



SYSFILMES

SysFilmes é uma programa de controle de filmes

Produzido por Kayky Castro Lacerda | 558165

Fluxograma

Acessar Fluxograma

Cabeçalho



```
#IMPORTAÇÃO DE FUNÇÕES
```

```
from IPython.display import clear_output
```

```
import os.path
```

```
#Banco de dados local
```

```
filmes = []
```

Escopo Principal do Programa

```
while option != 9:
```

```
    #Aqui amazena a escolha do usuário toda vida que o laço repete
```

```
    print(mostra_menu(len(filmes)))
```

```
    option = int(input("Digite a opção desejada: "))
```

```
    clear_output()
```



```
***** SysFilmes *****  
***** Existem: 0 filmes *****  
*****  
1. Cadastrar Filme  
2. Avaliar Filme  
3. Consultar Filme por Título  
4. Listar Filmes por Gênero  
5. Listar Filmes por Estrelas  
6. Listar Todos os Filmes  
7. Carregar Filmes de Arquivo  
8. Carregar Avaliações de Arquivo  
9. Sair do Sistema
```

Escopo Principal do Programa Opção 1

```
if option == 1:
```

```
    #Essa opção executa a criação de um novo filme e o cadastro dele
```


```
    novo_filme = cria_filme()
```

```
    filmes.append(novo_filme)
```

```
    clear_output()
```

```
***** SysFilmes *****
***** Existem: 0 filmes *****
*****

1. Cadastrar Filme
2. Avaliar Filme
3. Consultar Filme por Título
4. Listar Filmes por Gênero
5. Listar Filmes por Estrelas
6. Listar Todos os Filmes
7. Carregar Filmes de Arquivo
8. Carregar Avaliações de Arquivo
9. Sair do Sistema
```



```
def cria_filme():
    """ A função cadastrar_filmes mostra o menu de cadastro de filmes, captura alguns dados
    do filme e cadastra filmes
    """
    print(mostra_menu_cadastrar_filme())

    #Aqui está capturando os dados
    titulo = input("Titulo: ")
    ano = int(input("Ano: "))
    genero = input("Genero: ")
    estrelas = 0.0
    avaliacoes = 0

    #Aqui está armazenando em uma estrutura de registro para facilitar a utilização dos dados
    filme = {
        "title" : titulo,
        "year" : ano,
        "gender" : genero,
        "stars" : estrelas,
        "reviews" : avaliacoes
    }

    atualiza_filmes(filme)

    #Aqui está retornando os dados para e limpando o output
    input("[**Tecla enter para voltar ao Menu Principal**]")
    return filme
    clear_output()
```



```
def atualiza_filmes(filme):
    """Essa função atualiza o arquivo de filmes com um novo filme passado pelo o usuário
    """

    #titulo é uma variável que armazena o titulo do filme passado por parametro
    titulo = filme['title']
    #ano é uma variável que armazena o ano do filme passado por parametro
    ano = filme['year']
    #genero é uma variável que armazena o genero do filme passado por parametro
    genero = filme['gender']

    #filme é uma variável que armazena a estrutura correta para salvar o filme no arquivo de filmes
    filme = f"{titulo},{ano},{genero}\n"

    #nome_arquivo é uma variável que armazena o nome do arquivo
    nome_arquivo = input("Digite o nome do arquivo: ")

    #Aqui checa se o filme existe usando uma função chamada os.path ela retorna True ou False
    check_file = os.path.isfile(nome_arquivo)

    if check_file:
        #arquivo é uma variável que abre o arquivo e amazena
        arquivo = open(f"{nome_arquivo}", "r+")
        #arquivo_por_linha é uma variável que lê e separa por linhas o arquivo
        arquivo_por_linha = arquivo.readlines()

        #Esse loop percorre todos os filmes do arquivo passado pelo o usuário
        for filme in arquivo_por_linha:
            #verificando_ja_cadastrado é uma variável que retorna se o filme já existe no arquivo
            verificando_ja_cadastrado = verifica_filme_existente(titulo,ano,filme)

            #Essa essa estrutura condicional cadastra o filme, se somente se, o filme não exista no arquivo
            if verificando_ja_cadastrado:
                print("O filme já está cadastrado no arquivo! ")
            else:
                arquivo.write(filme)
                print("Filmes salvo com sucesso! ")
                arquivo.close()
            else:
                print("Arquivo não encontrado! ")
```



```
def verifica_filme_existente(titulo, ano, filme):  
    """Essa função verifica se o filme já existe no arquivo de filmes"""  
    if filme[0].lower() == titulo.lower() and filme[2] == ano:  
        return True  
    else:  
        return False
```


Escopo Principal do Programa Opção 2

```
elif option == 2:
```

```
    #Essa opção avalia o filme que o usuário escolher
```


```
    avalia_filme(filmes)
```

```
    input("[**Tecle enter para voltar ao Menu Principal**]")
```

```
    clear_output()
```



```
***** SysFilmes *****  
***** Existem: 0 filmes *****  
*****  
1. Cadastrar Filme  
2. Avaliar Filme  
3. Consultar Filme por Título  
4. Listar Filmes por Gênero  
5. Listar Filmes por Estrelas  
6. Listar Todos os Filmes  
7. Carregar Filmes de Arquivo  
8. Carregar Avaliações de Arquivo  
9. Sair do Sistema
```



```
def avalia_filme(filmes):
    """Essa função está avaliando os filmes que o usuário escolheu com base no título"""

    #title é a variável que armazena o título do filme que o usuário quer avaliar
    title = input("Digite o título do filme desejado: ")

    #resultado_da_pesquisa é a variável que armazena o resultado e os dados do filme, caso ele exista
    resultado_da_pesquisa = busca_titulo(title, filmes)

    #Essa estrutura condicional só funciona quando o resultado da pesquisa do filme tenha resultado verdadeiro
    if resultado_da_pesquisa:
        #star armazena a avaliação do usuário
        star = float(input("Quantas estrelas você atribui a esse filme? (1 a 5): "))


        #Esse laço verifica se a avaliação é válida
        while star > 5 or star < 0:
            print("Avaliação inválida! Tente novamente:")
            star = float(input("Quantas estrelas você atribui a esse filme? (1 a 5): "))

        #Aqui verifica se as estrelas são 0, caso seja adiciona a primeira estrela como valor total, caso não faz um cálculo e adiciona
        if resultado_da_pesquisa['stars'] == 0.0:
            resultado_da_pesquisa['stars'] = star
            atualiza_avaliacoes(title, star)

        #Cálculo
        resultado_da_pesquisa['stars'] = (resultado_da_pesquisa['stars'] + star) / 2
        resultado_da_pesquisa['reviews'] += 1
        atualiza_avaliacoes(title, resultado_da_pesquisa['stars'])
```



```
def busca_titulo(titulo, filmes):  
    r"""Essa função recebe o parametro titulo e verifica se há algum  
    filme com o mesmo título e retorna o mesmo,  
    caso não, retorna None"""  
    for filme in filmes:  
        if filme['title'].lower() == titulo.lower():  
            return filme  
  
    return None
```



```
def avalia_filme(filmes):
    """Essa função está avaliando os filmes que o usuário escolheu com base no título"""

    #title é a variável que armazena o título do filme que o usuário quer avaliar
    title = input("Digite o título do filme desejado: ")

    #resultado_da_pesquisa é a variável que armazena o resultado e os dados do filme, caso ele exista
    resultado_da_pesquisa = busca_titulo(title, filmes)


    #Essa estrutura condicional só funciona quando o resultado da pesquisa do filme tenha resultado verdadeiro
    if resultado_da_pesquisa:
        #star armazena a avaliação do usuário
        star = float(input("Quantas estrelas você atribui a esse filme? (1 a 5): "))

        #Esse laço verifica se a avaliação é válida
        while star > 5 or star < 0:
            print("Avaliação inválida! Tente novamente:")
            star = float(input("Quantas estrelas você atribui a esse filme? (1 a 5): "))

        #Aqui verifica se as estrelas são 0, caso seja adiciona a primeira estrela como valor total, caso não faz um cálculo e adiciona
        if resultado_da_pesquisa['stars'] == 0.0:
            resultado_da_pesquisa['stars'] = star
            atualiza_avaliacoes(title, star)

        #Cálculo
        resultado_da_pesquisa['stars'] = (resultado_da_pesquisa['stars'] + star) / 2
        resultado_da_pesquisa['reviews'] += 1
        atualiza_avaliacoes(title, resultado_da_pesquisa['stars'])
```

```
def atualiza_avaliacoes(titulo, stars):  
    """Essa função atualiza o arquivo de avaliações com uma nova avaliação passada pelo o  
    usuário  
    """  
  
    #filme_avalidado é uma variável que armazena a estrutura correta para salvar a valiação no  
    arquivo de avaliação  
    filme_avalidado = f"{titulo},{stars}\n"  
    #nome_arquivo é uma variável que armazena o nome do arquivo  
    nome_arquivo = input("Digite o nome do arquivo: ")  
  
    #Aqui checa se o filme existe usando uma função chamada os.path ela retorna True ou  
    False  
    check_file = os.path.isfile(nome_arquivo)  
  
    if check_file:  
        #arquivo é uma variável que abre o arquivo e amazena  
        arquivo = open(f"{nome_arquivo}", "r+")  
        #arquivo_por_linha é uma variável que lê e separa por linhas o arquivo  
        arquivo_por_linha = arquivo.readlines()  
  
        #Esse loop percorre todos as avaliações do arquivo passado pelo o usuário  
        for filme in arquivo_por_linha:  
            #verificando_ja_cadastrado é uma variável que retorna se a valiação já existe no arquivo  
            verificando_ja_cadastrado = verifica_filme_existente(titulo,stars,filme)  
  
            #Essa essa estrutura condicional cadastra a avaliação, se somente se, a avaliação não exista  
            no arquivo  
            if verificando_ja_cadastrado:  
                print("O filme já está cadastrado no arquivo! ")  
            else:  
                arquivo.write(filme_avalidado)  
                print("Filmes salvo com sucesso! ")  
                arquivo.close()  
            else:  
                print("Arquivo não encontrado! ")
```



```
def verifica_avaliacao_existente(titulo, stars, filme):  
    """Essa função verifica se a avaliação já existe no arquivo de avaliações"""  
    if filme[0].lower() == titulo.lower() and filme[3] == stars:  
        return True  
    else:  
        return False
```

Escopo Principal do Programa Opção 3

```
elif option == 3:
```

```
    #Essa opção consulta o filme de acordo com o titulo  
    que o usuário escolher
```


```
    consulta_titulo(filmes)
```

```
    input("[**Tecle enter para voltar ao Menu Principal**]")
```


```
    clear_output()
```



```
***** SysFilmes *****  
***** Existem: 0 filmes *****  
*****  
1. Cadastrar Filme  
2. Avaliar Filme  
3. Consultar Filme por Título  
4. Listar Filmes por Gênero  
5. Listar Filmes por Estrelas  
6. Listar Todos os Filmes  
7. Carregar Filmes de Arquivo  
8. Carregar Avaliações de Arquivo  
9. Sair do Sistema
```



```
def consulta_titulo(filmes):  
    """Essa função procura e lista filmes com base no título escolhido pelo usuário,  
    passando para outra função chamada busca_titulo() que verifica se existe ou não o  
    filme"""  
    #title é a variável que armazena o título do filme que o usuário quer procurar  
    title = input("Digite o título do filme desejado: ")  
    #resultado_da_pesquisa é a variável que armazena o resultado e os dados do filme,  
    caso ele exista  
    resultado_da_pesquisa = busca_titulo(title, filmes)  
  
    #Essa estrutura condicional verifica se o filme existe e exibe o filme caso ele exista  
    if resultado_da_pesquisa:  
        mostra_filme(resultado_da_pesquisa)  
    else:  
        print("Filme não encontrado!")
```

```
def mostra_filme(filme):  
    """Essa função está sendo usada apenas como estrutura de organização para printar os  
    dados da sua chamada"""  
    print("")  
    print(f"Título: {filme['title']}")  
    print(f"Ano: {filme['year']}")  
    print(f"Gênero: {filme['gender']}")  
    print(f"Estrelas: {filme['stars']}")  
    print(f"Número de avaliações: {filme['reviews']}")  
    print("")
```

Escopo Principal do Programa Opção 4

```
elif option == 4:
```

```
#Essa opção lista todos os filmes de acordo com o genero escolhido pelo usuário
```

```
genero = input("Digite o gênero desejado: ")
```


```
lista_genero(genero, filmes)
```

```
input("[**Tecle enter para voltar ao Menu Principal**]")
```

```
clear_output()
```



```
***** SysFilmes *****  
***** Existem: 0 filmes *****  
*****  
1. Cadastrar Filme  
2. Avaliar Filme  
3. Consultar Filme por Título  
4. Listar Filmes por Gênero  
5. Listar Filmes por Estrelas  
6. Listar Todos os Filmes  
7. Carregar Filmes de Arquivo  
8. Carregar Avaliações de Arquivo  
9. Sair do Sistema
```



```
def lista_genero(genero, filmes):  
    """Essa função está listando filmes com gênero específico da escolha do usuário, ela usa  
    um laço para  
    printar todos os filmes que condizem com a condição necessária """  
    for filme in filmes:  
        if filme['gender'] == genero:  
            mostra_filme(filme)
```

Escopo Principal do Programa Opção 5

```
elif option == 5:
```

```
    #Essa opção lista todos os filmes de acordo com a quantidade de estrelas  
    escolhida pelo usuário
```

```
    num = float(input("Digite o desejado: "))
```

```
    while num > 5 or num < 0:
```

```
        print("Número inválido. Tente novamente.")
```


```
    num = float(input("Digite o desejado: "))
```

```
    lista_estrelas(num, filmes)
```

```
    input("[**Tecle enter para voltar ao Menu Principal**]")
```

```
    clear_output()
```

```
***** SysFilmes *****  
***** Existem: 0 filmes *****  
*****  
1. Cadastrar Filme  
2. Avaliar Filme  
3. Consultar Filme por Título  
4. Listar Filmes por Gênero  
5. Listar Filmes por Estrelas  
6. Listar Todos os Filmes  
7. Carregar Filmes de Arquivo  
8. Carregar Avaliações de Arquivo  
9. Sair do Sistema
```



```
def lista_estrelas(num, filmes):  
    """Essa função está listando filmes com numero de estrelas específico da escolha do  
    usuário, ela usa um laço para  
    printar todos os filmes que condizem com a condição necessária """  
    for filme in filmes:  
        if filme['stars'] >= num:  
            mostra_filme(filme)
```

Escopo Principal do Programa Opção 6

```
elif option == 6:
```

```
    #Essa opção lista todos os filmes independente de qualquer parametro
```

```
    lista_todos(filmes)
```

```
    input("[**Tecle enter para voltar ao Menu Principal**]")
```

```
    clear_output()
```

```
***** SysFilmes *****
***** Existem: 0 filmes *****
*****
1. Cadastrar Filme
2. Avaliar Filme
3. Consultar Filme por Título
4. Listar Filmes por Gênero
5. Listar Filmes por Estrelas
6. Listar Todos os Filmes
7. Carregar Filmes de Arquivo
8. Carregar Avaliações de Arquivo
9. Sair do Sistema
```



```
def lista_todos(filmes):  
    r"""Essa função lista utilizando um laço, todos os filmes que estão armazenados"""  
    for filme in filmes:  
        mostra_filme(filme)
```

Escopo Principal do Programa Opção 7

```
elif option == 7:
```

```
    #Essa opção carrega filmes de um arquivo escolhido pelo usuário
```

```
    carrega_filmes(filmes)
```

```
    input("[**Tecle enter para voltar ao Menu Principal**]")
```

```
    clear_output()
```

```
***** SysFilmes *****
***** Existem: 0 filmes *****
*****
1. Cadastrar Filme
2. Avaliar Filme
3. Consultar Filme por Título
4. Listar Filmes por Gênero
5. Listar Filmes por Estrelas
6. Listar Todos os Filmes
7. Carregar Filmes de Arquivo
8. Carregar Avaliações de Arquivo
9. Sair do Sistema
```



```
def carrega_filmes(filmes):
    """A função carrega_filmes carrega filmes de um arquivo selecionado pelo usuário"""

    #nome_arquivo é uma variável que armazena o nome do arquivo
    nome_arquivo = input("Digite o nome do arquivo: ")

    #Aqui checo se o filme existe usando uma função chamada os.path ela retorna True ou False
    check_file = os.path.isfile(nome_arquivo)

    #Estrutura condicional que só entra se o arquivo exista
    if check_file:

        #arquivo é uma variável que abre o arquivo e amazena
        arquivo = open(f"{nome_arquivo}", "r")
        #arquivo_por_linha é uma variável que lê e separa por linhas o arquivo
        arquivo_por_linha = arquivo.readlines()

        #Essa estrutura pega os filmes do arquivo e adiciona a lista de filmes local
        for i in range(1, len(arquivo_por_linha)):
            #linha é uma variável que armazena os filmes por linha com os dados separados por virgulas
            linha = []
            linha = arquivo_por_linha[i].split(",")

            #Separando os dados dos filmes
            titulo = linha[0]
            ano = linha[1]
            genero = linha[2].replace("\n", "")
            estrelas = 0.0
            avaliacoes = 0

            #Dicionário organizador com os dados dos filmes
            filme = {
                "title": titulo,
                "year": ano,
                "gender": genero,
                "stars": estrelas,
                "reviews": avaliacoes
            }

            #Salvando o filme na lista
            filmes.append(filme)

        print("Filmes carregados com sucesso!")
        arquivo.close()

    else:
        print("Arquivo não encontrado!")
```

Escopo Principal do Programa Opção 8

```
elif option == 8:
```

```
    #Essa opção carrega avaliações de um arquivo escolhido pelo usuário
```

```
    carrega_avaliacoes(filmes)
```

```
    input("[**Tecle enter para voltar ao Menu Principal**]")
```

```
    clear_output()
```

```
***** SysFilmes *****  
***** Existem: 0 filmes *****  
*****  
1. Cadastrar Filme  
2. Avaliar Filme  
3. Consultar Filme por Título  
4. Listar Filmes por Gênero  
5. Listar Filmes por Estrelas  
6. Listar Todos os Filmes  
7. Carregar Filmes de Arquivo  
8. Carregar Avaliações de Arquivo  
9. Sair do Sistema
```

```
def carrega_avaliacoes(filmes):
    """Essa função carrega as avaliações que estão salvas em um arquivo que o usuario
    selecionar,
    caso o filme exista na lista"""

    #nome_arquivo é uma variável que armazena o nome do arquivo
    nome_arquivo = input("Digite o nome do arquivo: ")

    #Aqui checa se o filme existe usando uma função chamada os.path ela retorna True ou
    False
    check_file = os.path.isfile(nome_arquivo)


    if check_file:
        #arquivo é uma variável que abre o arquivo e amazena
        arquivo = open(f"{nome_arquivo}", "r")
        #arquivo_por_linha é uma variável que lê e separa por linhas o arquivo
        arquivo_por_linha = arquivo.readlines()

        for i in range(1, len(arquivo_por_linha)):
            #linha é uma variável que armazena os filmes por linha com os dados separados por
            virgulas
            linha = []
            linha = arquivo_por_linha[i].split(",")
            titulo = linha[0]
            estrelas = float(linha[1])

            #resultado_da_pesquisa armazena os dados do filme caso ele exista
            resultado_da_pesquisa = busca_titulo(titulo, filmes)

            #Essa estrutura condicional ativa caso o filme exista e amazena as estrelas e reviews na
            lista de filmes local
            if resultado_da_pesquisa:
                if resultado_da_pesquisa['stars'] == 0.0:
                    resultado_da_pesquisa['stars'] = estrelas
                    resultado_da_pesquisa['reviews'] += 1
                else:
                    resultado_da_pesquisa['stars'] = (resultado_da_pesquisa['stars'] + estrelas) / 2
                    resultado_da_pesquisa['reviews'] += 1
                print("Avaliações carregadas com sucesso!")
            arquivo.close()
        else:
            print("Arquivo não encontrado!")
```

Tratamento de Erros



```
elif option > 9 or option < 1:  
    #Essa condicional trata erros de digitação do usuário  
    print("Opção Inválida.Tecle enter para repetir!")  
    print("")  
    input("[**Tecle enter para voltar ao Menu Principal**]")
```

Escopo Principal do Programa Opção 9

```
while option != 9:
```

```
    #Aqui amazena a escolha do usuário toda vida que o laço repete
```

```
    print(mostra_menu(len(filmes)))
```

```
    option = int(input("Digite a opção desejada: "))
```

```
    clear_output()
```



```
***** SysFilmes *****  
***** Existem: 0 filmes *****  
*****  
1. Cadastrar Filme  
2. Avaliar Filme  
3. Consultar Filme por Título  
4. Listar Filmes por Gênero  
5. Listar Filmes por Estrelas  
6. Listar Todos os Filmes  
7. Carregar Filmes de Arquivo  
8. Carregar Avaliações de Arquivo  
9. Sair do Sistema
```

Fim do Programa

```
clear_output()  
print("[**Bye, você saiu do SysFilmes!**}")
```

Créditos SysFilmes

Organizado por Kayky Castro

Programado por Kayky Castro(Sem código do GPT)

Professor Instrutor por Linconln Rocha

Bugs por Kayky Castro

Produzido em Universidade Federal do Ceará

Código sujo por Kayky Castro

Modelo seguido: Go Horse

Fontes de pesquisa google, w3school, stack overflow

Modelo de documentação: PEP 257

Documentado por Kayky Castro

GitHub Kayky Castro

Acesse os arquivos do projeto(Click aqui!)

Contato: kaykycastrocontato@gmail.com