



# Entrega do Portfolio do modulo Computational Logic Using Python

Por: Marco Antonio Monteiro Pedro

## **CONTROLE DE EVENTOS DA UNIFECAF**

### **FUNCIONALIDADES PRINCIPAIS**

O sistema de Controle de Eventos da UniFECAF é uma aplicação interativa desenvolvida em Python que permite gerenciar eventos acadêmicos de maneira eficiente. O sistema oferece uma interface simples e intuitiva para que alunos e coordenadores possam visualizar, criar, editar, excluir e se inscrever em eventos.

## SUMÁRIO

|   |    |
|---|----|
| Funcionalidades Principais .....                                | 3  |
| Visualizar Eventos.....   | 3  |
| Criar Eventos .....   | 3  |
| Editar Eventos.....   | 3  |
| Excluir Eventos .....   | 3  |
| Visualizar Inscritos .....                                      | 3  |
| Inscrever Aluno .....   | 3  |
| Funcionamento do Sistema .....                                  | 3  |
| Menu Principal .....  | 3  |
| Área do Aluno .....   | 4  |
| Área da Coordenação .....                                       | 4  |
| Fluxo de Trabalho .....   | 4  |
| Exemplo de Fluxo.....   | 4  |
| Início .....  | 4  |
| Área do Coordenador.....  | 4  |
| Área do Aluno .....   | 4  |
| Código Fonte: .....   | 5  |
| Codigo Fonte usado: .....                                       | 5  |
| Sobre github.....   | 13 |
| Manual de Instrução para o Controle de Eventos da UniFECAF..... | 13 |
| Introdução .....  | 13 |
| Menu Principal .....  | 13 |
| Escolha uma Opção.....  | 13 |
| Área do Aluno .....   | 13 |
| Criar um Novo Evento .....                                      | 14 |
| Editar um Evento Existente .....                                | 14 |
| Excluir um Evento .....   | 14 |
| Visualizar Inscritos em um Evento.....                          | 14 |
| Visualizar Eventos.....   | 14 |
| Funcionalidades de Validação.....                               | 15 |
| Conclusão .....   | 15 |



# Funcionalidades Principais

## VISUALIZAR EVENTOS

- Permite que usuários (alunos e coordenadores) visualizem todos os eventos cadastrados.
- Exibe informações detalhadas sobre cada evento, incluindo a quantidade de pessoas permitidas, data, hora e a lista de inscritos.

## CRIAR EVENTOS

- Disponível para coordenadores.
- Permite que novos eventos sejam criados com nome, quantidade de pessoas permitidas, data e hora.
- Os eventos criados são armazenados no dicionário evento com seus detalhes.

## EDITAR EVENTOS

- Disponível para coordenadores.
- Permite editar os detalhes de um evento existente, incluindo a quantidade de pessoas permitidas, data e hora.
- A edição é realizada de maneira segura, tratando entradas inválidas.

## EXCLUIR EVENTOS

- Disponível para coordenadores.
- Permite que eventos sejam excluídos do sistema.
- A exclusão remove o evento do dicionário evento.

## VISUALIZAR INSCRITOS

- Disponível para coordenadores.
- Permite visualizar a lista de alunos inscritos em um evento específico.
- O coordenador pode optar por retornar ao menu principal após visualizar os inscritos.

## INSCREVER ALUNO

- Disponível para alunos.
- Permite que alunos se inscrevam em eventos disponíveis.
- Verifica a disponibilidade de vagas e permite a inscrição de múltiplos alunos até que todas as vagas sejam preenchidas.

# Funcionamento do Sistema

## MENU PRINCIPAL

- O sistema inicia exibindo o menu principal, onde o usuário pode escolher entre a área do aluno ou a área da coordenação.
- As opções disponíveis no menu principal são:
  - (1) Área do Aluno (inscrever em eventos)

- (2) Área da Coordenação
- (3) Visualizar Eventos
- (4) Sair

#### ÁREA DO ALUNO

- Alunos podem visualizar eventos disponíveis e se inscrever em eventos com vagas abertas.

#### ÁREA DA COORDENAÇÃO

- Coordenadores têm acesso a funcionalidades adicionais, como criar, editar e excluir eventos, além de visualizar inscritos.

#### FLUXO DE TRABALHO

- As funções são chamadas de acordo com as opções selecionadas pelo usuário.
- O sistema trata entradas inválidas e fornece mensagens de erro amigáveis para melhorar a experiência do usuário.

## Exemplo de Fluxo

#### INÍCIO

- O sistema exibe o menu principal.
- Usuário escolhe a opção desejada.

#### ÁREA DO COORDENADOR

- Coordenador escolhe "Criar Evento".
- O sistema solicita os detalhes do novo evento e os armazena no dicionário evento.
- Coordenador pode optar por criar mais eventos ou voltar ao menu principal.

#### ÁREA DO ALUNO

- Aluno escolhe "Inscrever em Eventos".
- O sistema exibe eventos disponíveis e o status das vagas.
- Aluno insere seu nome para se inscrever no evento selecionado.
- O sistema confirma a inscrição e permite adicionar mais alunos até que as vagas estejam completas.

## Código Fonte:

### CODIGO FONTE USADO:

```
# Importar biblioteca de temporizador
import time

# Dicionário para armazenar os eventos
evento = {}

# Função para visualizar todos os eventos cadastrados
def visualizar_eventos():
    print("Eventos Atuais:")

    if not evento:
        print("Nenhum evento cadastrado.") # Exibe mensagem se não houver eventos
    else:
        for nome, detalhes in evento.items():
            status_vagas = "Lotado" if len(detalhes['Inscritos']) >= detalhes['Quantidade de pessoas permitidas'] else "Vagas Disponíveis"

            print(f"Evento: {nome}")
            print(f"  Descrição: {detalhes['Descrição']}")
            print(f"  Quantidade de pessoas permitidas: {detalhes['Quantidade de pessoas permitidas']}")
            print(f"  Data: {detalhes['Data marcada']}")
            print(f"  Hora: {detalhes['Hora p/ inicio do Evento']}")
            print(f"  Inscritos: {' , '.join(detalhes['Inscritos']) if detalhes['Inscritos'] else 'Nenhum inscrito'}")
            print(f"  Status: {status_vagas}")
```

```

        print("\n")

time.sleep(2)

# Função para criar um novo evento
def criar_evento():
    while True:
        try:
            Evento_name = input('Digite o nome do evento: ')
            Descricao = input('Digite a descrição do evento: ')
            Quant_Pess_Perm = int(input('Quantidade de pessoas permitidas nesse evento: '))
            Data_Event = input('Informe a data do evento: ')
            Hora_Event = input('Informe a hora do evento: ')

            # Armazenar os detalhes do evento no dicionário
            evento[Evento_name] = {
                'Descrição': Descricao,
                'Quantidade de pessoas permitidas': Quant_Pess_Perm,
                'Data marcada': Data_Event,
                'Hora p/ inicio do Evento': Hora_Event,
                'Inscritos': []
            }
            print(f"Evento {Evento_name} marcado com sucesso.\n")

            # Perguntar se deseja adicionar outro evento
            adicionar_mais = input('Deseja adicionar outro evento? (S/N): ')
            if adicionar_mais.lower() != 's':
                break
        except ValueError:
            print("Entrada inválida. Por favor, insira valores corretos.\n")

```

```
time.sleep(2)
```

```
# Função para editar os detalhes de um evento existente
```

```
def editar_evento():
```

```
    if not evento:
```

```
        print("Nenhum evento disponível para edição.")
```

```
        return
```

```
    Evento_name = input('Digite o nome do evento que deseja editar: ')
```

```
    if Evento_name in evento:
```

```
        try:
```

```
            # Tenta executar o código abaixo, que pode causar erro se os dados de entrada forem inválidos
```

```
            print(f"Editando evento {Evento_name}")
```

```
            Descricao = input('Nova descrição do evento: ')
```

```
            Quant_Pess_Perm = int(input('Nova quantidade de pessoas permitidas: '))
```

```
            Data_Event = input('Nova data do evento: ')
```

```
            Hora_Event = input('Nova hora do evento: ')
```

```
        # Atualizar os detalhes do evento no dicionário
```

```
        evento[Evento_name] = {
```

```
            'Descrição': Descricao,
```

```
            'Quantidade de pessoas permitidas': Quant_Pess_Perm,
```

```
            'Data marcada': Data_Event,
```

```
            'Hora p/ início do Evento': Hora_Event,
```

```
            'Inscritos': evento[Evento_name]['Inscritos']
```

```
        }
```

```
        print(f"Evento {Evento_name} atualizado com sucesso.\n")
```

```
    except ValueError:
```



```
# Exibe mensagem de erro se os dados de entrada forem inválidos

print("Entrada inválida. Por favor, insira valores corretos.\n")

else:

    print(f"O evento {Evento_name} não existe.\n")

time.sleep(2)


# Função para excluir um evento existente

def excluir_evento():

    if not evento:

        print("Nenhum evento disponível para exclusão.")

        return

    visualizar_eventos()

    Evento_name = input('Digite o nome do evento que deseja excluir: ')

    if Evento_name in evento:

        del evento[Evento_name] # Remover o evento do dicionário

        print(f"Evento {Evento_name} excluído com sucesso.\n")

    else:

        print(f"O evento {Evento_name} não existe.\n")

    time.sleep(2)


# Função para visualizar os inscritos em um evento específico

def visualizar_inscritos():

    if not evento:

        print("Nenhum evento disponível.")

        return

    print("Eventos Disponíveis:")

    for nome in evento.keys():
```

```

    print(f"- {nome}")

print("\n")

Evento_name = input('Digite o nome do evento para visualizar os inscritos: ')

if Evento_name in evento:

    inscritos = evento[Evento_name]['Inscritos']

    if inscritos:

        print(f"Inscritos no evento {Evento_name}:")

        for aluno in inscritos:

            print(aluno)

# Loop para perguntar se deseja voltar ao menu principal

while True:

    voltar = input('Deseja voltar ao menu principal? (S/N): ')

    if voltar.lower() == 's':

        break

    elif voltar.lower() == 'n':

        pass

    else:

        print("Opção inválida. Tente novamente.\n")

else:

    print(f"Ninguém está cadastrado no evento {Evento_name}.")

else:

    print(f"O evento {Evento_name} não existe.\n")

time.sleep(2)

# Função para inscrever alunos em um evento

def inscrever_aluno():

    if not evento:

```

```

print("Nenhum evento disponível para inscrição.")

return

# Exibir a lista de eventos disponíveis
print("Eventos Disponíveis:")

for nome, detalhes in evento.items():

    # Verificar se o número de inscritos é maior ou igual ao número de vagas permitidas
    if len(detalhes['Inscritos']) >= detalhes['Quantidade de pessoas permitidas']:

        status_vagas = "Completo, nao tem vagas disponiveis"

    else:

        status_vagas = "Vagas Disponíveis"

    # Exibir o nome do evento e seu status de vagas
    print(f"Evento: {nome}, Status: {status_vagas}")

print("\n")

Evento_name = input('Digite o nome do evento que deseja se inscrever: ')

if Evento_name in evento:

    while len(evento[Evento_name]['Inscritos']) < evento[Evento_name]['Quantidade de pessoas permitidas']:

        Aluno_nome = input('Digite seu nome: ')

        evento[Evento_name]['Inscritos'].append(Aluno_nome)

        print(f"{Aluno_nome} inscrito com sucesso no evento {Evento_name}.\n")

    # Verificar se o evento atingiu o número máximo de inscritos
    if len(evento[Evento_name]['Inscritos']) >= evento[Evento_name]['Quantidade de pessoas permitidas']:

        print(f"O evento {Evento_name} atingiu o número máximo de inscritos.\n")

        break

# Perguntar se deseja adicionar outra pessoa ao evento

```

```

    adicionar_mais = input('Deseja adicionar outra pessoa a este evento? (S/N): ')

    if adicionar_mais.lower() != 's':

        break

else:

    print(f"O evento {Evento_name} não existe.\n")

    time.sleep(2)

# Função para o menu do coordenador, oferecendo várias opções de gerenciamento de eventos
def menu_cordenador():

    while True:

        print('Escolha a opção: \n\n (1) Visualizar Eventos\n (2) Criar Eventos\n (3) Editar Eventos\n (4) Excluir Eventos\n (5) Visualizar Inscritos\n (6) Voltar ao menu principal\n')

        try:

            Opcao_Cordenador = int(input("Digite a opção desejada: "))

            if Opcao_Cordenador == 1:

                visualizar_eventos()

            elif Opcao_Cordenador == 2:

                criar_evento()

            elif Opcao_Cordenador == 3:

                # Chamar a função editar_evento diretamente sem perguntar se deseja voltar ao menu principal

                visualizar_eventos()

                editar_evento()

            elif Opcao_Cordenador == 4:

                excluir_evento()

            elif Opcao_Cordenador == 5:

                visualizar_inscritos()

            elif Opcao_Cordenador == 6:

                break

        else:

```

```

        print("Opção inválida. Tente novamente.\n")

except ValueError:

    print("Entrada inválida. Por favor, digite um número válido.\n")

time.sleep(2)

# Função para o menu principal, permitindo escolher entre a área do aluno e a área da coordenação
def menu_principal():

    while True:

        print('***** Controle de Eventos da UniFECAF *****\n')

        print('Escolha a opção respeitando o critério de Aluno e Coordenador:\n')

        print('(1) Área do Aluno (inscrever em eventos) \n(2) Área da Coordenação\n(3) Visualizar  
Eventos\n(4) Sair\n')

        try:

            Escolha_Usuario = int(input("Digite a opção desejada: "))

            if Escolha_Usuario == 1:

                inscrever_aluno()

            elif Escolha_Usuario == 2:

                menu_cordenador()

            elif Escolha_Usuario == 3:

                visualizar_eventos()

                if not evento:

                    time.sleep(2)

            elif Escolha_Usuario == 4:

                print("Saindo do sistema...")

                break

            else:

                print("Opção inválida. Tente novamente.\n")

        except ValueError:

            print("Entrada inválida. Por favor, digite um número válido.\n")

```

```
time.sleep(2)
```

```
# Iniciar o menu principal
```

```
menu_principal()
```

## **SOBRE GITHUB**

O desenvolvimento e arquivo do projeto está no GitHub e é publico.  
<https://github.com/marcosamambaia/Portfolio>

# **Manual de Instrução para o Controle de Eventos da UniFECAF**

## **INTRODUÇÃO**

Bem-vindo ao sistema de Controle de Eventos da UniFECAF. Este manual irá guiá-lo passo a passo sobre como utilizar o programa para visualizar, criar, editar, excluir e se inscrever em eventos. O sistema é projetado para ser intuitivo e fácil de usar, tanto para alunos quanto para coordenadores.

## **MENU PRINCIPAL**

Ao iniciar o programa, você verá o menu principal com as seguintes opções:

1. Área do Aluno (inscrever em eventos)
2. Área da Coordenação (gerenciar eventos)
3. Visualizar Eventos
4. Sair

## **ESCOLHA UMA OPÇÃO**

- **Para alunos:** Escolha a opção 1 para se inscrever em eventos.
- **Para coordenadores:** Escolha a opção 2 para gerenciar eventos.
- **Para visualizar eventos:** Escolha a opção 3 para ver todos os eventos cadastrados.
- **Para sair do programa:** Escolha a opção 4.

## **ÁREA DO ALUNO**

### **Inscrever-se em Eventos**

1. Escolha a opção 1 no menu principal.

2. O programa exibirá uma lista de eventos disponíveis com o status das vagas (Vagas Disponíveis ou Lotado).
3. Digite o nome do evento em que deseja se inscrever.
4. Insira seu nome para se inscrever.
5. O programa confirmará sua inscrição.

## **Área da Coordenação**

### **CRIAR UM NOVO EVENTO**

1. Escolha a opção 2 no menu principal.
2. Digite o nome do evento.
3. Digite a descrição do evento.
4. Insira a quantidade de pessoas permitidas.
5. Informe a data do evento.
6. Informe a hora do evento.
7. Confirme os detalhes para criar o evento.

### **EDITAR UM EVENTO EXISTENTE**

1. Escolha a opção 2 no menu principal.
2. Digite o nome do evento que deseja editar.
3. Atualize os detalhes do evento conforme necessário (nome, descrição, quantidade de pessoas, data e hora).
4. Confirme as alterações.

### **EXCLUIR UM EVENTO**

1. Escolha a opção 2 no menu principal.
2. Digite o nome do evento que deseja excluir.
3. Confirme a exclusão.

### **VISUALIZAR INSCRITOS EM UM EVENTO**

1. Escolha a opção 2 no menu principal.
2. Digite o nome do evento para visualizar a lista de inscritos.
3. O programa exibirá todos os alunos inscritos no evento.

### **VISUALIZAR EVENTOS**

1. Escolha a opção 3 no menu principal.
2. O programa exibirá todos os eventos cadastrados com detalhes como nome, descrição, quantidade de pessoas permitidas, data, hora e lista de inscritos.
3. O status das vagas (Vagas Disponíveis ou Lotado) também será mostrado.

## **FUNCIONALIDADES DE VALIDAÇÃO**

- O programa valida entradas para evitar erros, como nomes duplicados ou dados inválidos.
- Mensagens de erro amigáveis são fornecidas para orientar o usuário em caso de entradas incorretas.

## **CONCLUSÃO**

Este manual cobre as funcionalidades básicas e o fluxo de trabalho do sistema de Controle de Eventos da UniFECAF. Siga estas instruções para utilizar o programa de forma eficaz. Se encontrar problemas ou tiver dúvidas, entre em contato com o suporte técnico da UniFECAF.