## CRUD Escolar com Algoritmo de Ordenação



Professores:

Henrique Ruiz Poyatos;

Rafael Priscila Cruz Moreira

Integrantes:
Lucas Aurélio Pereira Silva - 42113420
Guilherme Fernandes Loureiro - 42111626
Samuel Faria Garcia - 42111586
Thiago Luan Pereira - 42111150

## CRUD Escolar com Algoritmo de Ordenação



Tecnologias Utilizadas: Python Flask, SQLite, SQLAlchemy Componentes: API, Banco de Dados, ORM, Algoritmo de Ordenação Desenvolver um sistema CRUD para gerenciar dados escolares integrando algoritmos de ordenação para manipulação eficiente de dados.

## Principais Funcionalidades:

- Cadastro;
- Leitura;
- Atualização;
- Exclusão;

Com base nos dados: alunos, provas e matérias



Simplificar o gerenciamento de informações escolares facilitando a organização e o acesso rápido aos dados. Aplicamos conhecimentos em Python Flask, SQLite, SQLAlchemy e algoritmo de ordenação.

Esse projeto melhora organização de dados escolares, reduzindo erros e retrabalho na gestão de informações.



Backend: API desenvolvida em Python Flask

Banco de Dados: SQLite para armazenamento persistente

**ORM:** SQLAlchemy para mapeamento objeto-relacional



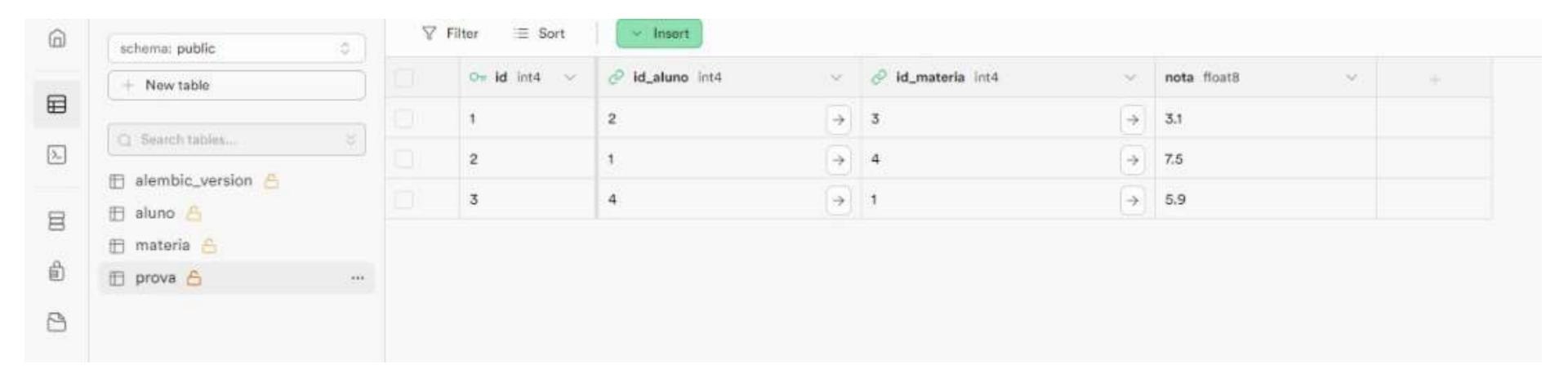
```
Estrutura das tabelas:
```

```
def upgrade():
19 V
           # ### commands auto generated by Alembic - please adjust! ###
20
           op.create table('aluno',
21
22
           sa.Column('id', sa.Integer(), nullable=False),
           sa.Column('nome', sa.String(length=100), nullable=False),
23
24
           sa.Column('idade', sa.Integer(), nullable=False),
25
           sa.Column('endereco', sa.String(length=200), nullable=True),
           sa.Column('ano_letivo', sa.String(length=100), nullable=True),
26
27
           sa.PrimaryKeyConstraint('id')
28
           op.create table('materia',
29
           sa.Column('id', sa.Integer(), nullable=False),
30
           sa.Column('nome', sa.String(length=100), nullable=False),
31
32
           sa.PrimaryKeyConstraint('id')
33
           op.create_table('prova',
34
           sa.Column('id', sa.Integer(), nullable=False),
35
           sa.Column('id_aluno', sa.Integer(), nullable=False),
36
37
           sa.Column('id_materia', sa.Integer(), nullable=False),
38
           sa.Column('nota', sa.Float(), nullable=False),
           sa.ForeignKeyConstraint(['id_aluno'], ['aluno.id'], ),
39
           sa.ForeignKeyConstraint(['id_materia'], ['materia.id'], ),
40
41
           sa.PrimaryKeyConstraint('id')
42
```



```
# Função de merge sort para ordenar as matérias por ID
def merge_sort(arr):
    Ordena uma lista de objetos Materia pelo atributo ID usar
    Parameters:
    arr : list
        Lista de objetos Materia a ser ordenada.
    if len(arr) > 1:
        meio = len(arr) // 2
        L = arr[:meio]
        R = arr[meio:]
        merge_sort(L)
        merge_sort(R)
        i = j = k = 0
        while i < len(L) and j < len(R):</pre>
            if L[i].id < R[j].id:</pre>
                arr[k] = L[i]
                i += 1
            else:
                arr[k] = R[j]
                j += 1
            k += 1
        while i < len(L):</pre>
            arr[k] = L[i]
            i += 1
            k += 1
        while j < len(R):</pre>
            arr[k] = R[j]
            j += 1
            k += 1
```







Projeto de Sistema CRUD funcional com ordenação eficiente; Dados escolares bem organizados e acessíveis; Pensamento futuro de expansão nas funcionalidades; expouna

Muito Obrigado!