



**INSTITUTO  
FEDERAL**

Sergipe

Campus Aracaju

**Coordenadoria do Curso Superior em Análise e  
Desenvolvimento de Sistemas**

**GESTÃO DE FILA POR ORDEM DE CHEGADA**

FELIPE SOUZA VIEIRA  
ADRIANO SILVA SANTOS

Aracaju  
2021

FELIPE SOUZA VIEIRA  
ADRIANO SILVA SANTOS

## **GESTÃO DE SENHAS**

Trabalho apresentado como conclusão da disciplina Estrutura de Dados do Curso Superior em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de Sergipe.

Professor: Marcus Aurelius

Aracaju  
2021

## 1. INTRODUÇÃO

Software escrito em Python para a 2ª Avaliação da disciplina Estruturas de Dados.

## 2. ESTUDO DE CASO

Trata-se de um software de gestão de fila por ordem de chegada com suporte a fila preferencial e módulos individuais para os usuários e os atendentes.

## 3. CÓDIGO-FONTE

### 3.1. CÓDIGO SUPLEMENTAR

O programa usou dois arquivos complementares:

1. limpa\_tela.py:

```
# Método limpar a tela em qualquer sistema Windows ou Unix
# Desenvolvido por Felipe Souza Vieira e Adriano Silva Santos com base
# no exemplo:
# https://stackoverflow.com/questions/2084508/clear-terminal-in-python

def limpa():
    import os
    os.system('cls' if os.name == 'nt' else 'clear')

# Depois de importar basta chamar o método
# limpa()
```

## 2. menus.py:

```

# Felipe Souza Vieira e Adriano Silva Santos
# Contém partes do sw Financial Management

# Menus
def cabecalho():
    print("-" * 39)
    print("|", " " * 10, " ATENDIMENTO ", " " * 10, '|')
    print("-" * 39)
def menuUsuario():
    print("\nEscolha um Atendimento:\n")
    print(" --> 1: Atendimento Comum")
    print(" --> 2: Atendimento Prioritário")
    print(" --> 3: Consultar Nome (em manutenção)")
    print("\n")
    print(" --> 0: Mudar Perfil")
    print("_" * 39, '\n')
def telaUsuario():
    from limpa_tela import limpa
    limpa()
    cabecalho()
    menuUsuario()
def menuAtendente():
    print("Atendimento:\n")
    # print("Próximas senhas: {}".format(senha))
    print(" --- 1: Chamar o próximo da fila")
    print(" --- 2: Mostrar fila")
    print("\n")
    print(" --- 0: Mudar Perfil")
    print("_" * 39, '\n')
def telaAtendente():
    from limpa_tela import limpa
    limpa()
    cabecalho()
    menuAtendente()

```

### 3.2 CÓDIGO FONTE PRINCIPAL

```
# 2ª Prova de Estrutura de Dados
# Felipe Souza Vieira e Adriano Silva Santos
# Gestão de fila por ordem de chegada

from menus import telaUsuario, telaAtendente
from limpa_tela import limpa

# Programa Principal:
#####

def main():
    limpa()
    print("\n 1. Atendente\n 2. Usuário")
    opcao = None
    opcao = int(input("--> "))
    if opcao == 1:
        programaAtendente()
    elif opcao == 2:
        programaUsuario()
    else:
        main()

# Programas do Atendente:
#####

def programaAtendente():
    telaAtendente()
    opcao = None
    opcao = int(input("--> "))
    if opcao == 1:
        chamaProximo()
        input("Enter para continuar")
    elif opcao == 2:
        telaAtendente()
        todaFila()
```

```

elif opcao == 0:
    main()
else:
    programaAtendente()

# Chama o próximo usuário respeitando a fila de prioridade
def chamaProximo():
    if senhaPrioritaria != None:
        print("Chamando {}".format(senhaPrioritaria[0]))
        senhaPrioritaria.pop(0)
    else:
        print("Chamando {}".format(senhaComum[0]))
        senhaComum.pop(0)

def todaFila():
    print("Prioritários em ordem: ",senhaPrioritaria, end=", ")
    print(" Demais em ordem: ",senhaComum)
    print()
    input('[ENTER para continuar]')
    programaAtendente()

# Programas do Usuário:
#####

def programaUsuario():
    telaUsuario()
    opcao = None
    opcao = int(input("--> "))
    if opcao == 1:
        telaUsuario()
        emiteSenhaComum()
        input('[ENTER] para continuar')
        programaUsuario()
    elif opcao == 2:
        telaUsuario()
        emiteSenhaPrioritaria()

```

```

    input('[ENTER] para continuar')
    programaUsuario()
elif opcao == 3:
    buscaPessoa()
elif opcao == 0:
    main()
else:
    telaUsuario()
    print(" Não existe opção {}, [ENTER] para tentar novamente:".format(opcao))
    input()
    programaUsuario()

# Busca alguém na fila
def buscaPessoa():
    telaUsuario()
    print()
    # Ainda tentando desenvolver essa joça
    # pessoa = str(input("Digite o nome: "))
    # max = (len(senhaComum) + len(senhaPrioritaria))
    # i = 0
    # while len(senhaComum) > 0:
    #     if pessoa == senhaComum[i]:
    #         print("{} é a {}ª pessoa da fila.".format(senhaComum[i], i))
    #         i += 1
    # while len(senhaPrioritaria) > 0:
    #     if pessoa == senhaPrioritaria[i]:
    #         print("{} é a {}ª pessoa da fila.".format(senhaPrioritaria[i], i))
    #         i += 1
    input("Opção em manutenção. [ENTER] para voltar")
    programaUsuario()

# Usuário entra na fila e vê quantas pessoas faltam
def emiteSenhaComum():
    senhaComum.append(str(input('Insira seu nome completo: ')))

```

```
    print("Você está em {}º na fila.".format((len(senhaComum) +
len(senhaPrioritaria))), end="")
def emiteSenhaPrioritaria():
    senhaPrioritaria.append(str(input("Insira seu nome completo: ")))
    print("Você está em {}º na fila.".format((len(senhaPrioritaria))), end="")

# Inicializa as Filas e o Programa
#####

senhaComum = []
senhaPrioritaria = []
main()
```



## 6. REFERÊNCIAS BIBLIOGRÁFICAS

1. Árvores binárias, método `def __str__(self, data):`:  
<https://www.youtube.com/watch?v=6E169kShoNU>