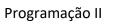


Documentação do projeto de Programação II

Instituto Politécnico da Lusófona Nº de aluno: a22100540

> Aluno: Felipe Silva Ano letivo: 2021/22

Turma: 1º ano - GSI

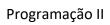






Índice de imagens

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







Índice

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



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.



Figura 1 - PyDocs



Funções

```
print("1 - Criar Disciplina")

print("2 - Listar Disciplina")

print("3 - Eliminar Disciplina")

print("4 - Criar Alung")

print("5 - Inscrever Alung")

print("6 - Listar Alung")

print("7 - Eliminar Alung")

print("8 - Listar Alungs inscrites 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 alungs de um ficheire")

print("\n0 - Sair")

pass

def menu_importar(): #opcoes do menu de importação

print("\n1 - Importar alungs de um ficheire")

print("\n1 - Importar alungs de um ficheire")

print("\n1 - Importar alungs para uma disciplinas, alungs e professores)\n")

print("\n1 - Importar alungs para uma disciplina\n")
```

Figura 2 - Menu

Crie duas funções apenas para fazer o print da opões 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, pressionado a tecla "Enter".

Está a "correr" dentro de um ciclo while, o utilizador por engano pressione 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 especias e se existe números na string.

```
def input_num(texto): #receber input do utilizador, validar erro do int

while True:

num_user = input(texto)

try:

num = int(num_user)

break

except:

print("\nEste campo deve conter apenas números inteiros.\n")

return num

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.



```
def input_idade(texto): #receber inpput do utilizador, validar idade entre 18 a 100 e erro do int

while True:
    idade_user = input(texto)
    try:
    idade = int(idade_user)
    if idade >= 18 and idade <= 100:
        break
    else:
        print("\nIdade deve estar entre 18 e 100.\n")
    except:
    print("\nEste campo deve conter apenas números inteiros.\n")
    return idade
pass</pre>
```

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 escolhio 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 utilizado "queries" pelo o programa fazendo importação da conexão e o cursor. Sendo ambos os ficheiros executados em simultâneo.

Figura 9 - Conexão Base de dados

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 seia 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

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

Figura 11 - Função Python

Figura 12 - Função Python 2