



**UNIVERSIDADE FEDERAL DO PARANÁ**  
**CURSO DE AGRONOMIA**

ANA MARIA DEMINCIANO LIMA  
JULIANA SARDI FERNANDES  
THAIS TRAJANO DA SILVA

**SISTEMA DE BIBLIOTECA EM PYTHON**

**CURITIBA**  
**2018**

ANA MARIA DEMINCIANO LIMA (GRR20140289)

JULIANA SARDI FERNANDES (GRR20180828)

THAIS TRAJANO DA SILVA (GRR20154378)

## **SISTEMA DE BIBLIOTECA EM PYTHON**

Relatório apresentado à disciplina  
Fundamentos de Programação de  
Computadores do Curso de Graduação  
em Agronomia da Universidade Federal  
do Paraná.

Orientador: Prof. Jackson Antônio do  
Prado Lima

**Curitiba, novembro de 2018**

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>4</b>
<b>2</b>	<b>OBJETIVOS</b>	<b>5</b>
2.1	OBJETIVO GERAL	5
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>6</b>

## **1 INTRODUÇÃO**

Python é uma linguagem de programação interpretada, de código-fonte aberto e disponível para vários sistemas operacionais. Para a presente atividade, a problematização utilizada, será a automatização do sistema de biblioteca, através da linguagem python.

## **2 OBJETIVOS**

### **2.1 OBJETIVO GERAL**

O objetivo é automatizar o sistema de biblioteca, tornando-o as atividades de consulta ao usuário, aos livros e os empréstimos, mais facilitado, além de atualização o estoque da biblioteca.

### 3 DESENVOLVIMENTO

Como já relatado, o desenvolvimento será através da linguagem python. O intermediário utilizado será uma pessoa com acesso ao sistema, há exemplo, um bibliotecário, onde irá proporcionar a vinculação do usuário ao sistema da biblioteca. Através da pessoa autorizado, poderá ser realizado a busca por livros, onde estes apresentarão características relacionadas ao seu código, nome, nome do autor, ano, editora e quantidade no estoque. O usuário também apresentará dados, como nome e curso. Quando um usuário realizar 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.

Para a realização do código, algumas bibliotecas foram necessárias:

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

Figura 1 – Bibliotecas utilizadas

O import uuid e hashlib, são bibliotecas utilizadas para realizar a criptografia da senha que será utilizada pelo bibliotecário. O import datetime é uma biblioteca utilizada para o controle da data, onde está, carrega uma série de dados que permite ter acesso a data do sistema, caso necessário para verificar os atrasos dos livros.

```
from datetime import date

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

#senha 123
admin_hashed_passwd="f1e9ec4d2a8d93cdbe7109bedcce6020c1ff248a4396acfd5ac0a53a61792b53:30d24e545e2f4fbeb34e40f4fcac107a"
```

Figura 2- Formatação da data

Continuando com o código, 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.

Para o presente código, a senha escolhida é 123.

A variável admin\_hashed\_passwd, apresenta o código para fins de criptografar a senha, posterior será usado para validar a mesma.

```

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,"calculo","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"]]

```

Figura 3 – Listas e matrizes

O código apresenta listas, representadas por labelBook, labelEmprestado. A lista labelBook, contém Num, que servirá como código do livro; Nome, referente ao livro; Autor, destinado ao autor; Ano, de publicação; Editora; Quantidade Estoque, determinando a quantidade do livro presente na biblioteca. 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, usuários, emprestimos. A matriz books, apresenta primeiramente o código do livro, ou seja, Num; posteriormente o nome do livro, do autor, ano de publicação, editora e quantidade em estoque. Para usuarios, a ordem de início é pelo usuário, nome do aluno e o seu curso. Por fim, empréstimos, representa quais usuários, assim, seus livros emprestados, código deste livro e a data desta ocorrência.

```

#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

```

Figura 4 – Criptografia da senha

A seguinte função, é responsável por gerar uma nova senha, para gerando 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.

```

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

```

Figura 5 – Função senha

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.

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

Figura 6 – Função autenticação

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.

```
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
```

Figura 7 – Função para consultar o usuário

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, se não achar nenhum usuário nas listas, dará como falso e retornará que aquele usuário não foi encontrado.



```

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

```

Figura 8 – Função para consultar empréstimos

Para `consulta_emprestados`, o argumento `user` analisará a matriz `emprestimos` para procurar se o usuário em questão apresenta algum empréstimo. Para essa análise começar no início, determina-se `qtd = 0`, para a análise começar na primeira posição. A variável `hoje`, servirá para armazenar a data do sistema. A linha posterior, armazenará o primeiro valor condido na primeira lista da matriz `emprestimo` e primeira posição, em `info`. Assim contendo o primeiro empréstimo dentro de `info`. Posterior, se `user` for igual ao `info`, buscasse determinar a quantidade de dias do empréstimo deste a data do empréstimo, se for maior que 15 dias – sendo 15 dias o limite para o usuário permanecer com os livros – 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.

```

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

```

Figura 9 – Função para consultar os livros

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 presente na lista `labelBook`, que representa o nome do livro ou igual a terceira posição, denominada autor, e ainda a posição cinco, que corresponde a quantidade estoque for maior que zero, os dados serão fornecido na tela e o usuário deverá confirmar o `Num` do livro, ou seja, o código do livro que está emprestando, caso queira emprestar o mesmo, se não for encontrado o livro de interesse ou não quiser emprestar nenhum livro, basta digitar -1.

```

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")

```

Figura 10 – Função para escolha do livro

A função escolha é mais uma interação com o usuário, apresenta como argumento limite, se referindo ao limite de livros que os alunos podem retirar, que no caso é até 3 livros e o atributo user, se referindo a este aluno/usuário. Assim, para realizar um empréstimo, o limite deve ser maior que zero, se for zero, é denominado que o presente usuário não apresenta nenhum empréstimo. A partir dessa função, o usuário poderá escolher o livro que deseja e saber quantos empréstimos pode realizar. Denomina-se índice=0, para o programa saber qual o valor da lista que este deve estar analisando. Se a opção for igual a um, quer dizer que o usuário pretende emprestar um livro. Assim, se livro for diferente de -1 quer dizer que o usuário pretende fazer o empréstimo, devido a condição estabelecida em consulta\_livros, que na qual, o -1 digitado pelo usuário, significava que o mesmo não achou o livro de interesse ou não queria emprestar. Posterior, será analisado a primeira informação contida na primeira lista em books e a primeira informação contida na lista, que se refere ao num, 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 referentes aos empréstimos realizados, essas informações é o user (usuário), o nome do livro, seu código e a 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 menos um. Assim, o empréstimo será realizado. 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.

```
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")
```

Figura 11 – Função emprestar

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. Caso quantidade for igual a zero, o usuário não apresenta nenhum empréstimo, então poderá emprestar três livros. A variável escolha, guardará a quantidade de livros que poderá ser emprestada e o usuário. Assim, 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.

```

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-> Empréstimo 2-> Trocar Usuário")
                opcao=int(input("Opção: "))
                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()

```

Figura 12 – Função main

Por fim, a última função, sendo ela a principal. Declara sair como false, pois, o intuito é permanecer no sistema. Auth também como false até que o bibliotecário insira a senha correta, a partir da inserção da senha correta, auth será alterado para true. Assim, sendo sair igual a false e auth igual a false, a autenticação será realizada com sucesso.

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, caso o encontrado for false, o usuário então não foi encontrado. Mas se encontrado for igual a true, quer dizer que o usuário que pretende realizar o empréstimo está cadastrado no sistema, podendo este, ter como opção, emprestar ou trocar de usuário. Para a primeira opção, este irá emprestar, caso seja, segunda opção, determina novamente encontrado como false, realizando a busca de

usuários novamente, se por acaso o usuário digitar qualquer outro número diferente de um ou dois, dará opção inválida.

Posterior, o usuário deve escolher permanecer ou não no programa, se a opção for igual a um, então sair se tornará true e o programa será finalizado.