

Sistema de Gerenciamento de Eventos

Um sistema em Python para **gerenciar eventos, participantes, pagamentos, relatórios e login**, organizado em módulos separados.

O projeto foi desenvolvido com foco em **modularidade, clareza e testes automatizados**.

Estrutura do Projeto

```
projeto/
├── main/
│   ├── sistema_eventos.py
│   ├── sistema_login.py
│   ├── sistema_pagamentos.py
│   └── sistema_relatorios.py
├── s-principal.py
├── teste_sistema.py
└── README.md
```

Arquivo/Pasta	Descrição
main/sistema_eventos.py	Define classes Evento , Participante , Workshop , Palestra e SistemaEventos . Responsável pelo gerenciamento de eventos e inscrições.
main/sistema_login.py	Módulo de autenticação e cadastro de usuários.
main/sistema_pagamentos.py	Simula o registro e listagem de pagamentos de eventos.
main/sistema_relatorios.py	Gera relatórios sobre eventos, participantes e presença.
s-principal.py	Arquivo principal que integra todos os módulos e executa o sistema.
teste_sistema.py	Testes automatizados (unitários e funcionais) com unittest .

Instalação e Execução

1. Clone o repositório:

```
git clone https://github.com/seu-usuario/projeto-eventos.git
cd projeto-eventos
```

2. Certifique-se de ter o Python 3.10+ instalado:

```
python --version
```

3. Execute o sistema principal:

```
python s-principal.py
```

Exemplo de Execução

```
=== Sistema de Gerenciamento de Eventos ===
Usuário 'andrew19@gmail.com' cadastrado com sucesso!
Login bem-sucedido para 'andrew19@gmail.com'!
Pagamento de R$20.0 registrado para Andrew no evento 'Artes Visuais'.

Andrew inscrito com sucesso no evento 'Artes Visuais'.

Lista de eventos disponíveis:
- Artes Visuais - 19/11/2025 - R$20.0
- Artes Cênicas - 20/11/2025 - R$50.0

[Relatório de Eventos]
Evento: Artes Visuais | Inscritos: 1 | Presenças: 1
Evento: Artes Cênicas | Inscritos: 0 | Presenças: 0
```

Testes Automatizados

Executar os testes:

```
python -m unittest teste_sistema.py -v
```

Saída esperada:

```
test_busca_por_categoria ... ok
test_cadastro_e_login ... ok
test_eventos_com_vagas ... ok
test_inscricao_evento ... ok
test_pagamento ... ok
test_dados_evento ... ok
```

```
-----
Ran 6 tests in 0.01s
OK
```

Funcionalidades

- Cadastro e autenticação de usuários
- Criação e gerenciamento de eventos
- Inscrição de participantes com controle de vagas
- Registro e listagem de pagamentos
- Relatórios de eventos e presenças
- Testes automatizados de todas as funções principais

Tecnologias Utilizadas

- Python 3.12
- unittest (testes automatizados)
- Paradigma de Programação Orientada a Objetos (POO)

minimapa:

