

Documentação do Sistema Gerenciador de Alunos

Equipe:

Ana Clara da Silva

Bruno Gomes Robim

Guilherme de Freitas Romao Vieira

José Rogério Do Nascimento

João Pedro Alvarado Cardoso

💡 Visão Geral

Este sistema é uma aplicação web desenvolvida em **Python** com **Flask** e **HTML** para gerenciar alunos, suas notas e calcular automaticamente a média e a situação (aprovado/reprovado).

Os dados são armazenados em um arquivo **alunos.json**, permitindo persistência das informações mesmo após reinicialização do servidor.

◆ Funcionalidades

1. Cadastro de Alunos

- **Nome do aluno**
- **4 notas** (avaliações)
- **Cálculo automático** da média e situação (aprovado se média ≥ 7 , reprovado caso contrário).

2. Listagem de Alunos

- Exibe todos os alunos cadastrados em uma tabela.
- Mostra:
 - **Nome**
 - **Notas** (N1, N2, N3, N4)
 - **Média**
 - **Situação** (Aprovado/Reprovado)

3. Busca de Alunos

- Permite filtrar alunos pelo **nome**.

4. Edição de Notas

- Possibilidade de **alterar uma nota específica** de um aluno.

5. Exclusão de Alunos

- Remove um aluno do sistema após confirmação.

Tecnologias Utilizadas

- **Backend:**
 - **Python** (Linguagem principal)
 - **Flask** (Framework web para rotas e renderização)
 - **JSON** (Armazenamento dos dados em alunos.json)
- **Frontend:**
 - **HTML** (Estrutura da página)
 - **CSS** (Estilização básica)
 - **JavaScript** (Interações dinâmicas, como confirmação de exclusão)

6. Estrutura do projeto

```

GerenciadorAlunos/
  └── Documents/
    ├── Slide_apresentacao.pptx
    └── documentacao.pdf
      # Apresentação do projeto
      # Documentação detalhada do sistema

  └── templates/
    └── css/
      └── base.css
        # Estilos CSS principais

    ├── base.html
    ├── index.html
    ├── adicionar.html
    ├── buscar.html
    └── editar.html
      # Template base (HTML principal)
      # Página inicial (lista de alunos)
      # Formulário de cadastro
      # Página de busca
      # Página de edição de notas

  └── tests/
    ├── test_t1_adicionar.py
    ├── test_t2-fora_intervalo.py
    ├── test_t3-campo_vazio.py
    ├── test_t4-calc_media.py
    ├── test_t5-edicao_nota.py
    └── test_t6-busca_aluno.py
      # Teste: Adicionar aluno
      # Teste: Notas fora do intervalo (0-10)
      # Teste: Campos obrigatórios vazios
      # Teste: Cálculo da média
      # Teste: Edição de notas
      # Teste: Busca de alunos

  ├── app.py
  └── alunos.json
    # Aplicação Flask (rotas e lógica)
    # Armazenamento dos dados dos alunos

```

⚙️ Como Executar o Projeto

Pré-requisitos

- Python 3.x instalado
- Flask instalado (pip install flask)

Passos para Execução

1. Clone o repositório ou crie os arquivos necessários.
2. Execute o servidor Flask:

python app.py

3. Acesse no navegador:

<http://localhost:5000>

📝 Observações

- Os dados são salvos em **alunos.json**, evitando perda de informações ao reiniciar o servidor.
- A média é calculada automaticamente como:

$$\text{média} = (N1 + N2 + N3 + N4) / 4$$

- Se a média for ≥ 7 , o aluno é aprovado; caso contrário, **Reprovado**.

Tabela de testes – Gerenciador Alunos

ID Teste	Tipo de Teste	Descrição do Teste	Condição/Entrada	Resultado Esperado
1	Funcional (Positivo)	Adicionar aluno válido	Nome: "Ana" Notas: 7, 8, 6.5, 9	Aluno é adicionado com sucesso Confirmação de exclusão aparece
2	Funcional (Negativo)	Adicionar aluno com nota inválida	Nome: "Pedro" Uma das notas: 15 Demais notas: 7	Sistema deve rejeitar a nota inválida Formulário não é submetido
3	Validação (Negativo)	Campos vazios	Nome vazio ou uma das notas vazia	Sistema deve impedir submissão Mensagem de erro deve aparecer
4	Cálculo (Positivo)	Verificação de cálculo de média	Notas: 7, 7, 7, 7	Média deve ser 7.0 Exibida corretamente na tabela
5	Funcional (Positivo)	Edição de notas	Aluno: "Jose" Notas iniciais: 5, 5, 5, 5 Edições: 9, 8, 7, 6	Notas devem ser atualizadas Média deve ser recalculada
6	Funcional (Positivo/Negativo)	Busca de alunos	Alunos cadastrados: Maria, Jose, Mario Termos de busca: "maria", "maRIO", "JOose", "jose"	Deve retornar resultados corretos (case-insensitive) Busca parcial deve funcionar Termo inválido não retorna resultados

Descrição:

Teste 1 - Adicionar Aluno Válido

Objetivo: Verificar se o sistema permite cadastrar um aluno com dados completos e válidos.

Pré-condições:

- Nenhum aluno cadastrado
- Sistema acessível via interface web

Fluxo Principal:

1. Acessar a página inicial
2. Clicar em "Adicionar Aluno"
3. Preencher:
 - a. Nome: "Ana"
 - b. Notas: 7, 8, 6.5, 9
4. Submeter o formulário

Resultados Esperados:

- Aluno aparece na lista com média calculada corretamente (7.625)
- Opção de exclusão disponível

Pós-condições:

- Aluno é removido após confirmação via modal

Teste 2 - Nota Inválida (Valor > 10)

Objetivo: Validar tratamento de erros para notas fora do intervalo permitido (0-10).

Critérios de Rejeição:

- Notas devem estar entre 0 e 10

Cenários Testados:

- Nota 15 em cada campo individual (4 variações)

Comportamento Esperado:

- ✗ Sistema deve bloquear submissão
- ✗ Exibir mensagem: "Nota deve estar entre 0 e 10"

Dados de Teste:

```
{  
  "nome": "Pedro",  
  "notas": ["7", "15", "7", "7"] # Variações para cada campo  
}
```

Teste 3 - Campos Obrigatórios

Tipo: Teste de Validação Negativa

Casos Cobertos:

1. Nome vazio
2. Campo de nota vazio (4 variações)

Fluxo de Erro:

1. Tentar submeter formulário incompleto
2. Verificar feedback visual

Saídas Esperadas:

- ✗ Highlight nos campos vazios
- ✗ Mensagem: "Campo obrigatório"

Teste 4 - Cálculo de Média

Requisito Verificado:

$\text{Média} = \frac{\sum_{i=1}^4 \text{Nota}_i}{4}$

Procedimento:

1. Cadastrar aluno com notas 7, 7, 7, 7
2. Verificar cálculo na tabela

Critério de Aceite:

- Média exibida: 7.0 (com precisão de 1 decimal)

Teste 5 - Edição de Notas

Fluxo Completo:

1. Cadastrar aluno com notas 5, 5, 5, 5
2. Para cada nota:
 - a. Selecionar "Editar"
 - b. Atualizar nota (9, 8, 7, 6 respectivamente)
 - c. Confirmar alteração

Verificações:

- Notas atualizadas na interface
 Média recalculada automaticamente (7.5 após todas as edições)

Teste 6 - Busca com Termos Variados

Matriz de Teste:

Termo Busca	Resultado Esperado
"maria"	Retorna aluno Maria
"maRIO"	Retorna Mario (case-insensitive)
"JOOse"	Nenhum resultado (negative test)
"jose"	Retorna Jose

Critérios Especiais:

- Busca parcial deve funcionar (ex: "Mari" retorna "Maria" e "Mario")
- Acentos devem ser ignorados (opcional)

Tabela de Cobertura de Requisitos

ID	Requisito Coberto	Casos de Teste Relacionados
R1	Cadastro de aluno	1, 2, 3
R2	Cálculo de média	4, 5
R3	Validação de entrada	2, 3
R4	Edição de dados	5
R5	Busca inteligente	6

LOGS dos Teste da aplicação Gerenciador de Alunos

Teste 1 - Adicionar aluno - (test_t1-adicionar.py)

```
(feipi) ✘ tests git:(main) ✘ pytest -v test_t1_adicionar.py
=====
platform linux -- Python 3.10.12, pytest-8.3.4, pluggy-1.5.0 -- /usr/bin/python3
cachedir: .pytest_cache
rootdir: /home/zzombi/DEV/GIT/StudyHub/GerenciadorAlunos/tests
collected 1 item

test_t1_adicionar.py::TesteAdicionar::test_adicionar_aluno PASSED [100%]

===== 1 passed in 4.12s =====
(feipi) ✘ tests git:(main) ✘
```

Teste 2 - Nota fora do intervalo permitido - (test_t2-fora_intervalo.py)

```
(feipi) ✘ tests git:(main) ✘ pytest -v test_t2-fora_intervalo.py
=====
platform linux -- Python 3.10.12, pytest-8.3.4, pluggy-1.5.0 -- /usr/bin/python3
cachedir: .pytest_cache
rootdir: /home/zzombi/DEV/GIT/StudyHub/GerenciadorAlunos/tests
collected 1 item

test_t2-fora_intervalo.py::TesteForaIntervalo::test_nota_invalida PASSED [100%]

===== 1 passed in 20.55s =====
(feipi) ✘ tests git:(main) ✘
```

Teste 3 - campos vazios - (test_t3-campo_vazio.py)

```
(feipi) ✘ tests git:(main) ✘ pytest -v test_t3-campo_vazio.py
=====
platform linux -- Python 3.10.12, pytest-8.3.4, pluggy-1.5.0 -- /usr/bin/python3
cachedir: .pytest_cache
rootdir: /home/zzombi/DEV/GIT/StudyHub/GerenciadorAlunos/tests
collected 1 item

test_t3-campo_vazio.py::TesteCampoVazio::test_campos_vazios PASSED [100%]

===== 1 passed in 20.30s =====
(feipi) ✘ tests git:(main) ✘
```

Teste 4 - cálculo da média - (test_t4-calc_media.py)

```
• (feipi) ~ tests git:(main) ✘ pytest -v test_t4-calc_media.py
=====
platform linux -- Python 3.10.12, pytest-8.3.4, pluggy-1.5.0 -- /usr/bin/python3
cachedir: .pytest_cache
rootdir: /home/zzombi/DEV/GIT/StudyHub/GerenciadorAlunos/tests
collected 1 item

test_t4-calc_media.py::TesteMedia::test_media_nota ✘ Teste de média passou! A média foi calculada corretamente como 7.0
PASSED

===== 1 passed in 8.36s =====
○ (feipi) ~ tests git:(main) ✘
```

Teste 5 - edição nota de aluno - (test_t5-edicao_aluno.py)

```
• (feipi) ~ tests git:(main) ✘ pytest -v test_t5-edicao_nota.py
=====
platform linux -- Python 3.10.12, pytest-8.3.4, pluggy-1.5.0 -- /usr/bin/python3
cachedir: .pytest_cache
rootdir: /home/zzombi/DEV/GIT/StudyHub/GerenciadorAlunos/tests
collected 1 item

test_t5-edicao_nota.py::TesteEdicaoNota::test_edita_ PASSED [100%]

===== 1 passed in 15.03s =====
○ (feipi) ~ tests git:(main) ✘
```

Teste 6 – busca de aluno - (test_t6-busca_aluno.py)

```
• (feipi) ~ tests git:(main) ✘ pytest -v test_t6-busca_aluno.py
=====
platform linux -- Python 3.10.12, pytest-8.3.4, pluggy-1.5.0 -- /usr/bin/python3
cachedir: .pytest_cache
rootdir: /home/zzombi/DEV/GIT/StudyHub/GerenciadorAlunos/tests
collected 1 item

test_t6-busca_aluno.py::TesteBuscaAluno::test_buscaraluno PASSED [100%]

===== 1 passed in 32.35s =====
○ (feipi) ~ tests git:(main) ✘
```