**, são bibliotecas utilizadas para realizar a criptografia da senha que será utilizada pelo bibliotecário.

Formatação da data

```
from datetime import date

formatacaoData = "%m/%d/%Y"
now = datetime.datetime.now()

#senha 123
admin_hashed_passwd="f1e9ec4d2a8d93cdbc7109bedcce6020c1ff248a4396acfd5ac0a53a61792b53:30d24e545e2f4fbeb34e40f4fcac107a"
```

- A declaração **from datetime import date**, relata a importação do objeto datetime da biblioteca date.
- A variável **formatacaoData** é apenas para declarar a forma desejada da data.
- Já **now**, é uma variável que representa **datetime.datetime.now()**, onde é o objeto que contém o **now**.
- A variável **admin_hashed_passwd**, apresenta o código para fins de criptografar a senha, posterior será usado para validar a mesma.

Listas e matrizes

```
labelBook=["Num", "Nome", "Autor", "Ano", "Editora", "Quantidade Estoque"]
labelEmprestado=["Usuario", "Livro", "Num", "DataEmprestimo"]

books=[[1, "botanica", "lorenzi", 1990, "editoral", 10], [2, "entomologia", "lutz", 1990, "editora2", 10], [3, "genetica", "thompson", 1990, "editoral", 10],
        [4, "programação", "menezes", 1990, "editoral", 10], [5, "fisiologia", "taiz", 1990, "editoral", 10], [6, "fisica", "harley", 1990, "editoral", 10],
        [7, "calculculo", "adams", 1990, "editoral", 10]]

usuarios=[["aluno1", "thais", "agronomia"], ["aluno2", "ana", "direito"], ["aluno3", "juliana", "civil"],
           ["aluno4", "lucas", "fisica"], ["aluno5", "yuri", "mecanica"], ["aluno6", "paola", "zootecnia"]]

emprestimos=[["aluno1", "botanica", 1, "10/15/2018"], ["aluno2", "entomologia", 2, "10/11/2018"], ["aluno3", "genetica", 6, "11/02/2018"]]
```

- O código apresenta listas. A lista **labelBook**, contém **Num**, que servirá como código do livro; **Nome**, referente ao livro; **Autor**; **Ano**, de publicação; **Editora**; **Quantidade Estoque**.
- Já **labelEmprestado**, é representado por **Usuario**, que será referente ao aluno; **Livro**, determinando qual livro o usuário em questão emprestou; **Num**, o código do livro; **DataEmprestimo**, a data que o empréstimo foi realizado.
- Há também matrizes, sendo elas: **books** (apresenta código do livro; nome do livro; autor; ano de publicação; editora e quantidade no estoque), **usuários** (usuário; nome do aluno e seu curso) e **emprestimos** (representa quais usuários; seus livros emprestados; código do livro e a data dessa ocorrência).

Criptografia da senha

```
#Usar hash_password para cadastrar uma nova senha se quiser
def hash_password(password):
    salt = uuid.uuid4().hex
    return hashlib.sha256(salt.encode() + password.encode()).hexdigest() + ':' + salt

#hashed_password = hash_password(new_pass)
#print('The string to store in the db is: ' + hashed_password) o valor que será printado aqui é o valor criptografado da nova senha utilizada
```

- A seguinte função, é responsável por gerar uma nova senha, para gerar um novo código hash.
- As últimas linhas, onde apresentam-se como comentário, quando retirado suas **hashtags** proporcionarão ao usuário, um valor criptografado da nova senha.

Função senha

```
def check_password(hash_password, user_password):  
    password, salt = hash_password.split(':')  
    return password == hashlib.sha256(salt.encode() + user_password.encode()).hexdigest()
```

- A partir desta função, quando inserir uma senha, esta será criptografada e irá gerar um código hash, onde será comparado com o código da variável **admin_hashed_passwd**, validando os mesmos.

Função autenticação

```
def autenticar():  
    print("Login Bibliotecária\n")  
    passwd=input("Senha: ")  
    if check_password(admin_hashed_psswd, passwd):  
        print('Senha correta!\n')  
        return True  
    else:  
        print('Senha incorreta!\n')  
        return False
```

- Basicamente, essa função é destinada para **autenticar** o login do bibliotecário, onde necessita da senha.
- Assim, se **check_password**, onde apresenta como argumento o valor criptografado e a senha, for verdadeiro, a senha estará correta, caso contrário, incorreta.

Função para consultar o usuário

```
def consultar_usuario():  
    print("Insira o usuario que fará o empréstimo\n")  
    user = input('Usuario: ')  
    for info in usuarios:  
        if user == info[0]:  
            return True,user  
    return False,user
```

- Essa função representa a primeira integração com o usuário, onde a variável **user** receberá o nome do usuário. **Usuarios** é uma matriz, para cada lista presente, existe um usuário, que será destinado ao **info**, um em cada repetição realizada.
- Assim, quando **info** for igual ao **user**, significa que aquele usuário foi encontrado, sendo a sentença verdadeira, caso contrário, dará como falso e retornará que aquele usuário não foi encontrado.

Função para consultar empréstimos

```
def consulta_emprestados(user):  
    qtd=0  
    hoje=datetime.datetime.now()  
    for info in empréstimos:  
        if user == info[0]:  
            dloc=datetime.datetime.strptime(info[3],formatacaoData)  
            dif = hoje-dloc  
            if dif.days > 15:  
                return -1*dif.days  
            qtd+=1  
    return qtd
```

- Para `consulta_emprestados`, o argumento **user** analisará a matriz **empréstimos** para procurar se o usuário em questão apresenta algum empréstimo.
- Para essa análise, determina-se **qtd = 0**. A variável **hoje**, servirá para armazenar a data no sistema.
- Será armazenado o primeiro valor condido na primeira lista da matriz, em **empréstimo** e primeira posição, em **info**.
- Se **user** for igual ao **info**, buscasse determinar a quantidade de dias do empréstimo. Caso seja maior que 15 dias (limite), o usuário apresenta atraso na entrega, retornando então, os dias em atraso. Se o usuário não apresentar atrasos, o próximo **return**, oferece a quantidade de livros que o usuário possui emprestado.

Função para consultar os livros

```
def consulta_livros():
    print("Por favor insira o nome do livro ou o autor\n")
    busca=input("Nome/Autor: ")
    print(*labelBook)
    for info in books:
        if (busca == info[1] or busca == info[2]) and info[5]>0:
            print(*info)
    print("Escolha o livro a partir do Num equivalente, caso não tenha sido encontrado nenhum livro, ou você não queira nenhuma da lista utilize -1\n")
    opcao=int(input("Num: "))
    return opcao
```

- A função tem como objetivo auxiliar na busca do livro, pelo seu nome ou autor, para isso, a função utilizará a lista **labelBook**, assim, se a variável **busca** for igual a segunda posição (nome do livro) presente na lista **labelBook** ou igual a terceira posição (autor), e ainda a posição cinco (quantidade no estoque), for maior que zero, os dados serão fornecido na tela e o usuário deverá confirmar o código do livro do livro, caso queira emprestar.
- Se não for encontrado o livro de interesse ou não quiser emprestar nenhum livro, basta digitar -1.

Função para escolha do livro

```
def escolha(limite,user):
    while limite > 0:
        print("Escolha o tipo de exemplar que você deseja emprestar\n")
        print("E você ainda pode realizar ", limite, " empréstimos\n")
        print("1-> Livro 2->Sair")
        indice=0
        opcao=int(input("Opcao :"))
        if opcao ==1:
            livro=consulta_livros()
            if livro!=-1:
                for info in books:
                    if info[0]==livro:
                        books[indice][5]-=1
                        empréstimos.append([user,books[indice][1],books[indice][0],datetime.datetime.now().strftime("%m/%d/%Y")])
                        limite-=1
                        print("EMPRÉSTIMO REALIZADO COM SUCESSO\n")
                        break
                    else:
                        indice+=1
            elif opcao==2:
                print("Empréstimo cancelado")
                return
        if limite == 0:
            print("Você atingiu o limite de empréstimos, para poder emprestar outros primeiro devolva algum\n")
```

- A função **escolha** se refere ao limite de livros que os alunos podem retirar, que no caso é até três. Assim, para realizar um empréstimo, o limite deve ser maior que zero.
- Será analisado a primeira informação contida na primeira lista em **books** e a primeira informação contida na lista, que se refere ao código do livro, se essa informação for igual ao **livro** selecionada, então em **books** na posição cinco, que se refere a quantidade em estoque, irá diminuir uma quantidade, atualizando a saída daquele livro, pois ocorreu um empréstimo.

■ ■ ■

- O **append** é uma função que irá adicionar no final da matriz empréstimos informações, essas serão: **user** (usuário), nome do livro, seu código e data de empréstimo, para a data, utiliza o recurso do **datetime**.
- Como o empréstimo está sendo realizado, deve-se diminuir uma possibilidade de emprestar do usuário, por isso, o **limite -1**.
- O **else**, está se referindo a análise do código do livro igual ao livro de escolha, fazendo o programa percorrer toda a lista, até as duas informações serem iguais. Para opção igual a dois, o empréstimo é cancelado e se o limite for igual a zero, o usuário não poderá emprestar.

Função emprestar

```
def emprestar(user):
    quantidade=consulta_emprestados(user)
    if quantidade==0:
        print("O usuário selecionado é ", user," , e este poderá emprestar apenas três exemplares por vez e o prazo de expiração de cada empréstimo é de 15 dias!\n")
        escolha(3,user)
    elif quantidade==1:
        print("O usuário selecionado é ", user," , e este poderá emprestar apenas 2 exemplares por 15 dias pois já possui um exemplar emprestado!\n")
        escolha(2,user)
    elif quantidade==2:
        escolha(1,user)
        print("O usuário selecionado é ", user," , e este poderá emprestar apenas 1 exemplar por 15 dias pois já possui dois exemplares emprestados!\n")
    elif quantidade==3:
        print("O usuário não pode emprestar nenhum livro pois já possui três emprestados, você deve devolver um primeiro!.\n")
    elif quantidade<0:
        print("O usuário não pode emprestar nenhum pois possui um livro com entrega atrasada em ", -1*quantidade , " dia(s)\n")
```

- Para a função **emprestar**, a quantidade estará relacionada a **consulta_emprestados** para determinar se aquele usuário apresenta algum livro e se esse está em atraso.
- A variável **escolha**, guardará a quantidade de livros que poderá ser emprestada, se quantidade for igual a três, o usuário já apresenta três empréstimos e não poderá emprestar outro livro até devolver algum livro. E para quantidade menor que zero, representa que o usuário apresenta devoluções em atraso. Para essa última opção ser realizada, determinou-se **-1*quantidade**, com a finalidade de expressar os dias em atraso.

Função main

```
def main():
    sair=False
    auth=False

    while sair==False:
        #print(*emprestimos)
        if auth==False:
            auth=autenticar()
            print("Autenticação realizada com sucesso!\n")
            encontrado=False

        if auth:
            while encontrado ==False:
                encontrado,user=consultar_usuario()
                if encontrado == False:
                    print("Usuario não encontrado, procure outro")

            while encontrado == True:
                print("O que você deseja fazer?\n")
                print("1-> Emprestar 2-> Trocar Usuário")
                opcao=int(input("Opcao :"))
                if opcao ==1:
                    emprestar(user)
                elif opcao ==2:
                    encontrado=False
                else:
                    print("OPÇÃO INVÁLIDA!\n")
            else:
                print("Usuario nao encontrado\n")

        print("Deseja sair do programa? 1 ->Sim 2-> Não")
        op=int(input("Opção: "))

        if op==1:
            sair=True
            print("Programa finalizado\n")

if __name__=="__main__":
    main()
```

- Por fim, a última função, declara **sair** como **False**, pois, o intuito é permanecer no sistema, **auth** também como **False** até que o bibliotecário insira a senha correta.
- Estabelece **encontrado** igual a **False**, para que a busca pelo usuário só se torne True com o usuário correto, assim se **auth**, o **encontrado** for igual a **False**, irá buscar o **user** em **consultar_usuario**.
- Mas se **encontrado** for igual a **True**, quer dizer que o usuário que pretende realizar o empréstimo e está cadastrado no sistema.
- Se o usuário digitar qualquer outro número diferente de um ou dois, dará opção inválida.
- No final, o usuário deverá escolher permanecer ou não no programa, se a opção for igual a um, então **sair** se tornará **True** e o programa será finalizado.

➤ Obrigada a todos por assistirem a nossa apresentação!

```
def consultar_usuario():
    print("Insira o usuario que fará o empréstimo\n")
    user = input('Usuario: ')
    for info in usuarios:
        if user == info[0]:
            return True,user
    return False,user

def consulta_emprestados(user):
    qtd=0
    hoje=datetime.datetime.now()
    for info in emprestimos:
        if user == info[0]:
            dloc=datetime.datetime.strptime(info[3],formatacaoData)
            dif = hoje-dloc
            if dif.days > 15:
                return -1*dif.days
            qtd+=1
    return qtd
```

```
def escolha(limite,user):
    while limite > 0:
        print("Escolha o tipo de exemplar que você deseja emprestar\n")
        print("E você ainda pode realizar ", limite, " empréstimos\n")
        print("1-> Livro 2->Sair")
        indice=0
        opcao=int(input("Opcao :"))
        if opcao ==1:
            livro=consulta_livros()
            if livro!=-1:
                for info in books:
                    if info[0]==livro:
                        books[indice][5]-=1
                        emprestimos.append([user,books[indice][1],books[indice][0],datetime.datetime.now().strftime("%m/%d/%Y")])
                        limite-=1
                        print("EMPRÉSTIMO REALIZADO COM SUCESSO\n")
                        break
                else:
                    indice+=1
            elif opcao==2:
                print("Empréstimo cancelado")
                return
            if limite == 0:
                print("Você atingiu o limite de empréstimos, para poder emprestar outros primeiro devolva algum\n")
```

```
def main():
    sair=False
    auth=False

    while sair==False:
        #print("empréstimos")
        if auth==False:
            auth=autenticar()
            print("Autenticação realizada com sucesso!\n")
            encontrado=False

        if auth:
            while encontrado ==False:
                encontrado,user=consultar_usuario()
                if encontrado == False:
                    print("Usuario não encontrado, procure outro")

            while encontrado == True:
                print("O que você deseja fazer?\n")
                print("1-> Empréstimo 2-> Trocar Usuário")
                opcao=int(input("Opcao :"))
                if opcao ==1:
                    emprestar(user)
                elif opcao ==2:
                    encontrado=False
                else:
                    print("OPÇÃO INVÁLIDA!\n")

            else:
                print("Usuario nao encontrado\n")

        print("Deseja sair do programa? 1 ->Sim 2-> Não")
        op=int(input("Opção: "))

        if op==1:
            sair=True
            print("Programa finalizado\n")

if __name__=="__main__":
    main()
```