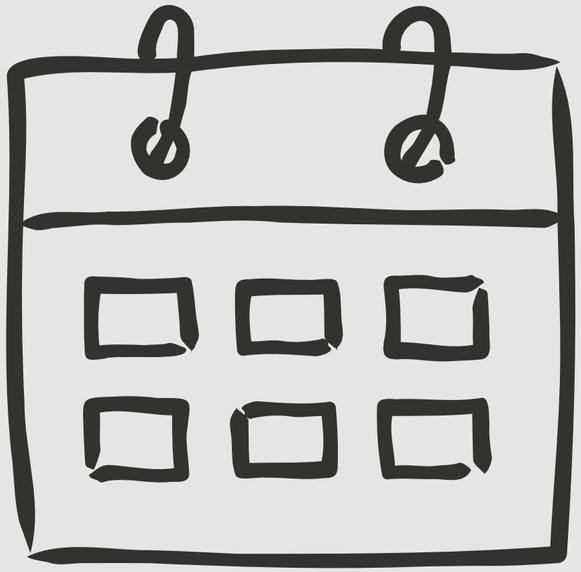
Gestão de Calendário Pessoal

Martim Amador Vasco 28694

Programação Orientada Objetos

Objetivo do Projeto



1

Desenvolver um sistema de gestão de calendário pessoal que permita gerir utilizadores, eventos e categorias, com validações e persistência de dados.

Tecnologias Utilizadas

1. Python
2. Programação Orientada a Objetos
3. UML (PlantUML)
4. JSON (persistência)
5. VS Code

Diagramas

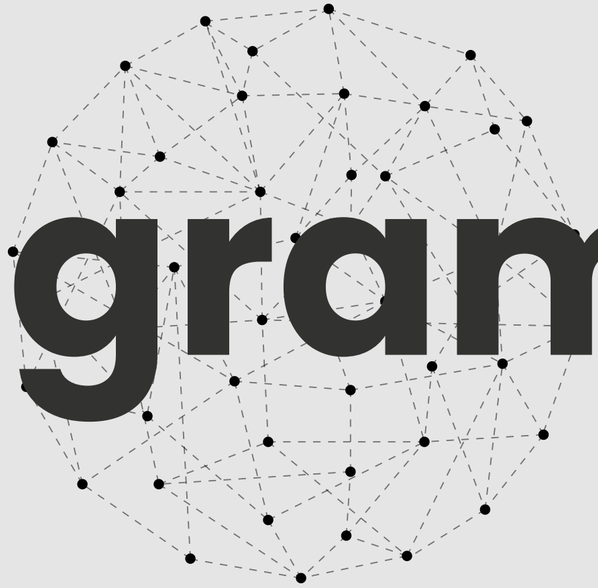


Diagrama de Classes

1

O sistema é composto pelas classes Event, Category, User e CalendarManager, responsáveis pela lógica e gestão dos eventos.

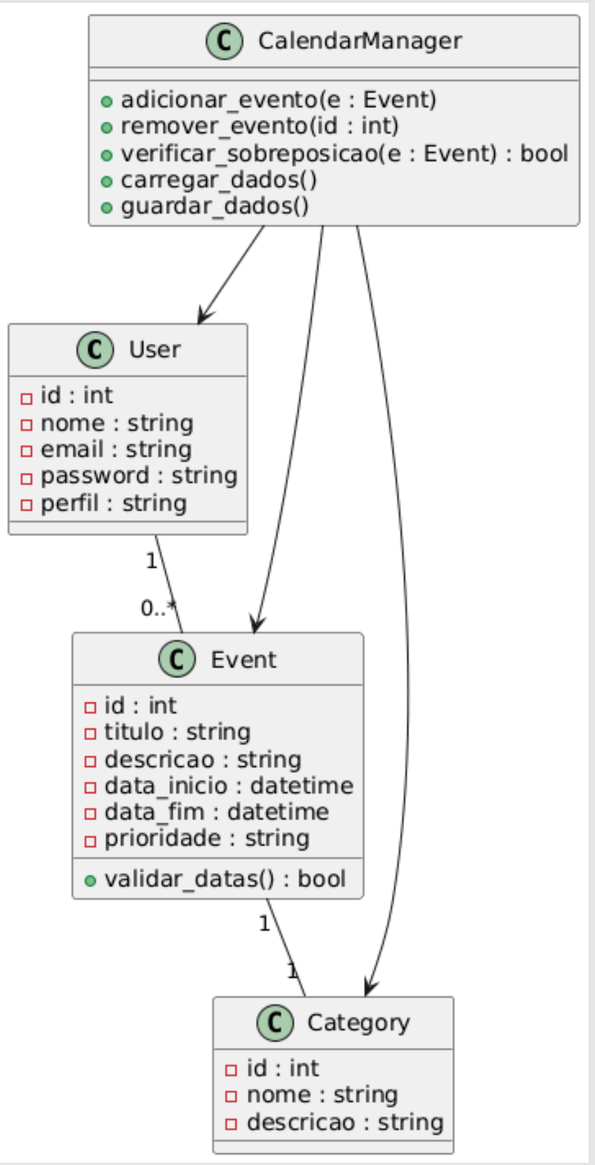


Diagrama de Casos de Uso

1

O utilizador pode adicionar, remover e visualizar eventos no sistema

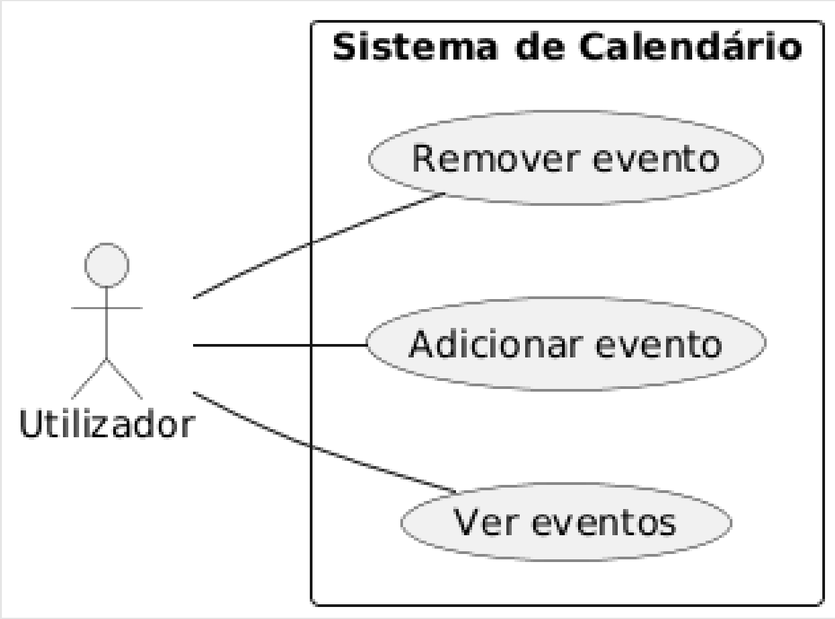
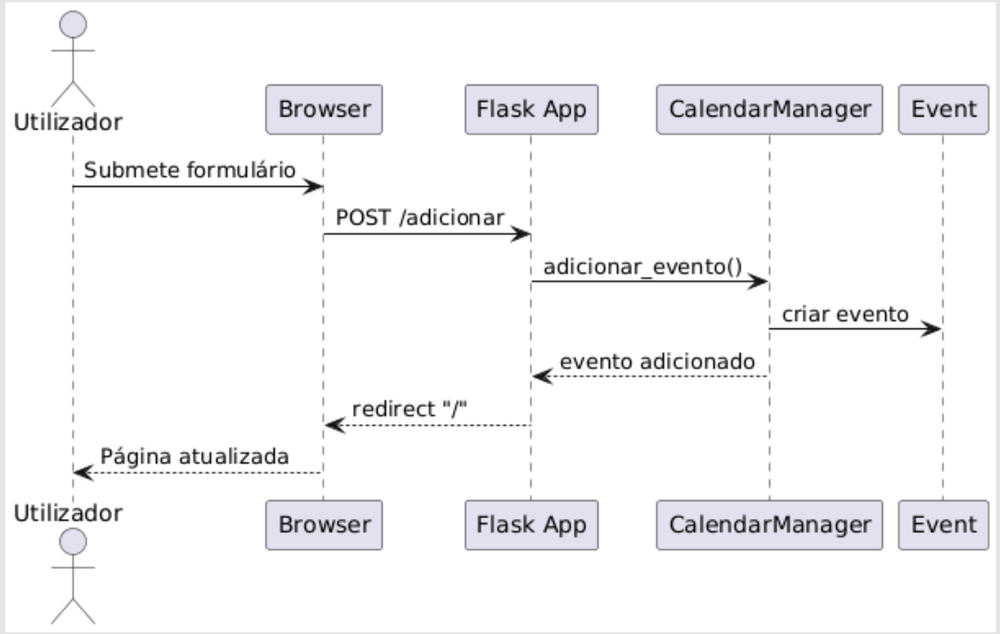


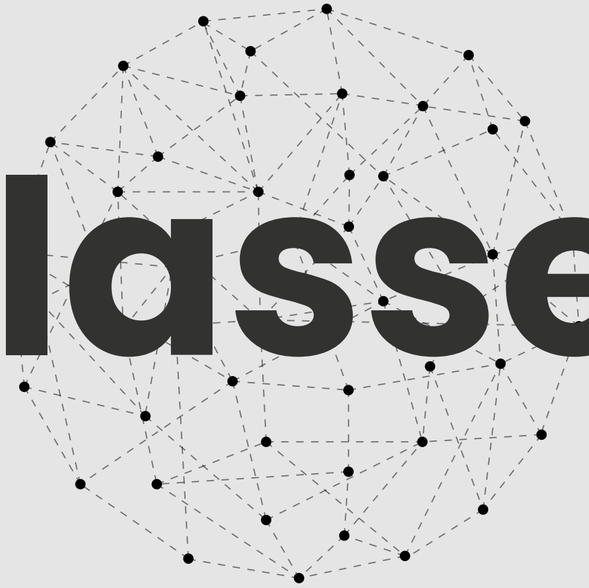
Diagrama de Sequência

1

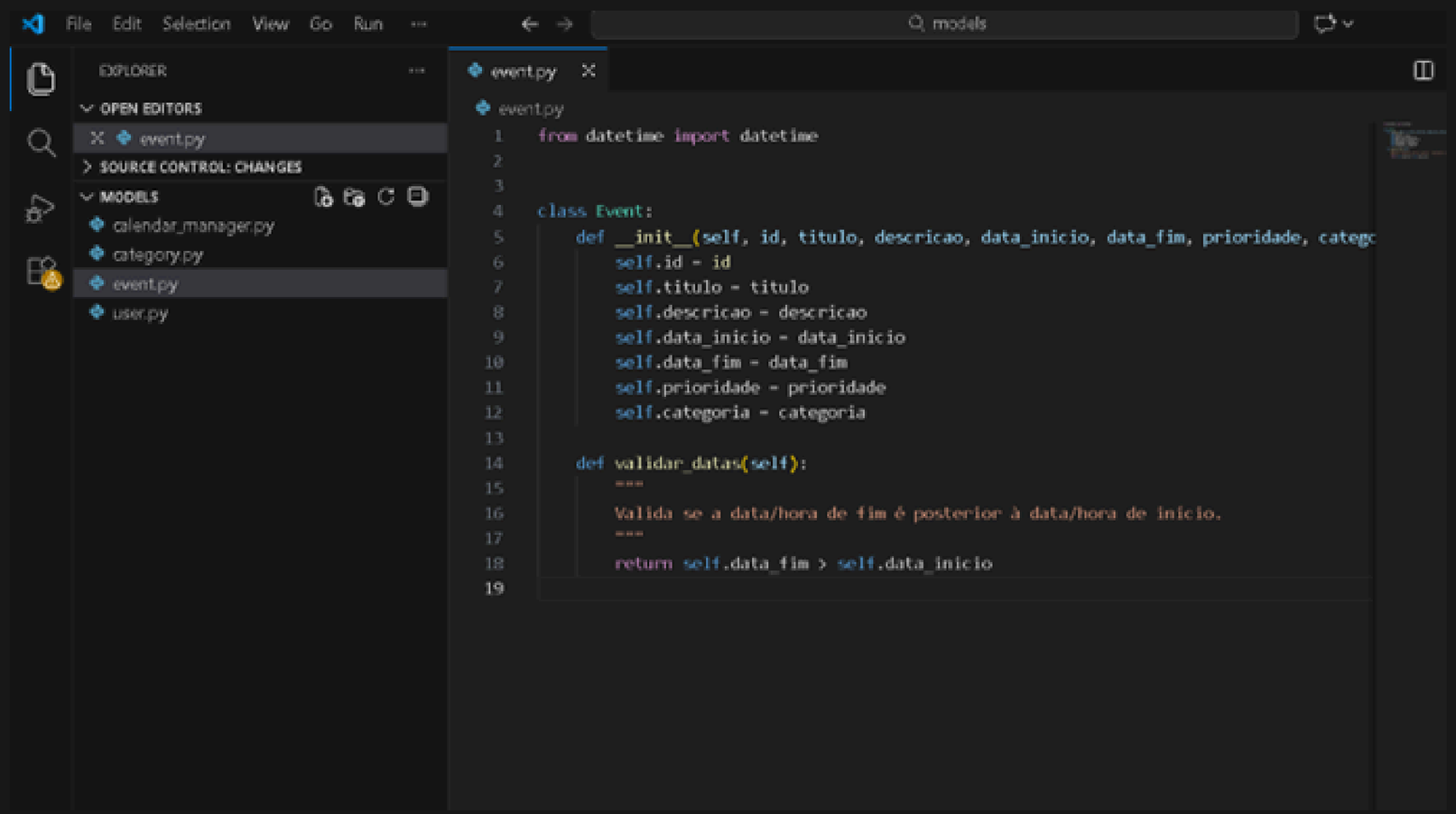
O diagrama representa o fluxo de criação de um evento desde a interface até ao gestor do calendário



Classes

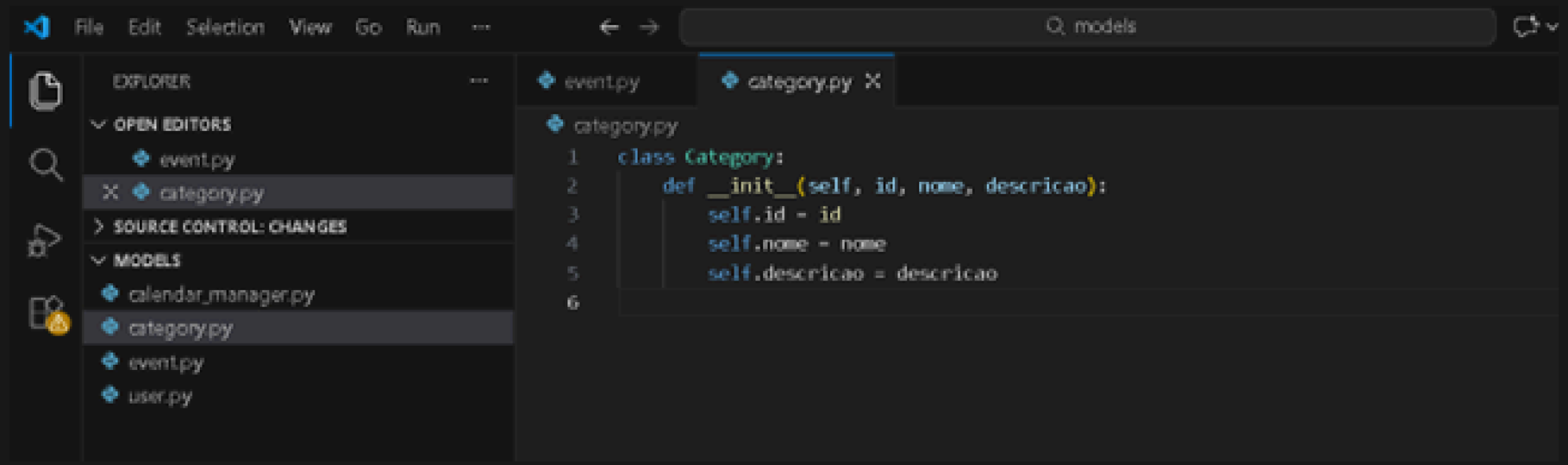


ClasseEvent



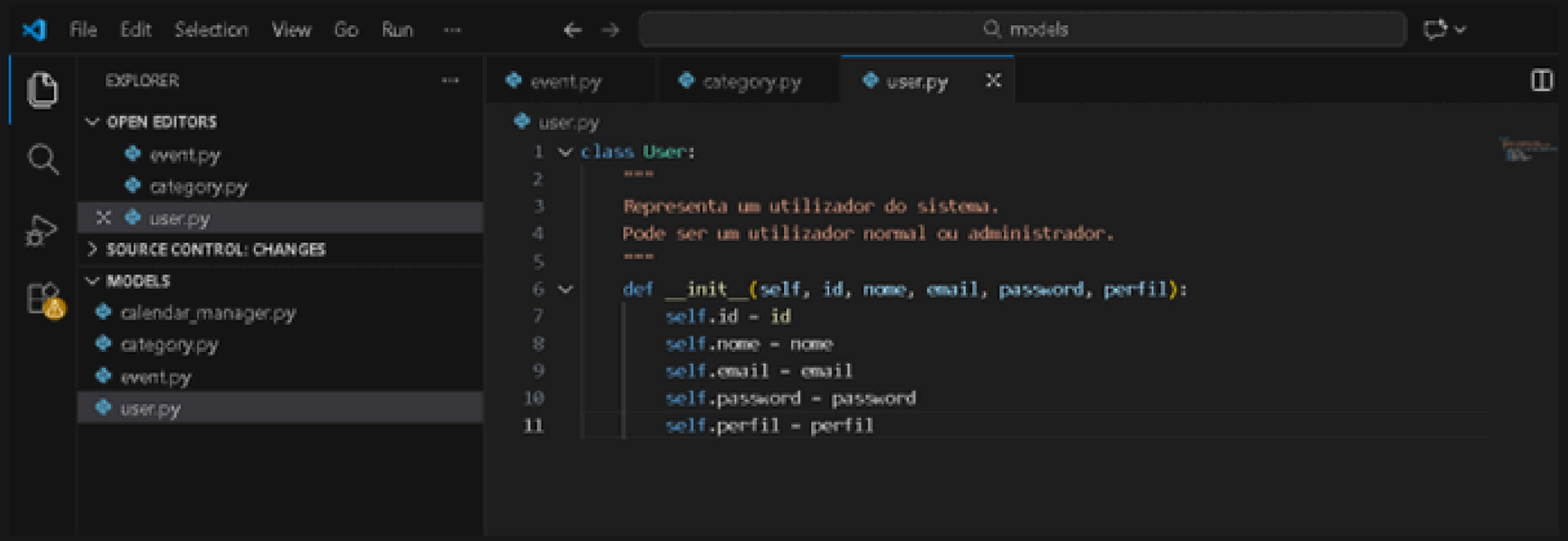
A classe Event representa um evento do calendário e valida as datas de início e fim

Classe Category



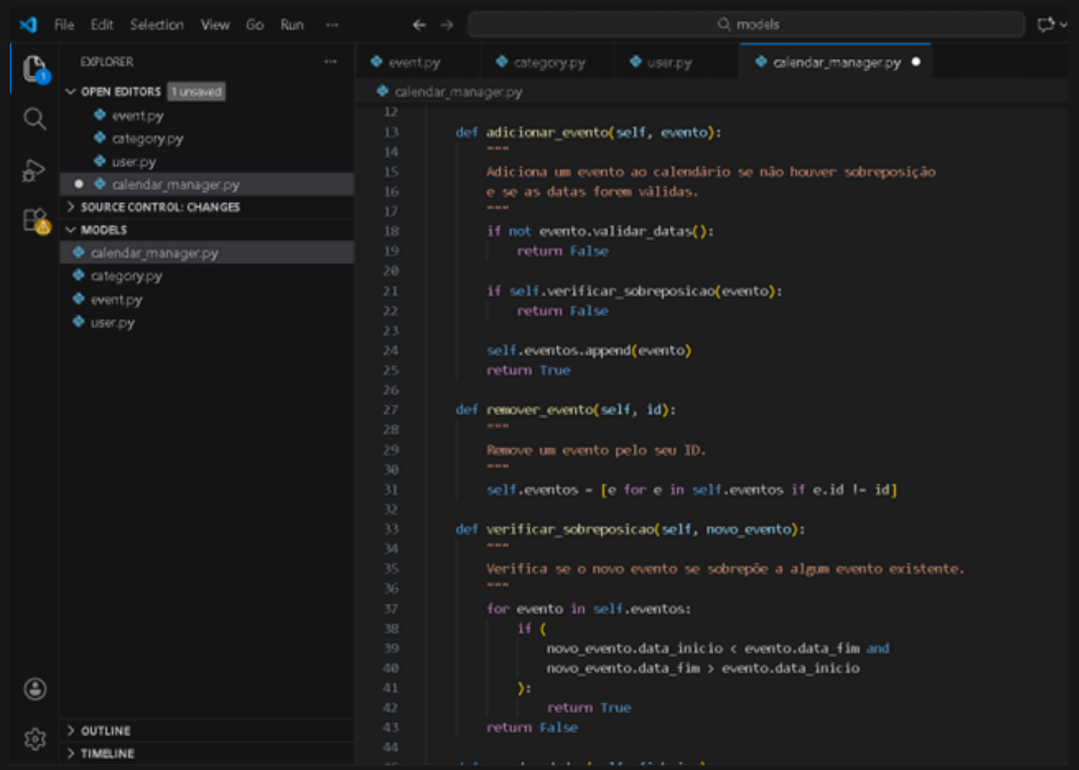
A classe Category permite classificar eventos sem conter regras de negócio próprias

Classe User



Representa os utilizadores do sistema, distinguindo perfis.

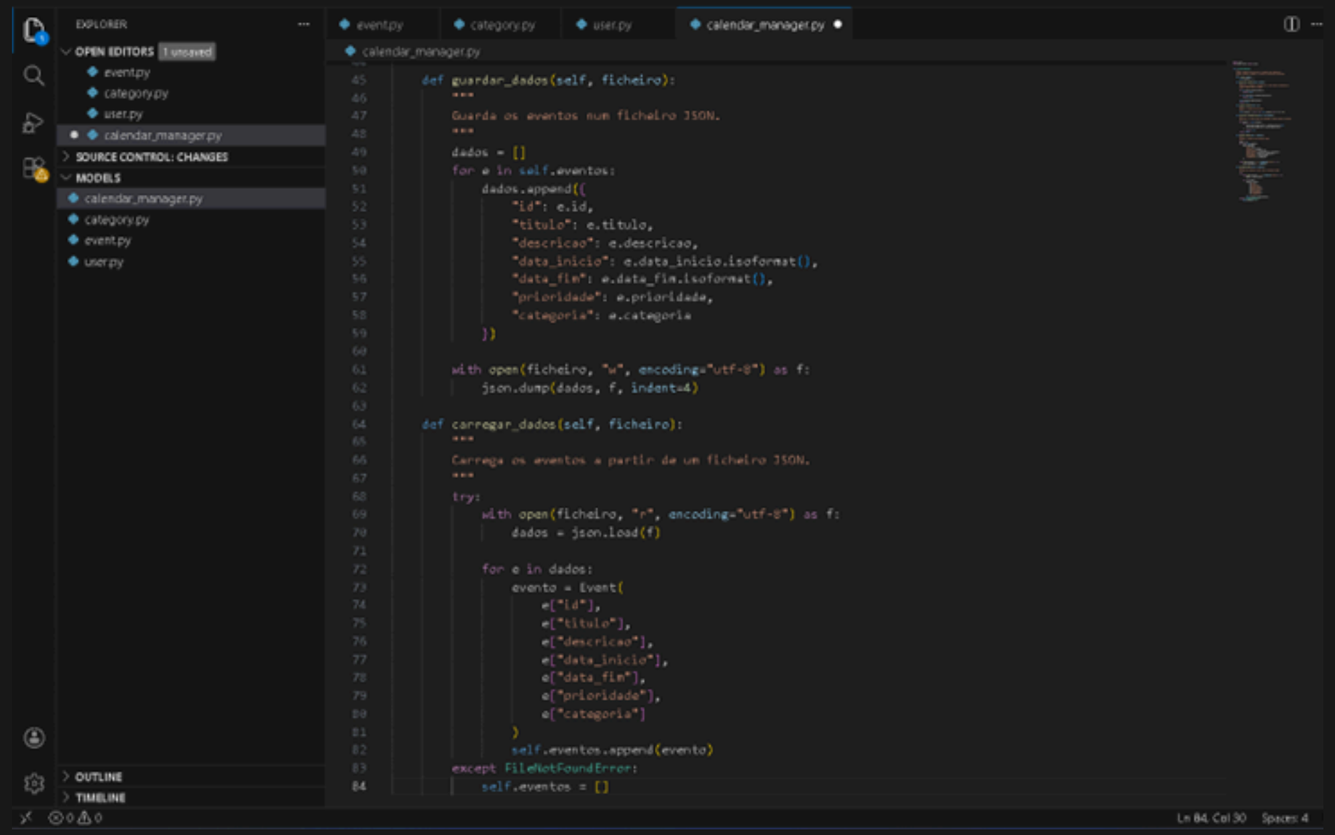
Classe CalendarManager



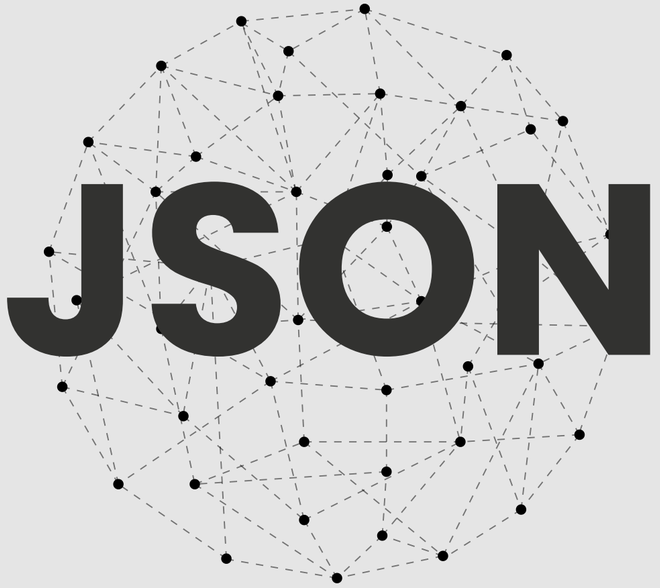
- adicionar evento
- verificar sobreposição

Centraliza as regras de negócio, validações e gestão de eventos.

Classe CalendarManager



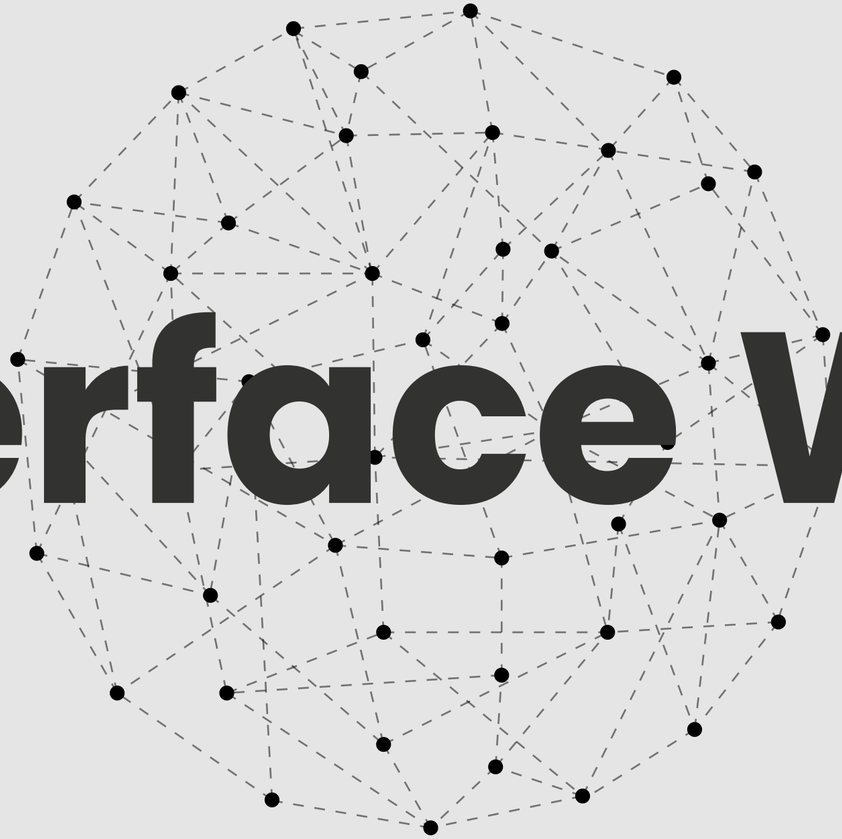
Guardar eventos e Carrega os eventos a partir de um ficheiro JSON


JSON

Persistência de dados (JSON)

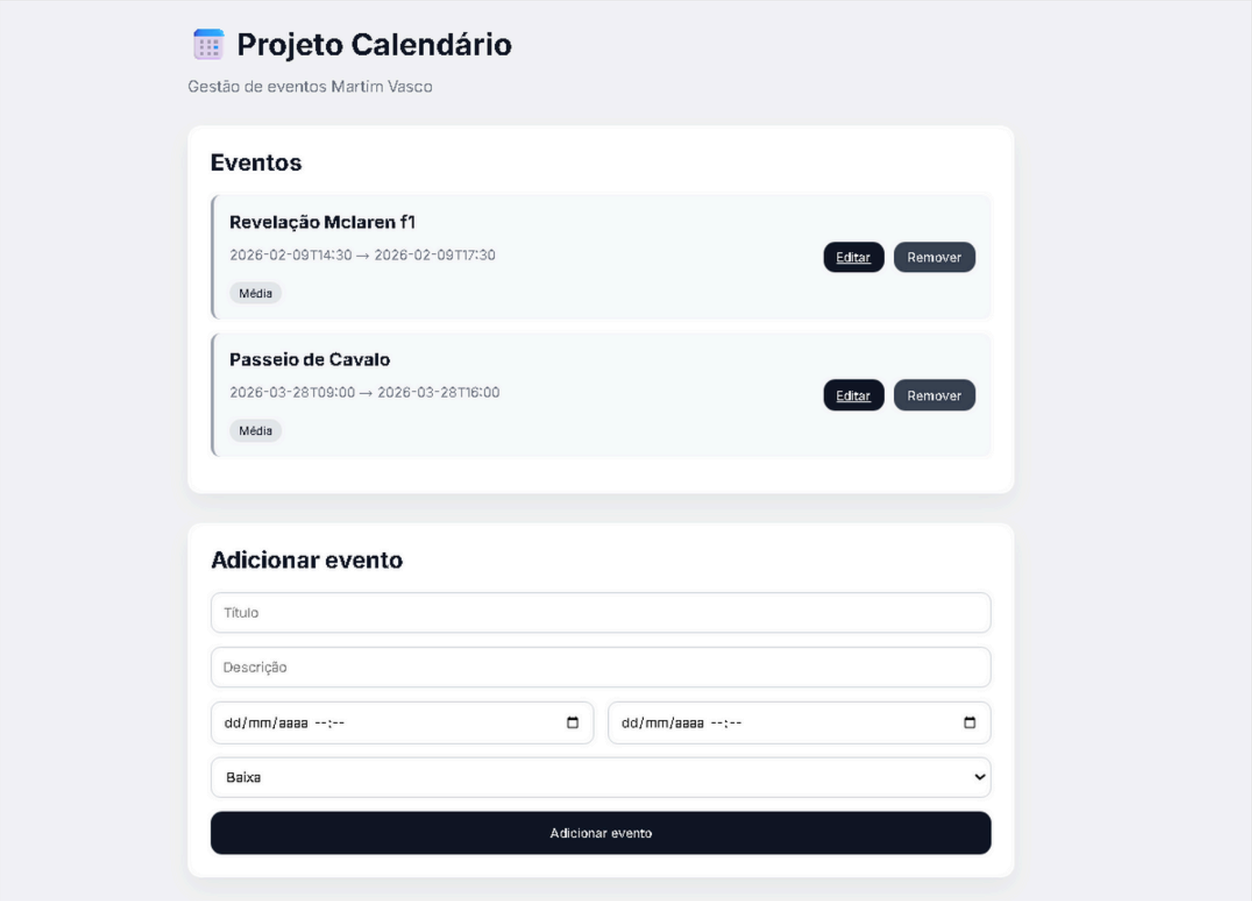
1

Os eventos são guardados num ficheiro JSON, permitindo persistência dos dados entre execuções da aplicação.



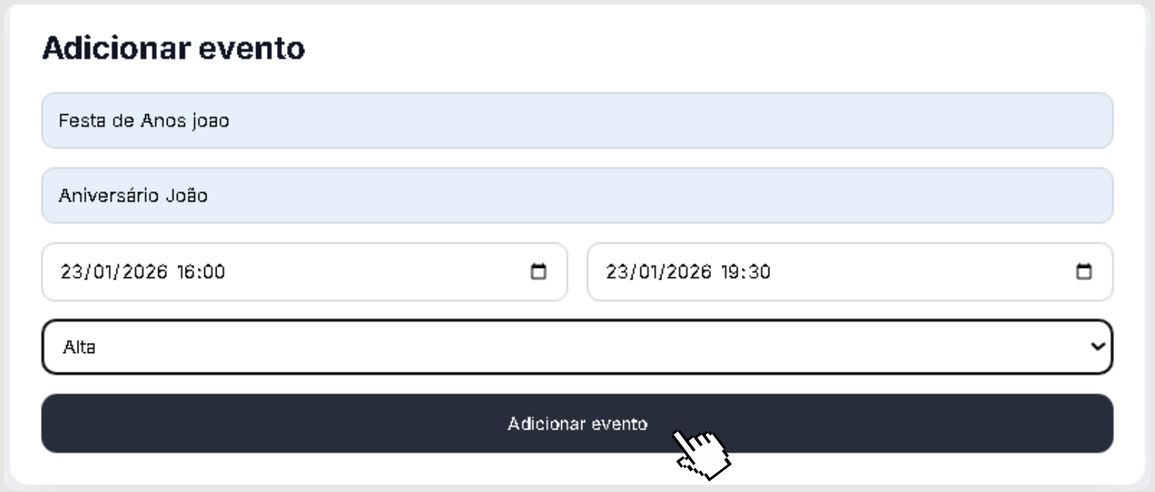
Interface Web

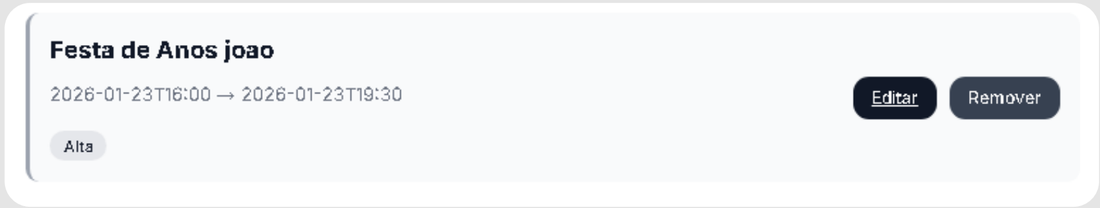
Interface Web da Aplicação

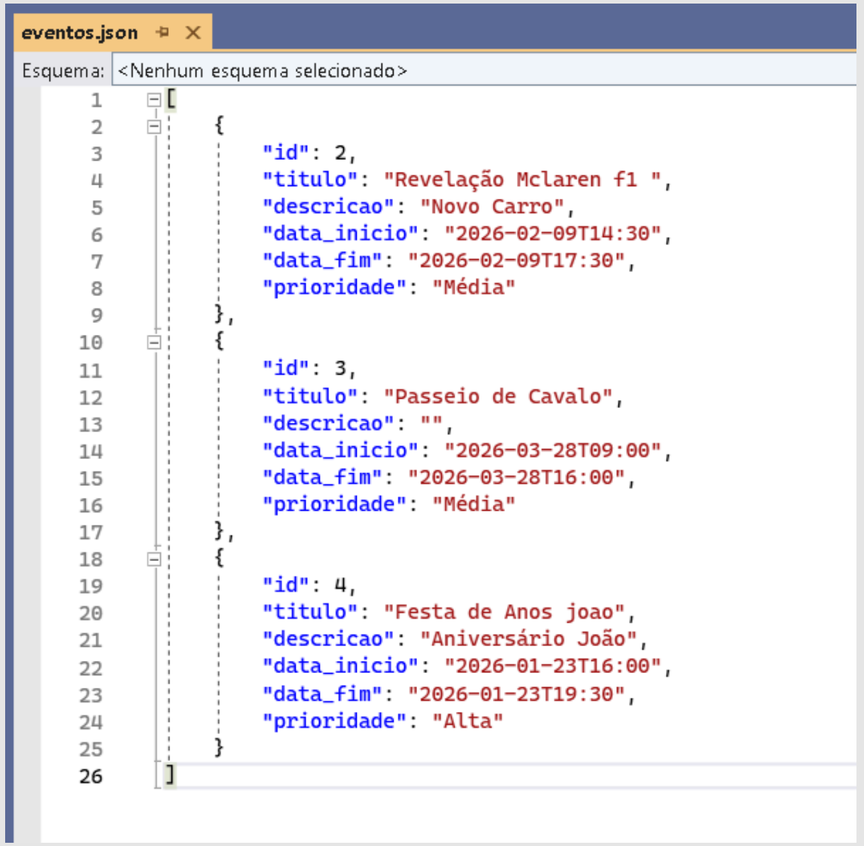


A aplicação disponibiliza uma interface web desenvolvida com Flask, permitindo adicionar, editar e remover eventos de forma intuitiva. http://127.0.0.1:5000/

Fluxo de Funcionamento

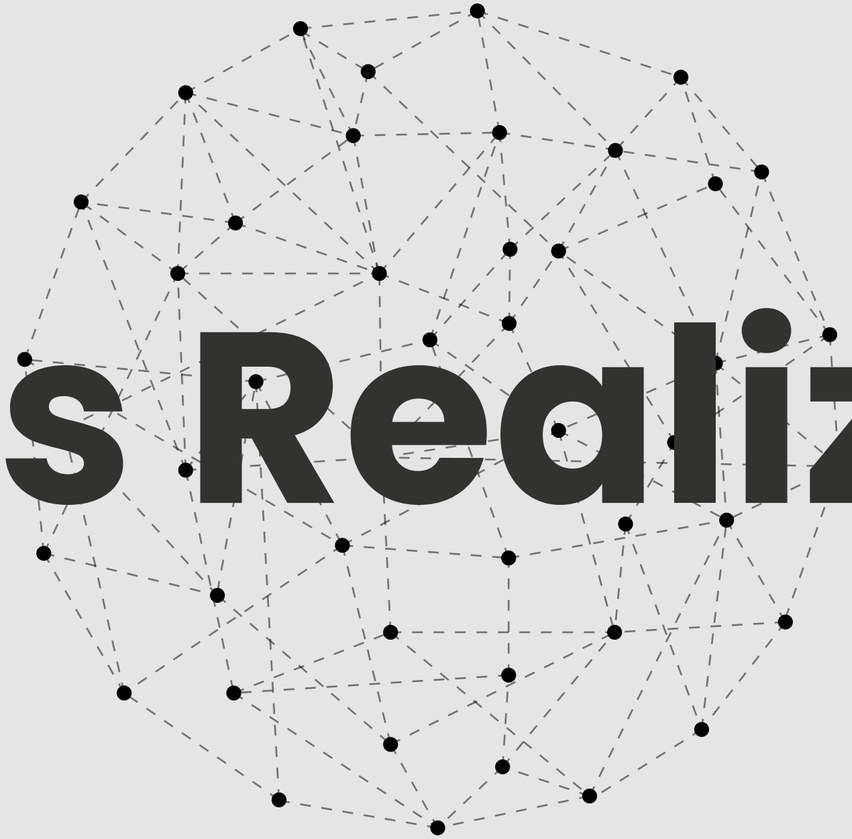






O utilizador interage com a interface web, os dados são processados em Python e armazenados em JSON.

Testes Realizados



Testes

```
PS C:\Users\marti\OneDrive\Ambiente de Trabalho\Projeto_Calendario\E4_Projeto> python app.py
>>
True
False
PS C:\Users\marti\OneDrive\Ambiente de Trabalho\Projeto_Calendario\E4_Projeto>
```

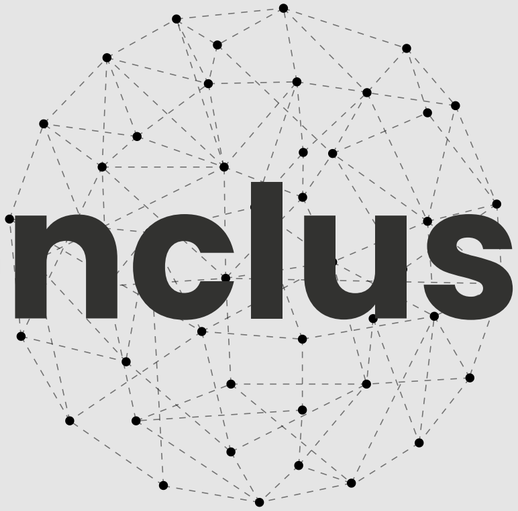
Foram realizados testes para validar a criação de eventos e deteção de sobreposição.

Limitações e melhorias futuras

Tecnologias Utilizadas

1. Visualização em calendário mensal
2. Filtros por prioridade
3. Filtros por prioridade

Conclusão



O projeto cumpre os requisitos propostos e demonstra a aplicação de POO, validações e persistência de dados.