



Documentação do projeto de Programação II

Instituto Politécnico da Lusófona
Nº de aluno: a22100540
Aluno: Felipe Silva
Ano letivo: 2021/22
Turma: 1º ano - GSI



Índice de imagens

Figura 1 - PyDocs.....	4
Figura 2 - Menu.....	5
Figura 3 - Função voltar	5
Figura 4 - Função validar string.....	6
Figura 5 - Função validar int.....	6
Figura 6 - Função idade.....	7
Figura 7 - Função Classe Info	8
Figura 8 - Função Classe apagar.....	8
Figura 9 - Conexão Base de dados	9
Figura 10 – Query.....	9
Figura 11 - Função Python	10
Figura 12 - Função Python 2	10



Índice

<i>PyDocs</i>	<i>4</i>
<i>Funções em classes.....</i>	<i>8</i>
<i>Base de dados</i>	<i>9</i>
<i>Funções Python</i>	<i>10</i>



PyDocs

Depois de feita a pesquisa sobre a documentação “Python”, não foi possível obter os resultados pretendidos. De qualquer das formas, adiciono aqui uma imagem do resultado. Na documentação “Docstrings”, consegui fazer comentários e navegar pela documentação dos módulos de “Python”.

Entretanto, decidi fazer as explicações das funções, utilizando “screen shots” código.

```
main index  
/Users/felipe/Documents/FelipeSilva_a22100540_ProgII/src/main.py  
#main  
Modules  
pandas  
Data  
aluno = <class_aluno.Aluno object>  
c = <mysql.connector.cursor.MySQLCursor object>  
db = <mysql.connector.connection.MySQLConnection object>  
disc = <class_disciplina.Disciplina object>  
escolha = 0  
lista_alunos = []  
lista_discs = []  
lista_profs = []  
prof = <class_professor.Professor object>
```

Figura 1 - PyDocs



Funções

```
def menu(): #opcoes do menu
    print("1 - Criar Disciplina")
    print("2 - Listar Disciplinas")
    print("3 - Eliminar Disciplina")
    print("4 - Criar Aluno")
    print("5 - Inscrever Aluno")
    print("6 - Listar Alunos")
    print("7 - Eliminar Aluno")
    print("8 - Listar Alunos inscritos numa dada disciplina")
    print("9 - Criar Professor")
    print("10 - Adicionar professor a uma disciplina")
    print("11 - Lista de professores para cada disciplina")
    print("12 - Importar alunos de um ficheiro")

    print("\n0 - Sair")
    pass

def menu_importar(): #opcoes do menu de importacao
    print("\n11 - Importar alunos de um ficheiro")
    print("\n1 - Importar ficheiro (disciplinas, alunos e professores)\n")
    print("2 - Importar alunos para uma disciplina\n")
```

Figura 2 - Menu

Crie duas funções apenas para fazer o print das opções do menu principal e do menu de importações de ficheiros.

```
def voltar(): #Enter para retornar

    while True:
        enter = input("Pressione Enter para retornar..")
        print("")
        if not enter:
            break
```

Figura 3 - Função voltar

A função “voltar”, serve apenas para quando necessário, o utilizador retornar para o menu principal, pressionando a tecla “Enter”.

Está a “correr” dentro de um ciclo while, o utilizador por engano pressiona ou digite outros dados incorretos, neste caso, continua após o valor estar correto.



Continuação – Funções de Input

```
def input_str(texto): #receber input do utilizador e validar, numeros e caracteres especiais
    while True:
        user = input(texto)
        if any(i.isdigit() for i in user) or any(not i.isalnum() for i in user):
            print("\nEste campo deve conter somente letras.\n")
        elif ' ' in user or len(user) == 0:
            print("\nEste campo não deve conter espaços.\n")
        else:
            break
    return user
pass
```

Figura 4 - Função validar string

A primeira função serve para receber um input do utilizador, que seja “string”.

A função vai validar se existe caracteres especiais e números.

Deverá ser preenchida apenas por letras.

É utilizado também, as funções do “Python”, “.isdigit()” e “.isalnum()”, para identificar caracteres especiais e se existe números na string.

```
63
64 def input_num(texto): #receber input do utilizador, validar erro do int
65     while True:
66         num_user = input(texto)
67         try:
68             num = int(num_user)
69             break
70         except:
71             print("\nEste campo deve conter apenas números inteiros.\n")
72     return num
73     pass
```

Figura 5 - Função validar int

A segunda função recebe o input do utilizador de tipo string, depois faz passa para uma variável “int” e faz a validação.



```
48
49 def input_idade(texto): #receber input do utilizador, validar idade entre 18 a 100 e erro do int
50     while True:
51         idade_user = input(texto)
52         try:
53             idade = int(idade_user)
54             if idade >= 18 and idade <= 100:
55                 break
56             else:
57                 print("\nIdade deve estar entre 18 e 100.\n")
58         except:
59             print("\nEste campo deve conter apenas números inteiros.\n")
60     return idade
61     pass
62
```

Figura 6 - Função idade

A terceira função assim como as anteriores recebe um input e faz as validações. Entretanto, neste caso tem uma validação acrescentada, sendo que a idade deverá estar entre os 18 e os 100 anos, parâmetro proposto pelo o desenvolvedor para controlar minimamente discrepâncias.



Funções em classes

Para cada classe criei uma função para mostrar os valores dentro o objeto, seguindo a formatação do enunciado

```
def mostrarInfo(self, lista):..mostrar info dos alunos existentes

    print("Alunos existentes:\n")

    contador = 0

    if len(lista) > 0:
        for aluno in lista:
            print(contador + 1, " - ", aluno.nome_aluno, aluno.apelido_aluno, "| N°", aluno.num_aluno)
            print("")
            contador += 1
        pass
    else:
        print(f"Tem {len(lista)} alunos neste momento.\n")
```

Figura 7 - Função Classe Info

Na “class_disciplina” criei uma função para apagar todos os alunos que estiverem inscritos nas disciplinas. É passado para a função a posição do aluno escolhido pelo o utilizador a ser apagado.

```
def apagar(self, x):..apagar alunos de todas as disciplinas que estiverem inscritos

    for a in self.listaAlunos:
        self.listaAlunos.pop(x)
        break
    pass
```

Figura 8 - Função Classe apagar



Base de dados

No ficheiro “BD” é criado e feita a conexão.

Os dados são enviados para a base de dados utilizando “queries” pelo o programa fazendo importação da conexão e o cursor. Sendo ambos os ficheiros executados em simultâneo.

```
db = mariadb.connect(  
    user="root",  
    password="",  
    host="localhost",  
    port="3306",  
    database=""  
)  
c = db.cursor() # c = cursor
```

Figura 9 - Conexão Base de dados

```
disc = Disciplina() # disc = disciplina  
disc.preencherInfo(nome_disc)  
lista_discs.append(disc)  
  
query = f"insert into disciplinas (nome) values ({nome_disc})"  
c.execute(query)  
db.commit()  
  
aluno = Aluno()  
aluno.preencherInfo(nome_aluno, apelido_aluno, idade_aluno, morada_aluno, cc_aluno, num_aluno)  
lista_alunos.append(aluno)  
  
query = f"insert into alunos (nome, apelido_aluno, num_aluno) values ({nome_aluno}, {apelido_aluno}, {num_aluno})"  
c.execute(query)  
db.commit()  
  
prof = Professor() # prof = professor  
prof.preencherInfo(nome_prof, apelido_prof, idade_prof, morada_prof, catprofi_prof, anosexp_prof, num_prof)  
lista_profs.append(prof)  
  
query = f"insert into professores (nome, apelido_prof, num_prof) values ({nome_prof}, {apelido_prof}, {num_prof})"  
c.execute(query)  
db.commit()  
pass
```

Figura 10 – Query



Funções Python

Para apagar usei “.isalpha()” e “.isnumeric()”, assim o utilizador apenas tem de inserir o número “index” ou o nome do aluno e o programa vai ler caso seja string executa uma parte do código, caso seja int, o programa identifica e executa a segunda codição.

```
if nomeAluno.isalpha(): #caso seja string, executa abaixo
    for aluno in lista_alunos:
        if nomeAluno == aluno.nome_aluno: #valida se os nomes são iguais
            try:
                lista_alunos.pop(i)

                for inscrito in lista_discs: #validar e apagar alunos de todas as disciplinas
                    inscrito.apagar(i)

                print(f"\nAluno(a) {aluno.nome_aluno} eliminado.\n")

                aluno.mostrarInfo(lista_alunos)

                query = f"delete from alunos where nome = '{nomeAluno}'"
                c.execute(query)
                db.commit()
                voltar()
                break

            except:
                print("Erro")
        i += 1
    else:
        print(f"\nAluno(a) {nomeAluno} não existe no programa.\n")
    pass
```

Figura 11 - Função Python

```
elif nomeAluno.isnumeric(): #caso seja int, executa abaixo
    try:
        numAluno = int(nomeAluno)

        for alunos in lista_alunos:
            try:
                if lista_alunos[numAluno - 1].nome_aluno == alunos.nome_aluno:

                    print(f"\nAluno(a) {lista_alunos[numAluno - 1].nome_aluno} eliminado.\n")

                    query = f"delete from alunos where nome = '{lista_alunos[numAluno - 1].nome_aluno}'"
                    c.execute(query)
                    db.commit()

                    lista_alunos.pop(i)

                    for inscrito in lista_discs: #validar e apagar alunos de todas as disciplinas
                        inscrito.apagar(i)

                    aluno.mostrarInfo(lista_alunos)
                    voltar()
                    break

            except:
                print(f"\nAluno(a) n° {numAluno} não existe no programa.\n")
                break
        i += 1
```

Figura 12 - Função Python 2